

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Ciencias de la Tierra**

Validación de sensores para monitorear la calidad del aire en minería subterránea de carbón mediante el uso de autoencoders.

### **PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**Ingeniero de Minas**

Presentado por:

Carlos Tomas Cevallos Bautista

GUAYAQUIL - ECUADOR

Año: 2022

## DEDICATORIA

Dedico el presente trabajo a mis padres por sus esfuerzos, enseñanzas, sacrificios y sobre todo por su apoyo en cada uno de mis días como estudiantes, a mis hermanos que siempre me han apoyado, a mis tíos que sin sus consejos y ayudas nunca habría alcanzado esta meta, a Dayanna que es un pilar fundamental en mi vida y que me ayudó en mis momentos más difíciles, a mis amigos mejores amigos que siempre nos apoyábamos unos a otros, a mi grupo incondicional de amigos “La Sabliza” y a cada una de las personas que aportó con un granito de arena en mi formación y en mi vida.

## **AGRADECIMIENTOS**

Agradezco profundamente a mis tutores Samantha y Jorge por las enseñanzas, paciencia y por apoyarme para poder sacar adelante este proyecto. Agradezco también a cada uno de mis profesores que aportaron con su grano de arena en mi formación académica y profesional, en especial al Licenciado Gustavo por sus consejos.

## DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Carlos Tomas Cevallos Bautista y doy mi consentimiento para que la ESPOl realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Carlos Tomas Cevallos  
Bautista

# EVALUADORES

A handwritten signature in blue ink, reading "Samantha Jiménez Oyola". The signature is written in a cursive style with a large initial 'S'.

**MSc. Samantha Jiménez Oyola**

**Tutor y profesor Materia Integradora**

## RESUMEN

La minería es una de las actividades más antiguas en el mundo, y la gestión de los peligros es uno de los grandes retos que enfrenta la industria minera. En la minería subterránea, específicamente en la minería de carbón, uno de los métodos más empleado es el Longwall, sin embargo, la presencia de gas metano es un problema recurrente en este tipo de minería. En las minas de carbón, usualmente se emplean sensores de monitoreo para controlar la calidad del aire, pero estos sistemas pueden presentar errores de lectura ocasionando situaciones de riesgo. Por lo tanto, el control y la validación de los sensores es fundamental para garantizar datos fiables y por ende condiciones seguras en el interior mina. El presente proyecto tiene como objetivo validar los sensores que monitorean el gas metano dentro de las minas de carbón mediante la implementación de redes neuronales o AE no lineales. Para este propósito, se programó y se implementó AE para la validación de los sensores objeto de estudio, y se los comparó con otros métodos tradicionales como PCA, ICA y KPCA. Los resultados muestran que el método de AE tiene una exactitud mayor que los otros métodos evaluados y además presenta un menor error en la reconstrucción o validación de los datos. La información recaba en este proyecto puede contribuir en los procesos de monitoreo oportuno y eficaz de la calidad de aire en proyectos mineros de extracción de carbón en minería subterránea.

**Palabras claves:** Autoencoders, validación de sensores, metano, reconstrucción.

## **ABSTRACT**

*Mining is one of the oldest activities in the world, and hazard management is one of the great challenges facing the mining industry. In underground mining, specifically in coal mining, one of the most used methods is the Longwall, however, the presence of methane gas is a recurring problem in this type of mining. In coal mines, monitoring sensors are usually used to control air quality, but these systems can present reading errors, causing risk situations. Therefore, the control and validation of sensors is essential to guarantee reliable data and therefore safe conditions inside the mine. The objective of this project is to validate the sensors that monitor methane gas inside coal mines through the implementation of neural networks or nonlinear AE. For this purpose, AE was programmed and implemented for the validation of the sensors under study, and they were compared with other traditional methods such as PCA, ICA and KPCA. The results show that the AE method has a greater accuracy than the other methods evaluated and presents a lower error in the reconstruction or validation of the data. The information collected in this project can contribute to timely and effective monitoring processes of air quality in mining projects for coal extraction in underground mining.*

**Keywords:** *Autoencoders, sensor validation, methane, reconstruction.*

# ÍNDICE GENERAL

RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS .....	VI
SIMBOLOGÍA .....	VII
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS .....	X
CAPÍTULO 1 .....	11
1.    Introducción .....	11
1.1    Descripción del problema .....	12
1.2    Justificación del problema.....	13
1.3    Objetivos.....	13
1.3.1    Objetivo General .....	13
1.3.2    Objetivos Específicos .....	13
1.4    Marco teórico .....	13
1.4.1    Descripción del sitio .....	13
1.4.2    Método de explotación .....	14
1.4.3    Deep Learning.....	15
1.4.4    Validación de sensores .....	15
1.4.5    Autoencoders (AE).....	16
1.4.6    Análisis de componentes principales-PCA.....	17
CAPÍTULO 2.....	18
2.    Metodología .....	18
2.1    Etapa 1: Detección de sensores .....	18
2.1.1    Etapa 1: Datos de entrada .....	18

2.1.2	Etapa 1: Contaminación de datos .....	23
2.1.3	Etapa 1: Extracción de atributos.....	27
2.1.4	SPE y KDE .....	28
2.2	Etapa 2: Reconstrucción de datos .....	29
2.2.1	Métricas de reconstrucción .....	29
CAPÍTULO 3.....		30
3.	Resultados Y ANÁLISIS.....	30
3.1	Etapa 1: Detección de sensores .....	30
3.1.1	Etapa 1: Datos de entrada .....	30
3.1.2	Etapa 1: Contaminación de los datos.....	30
3.1.3	Etapa 1: Extracción de atributos.....	34
3.2	Etapa 2: Reconstrucción de datos .....	38
3.2.1	AE Reconstrucción.....	38
3.2.2	PCA Reconstrucción .....	40
3.2.3	ICA Reconstrucción.....	42
3.2.4	KPCA Reconstrucción.....	44
3.2.5	Métricas de reconstrucción .....	46
CAPÍTULO 4.....		47
4.	Conclusiones Y RECOMENDACIONES .....	47
4.1	Conclusiones .....	47
4.2	Recomendaciones .....	48
BIBLIOGRAFÍA.....		49
5.	ANEXOS .....	51
5.1	Anexo 1.....	51
5.2	Anexo 2.....	52
5.3	ANEXO 3.....	53

5.4 ANEXO 4 ..... 54

5.5 ANEXO 5 ..... 55

5.6 ANEXO 6 ..... 56

## ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
RMSE	Root mean square error
SPE	Square prediction error
KDE	Kernel density estimator
PCA	Principal component analysis
KPCA	Kernel principal component analysis
ICA	Independent component analysis
AE	Autoencoder

## SIMBOLOGÍA

MMTCO <sub>2</sub> E	Million metric tonnes of carbon Dioxide equivalents
m/s	Metros por segundo
m <sup>3</sup> /min	Metros cúbicos por minuto
°C	Grados Celsius
%HR	Concentración de humedad
hPa	Hectopascales
%CH <sub>4</sub>	Concentración de metano
kPa	KiloPascales
Pa	Pascal

## ÍNDICE DE FIGURAS

Figura 1.1 Ubicación de la concesión minera. ....	14
Figura 1.2 Arquitectura de la red neuronal AE. Fuente: Yan et al, 2016.....	16
Figura 2.1 Metodología del proyecto.....	18
Figura 2.2 Monitoreo de un Autoencoder.....	19
Figura 2.3 Mapa de la mina que contiene la ubicación de cada sensor.....	19
Figura 2.4 Tendencia del sensor #4 TP1721. ....	24
Figura 2.5 Tendencia del sensor #8 RH1712. ....	25
Figura 2.6 Tendencia del sensor #9 BA1713.....	25
Figura 2.7 Tendencia del sensor #10 MM261.....	26
Figura 2.8 Matriz de confusión y sus componentes. Fuente: Barrios, 2019.....	28
Figura 3.1 Falla completa al sensor #4, código.....	31
Figura 3.2 Falla tipo drifting al sensor #8, código.....	31
Figura 3.3 Falla tipo Degradación de la precisión al sensor #9, código. ....	32
Figura 3.4 Falla tipo Bias al sensor #10, código. ....	32
Figura 3.5 Falla completa, sensor #4.....	33
Figura 3.6 Falla tipo drifting, sensor #8.....	33
Figura 3.7 Falla degradación de la precisión, sensor #9.....	34
Figura 3.8 Falla tipo Bias, sensor #10.....	34
Figura 3.9 SPE límite del AE .....	35
Figura 3.10 Reconstrucción AE, Sensor #4 TP1721.....	38
Figura 3.11 Reconstrucción AE, Sensor #8 RH1712.....	39
Figura 3.12 Reconstrucción AE, Sensor #9 BA1713 .....	39
Figura 3.13 Reconstrucción AE, Sensor #10 MM261 .....	40
Figura 3.14 Reconstrucción PCA, Sensor #4 TP1721 .....	40
Figura 3.15 Reconstrucción PCA, Sensor #8 RH1712 .....	41
Figura 3.16 Reconstrucción PCA, Sensor #9 BA1713.....	41
Figura 3.17 Reconstrucción PCA, Sensor #10 MM261.....	42
Figura 3.18 Reconstrucción ICA, Sensor #4 TP1721 .....	42
Figura 3.19 Reconstrucción ICA, Sensor #8 RH1712.....	43
Figura 3.20 Reconstrucción ICA, Sensor #9 BA1713 .....	43

Figura 3.21 Reconstrucción ICA, Sensor #10 MM261 .....	44
Figura 3.22 Reconstrucción KPCA, Sensor #4 TP1721.....	44
Figura 3.23 Reconstrucción KPCA, Sensor #8 RH1712 .....	45
Figura 3.24 Reconstrucción KPCA, Sensor #9 BA1713 .....	45
Figura 3.25 Reconstrucción KPCA, Sensor #10 MM261 .....	46

## ÍNDICE DE TABLAS

Tabla 2.1 Características de los sensores, tipos de datos que recopilan. ....	20
Tabla 2.2 Características de los sensores, tipos de datos que recopilan .....	21
Tabla 2.3 Características de los sensores, tipos de datos que recopilan. ....	22
Tabla 2.4 Parámetros estadísticos de cada sensor. ....	22
Tabla 2.5 Parámetros de sensores seleccionado para la contaminación. ....	24
Tabla 3.1 SPE límite de todos los métodos aplicados. ....	36
Tabla 3.2 Matriz de confusión AE. ....	36
Tabla 3.3 Matriz de confusión PCA.....	36
Tabla 3.4 Matriz de confusión KPCA. ....	37
Tabla 3.5 Matriz de confusión ICA. ....	37
Tabla 3.6 Métricas de la matriz de confusión.....	37
Tabla 3.7 Métricas de reconstrucción. ....	46

# CAPÍTULO 1

## 1. INTRODUCCIÓN

La minería es una de las actividades más antiguas en el mundo, y la gestión de los peligros que con lleva esta actividad es uno de los grandes retos que enfrenta la industria minera. La minería subterránea de carbón es especialmente peligrosa debido a concentración de gases, principalmente gas metano en las actividades de extracción minera (Kozielski et al., 2021). Un estudio realizado en el año 2010 estimó que las emisiones a nivel mundial del gas metano, que fueron producidas por las minas de carbón, alcanzaron un valor aproximado de 584 MMTCO<sub>2</sub>E (Million metric tonnes of carbon Dioxide equivalents), siendo la minería responsable de un 8% total de las emisiones del gas metano a nivel mundial (Global Methane Initiative, 2011).

El monitoreo constante del gas metano es indispensable en las minas de carbón, debido a su alta emisión. Este gas provoca asfixia a los trabajadores y ha ocasionado numerosas pérdidas humanas debido a explosiones provocadas por gas el metano. En Colombia, en el año 2010 el índice de fatalidad que presentó la actividad minera fue de 173 víctimas y hasta junio de 2011 se reportó 81 víctimas (Romero Acero et al., 2013).

El monitoreo oportuno de los gases en interior mina permite controlar que las concentraciones y la presencia de metano esté dentro de los estándares establecidos. Este monitoreo de la calidad de aire en las minas subterráneas se realiza usualmente mediante el empleo de sensores (Gómez et al., 2021). Sin embargo, es importante que los sensores estén calibrados y validados para que los resultados de las mediciones sean confiables. Varios de los motivos por los cuales los sensores necesitan una previa validación para un funcionamiento óptimo son: fallas por reconexiones de electricidad, reinicio de sistema, entre otros.

La validación de los sensores de monitoreo es de suma importancia para evitar tener datos fuera de lo normal. Una de las técnicas estadísticas más empleadas en los procesos de evaluación de los datos de sensores es el análisis de componentes principales PCA (Principal Component Analysis), el cual es un método que permite reducir las dimensiones o variables que se tengan de los datos estadísticos recolectados por cada uno de los sensores. De esta manera, al reducir las dimensiones se conserva la información relevante para seguir con la validación de los sensores (Amat Rodrigo,

2017). El método estadístico PCA, es utilizado en diferentes industrias para el mejor manejo de datos.

Otro método ampliamente utilizado es el método de Autoencoder (AE), el cual es un medio eficaz para la extracción de características no lineales y para la reducción de dimensiones. Los AE variantes representan una mejora con respecto a los AE tradicionales (Yan et al., 2016).

El presente proyecto busca la validación de sensores de medición de gas metano en minas de carbón mediante la aplicación de AE. Esto ayudaría en la automatización de los procesos de control en temas de seguridad y salud ocupacional en minería, debido a que en algunas minas aún se trabaja con sistemas poco eficientes para el control de los gases en interior mina.

## **1.1 Descripción del problema**

En la minería subterránea, específicamente en la minería de carbón, se trabaja con el método Longwall el cual es un método de explotación de carbón muy usado en minas subterráneas, sin embargo, la presencia de gas metano es un problema recurrente en este tipo de minería, debido a que este gas es el causante de explosiones en el interior de mina por sus altas concentraciones acumuladas por el método de explotación empleado. Usualmente se emplean sensores de monitoreo para controlar la calidad del aire, pero estos sistemas pueden presentar errores de lectura ocasionando situaciones de riesgo. Por lo tanto, el control y la validación de los sensores es fundamental para garantizar condiciones seguras en el interior mina. Sin embargo, estos sistemas pueden presentar errores de lectura si no se tiene un control oportuno; estos errores se dan debido a reconexiones de los sistemas operativos, fluctuaciones de voltaje que desconfiguran los equipos, entre otros. Por lo tanto, requiere procesos de validación para evitar lecturas erróneas de la concentración de gas metano y otras variables de calidad de aire subterráneo.

## **1.2 Justificación del problema**

El presente proyecto surge de la necesidad de un monitoreo oportuno y eficaz de la calidad de aire en proyectos mineros de extracción de carbón en minería subterránea. Para este propósito, la validación de sensores de medición es fundamental ya que nos brindará datos confiables de la calidad de aire. Esta información puede ser utilizadas por las autoridades para tener un mejor control de datos de concentración de gases que se tienen en la mina, específicamente en los sectores donde se encuentran ubicados cada uno de los sensores.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Validar sensores para monitoreo de gas metano en minas de carbón, mediante la implementación de autoencoders no lineales.

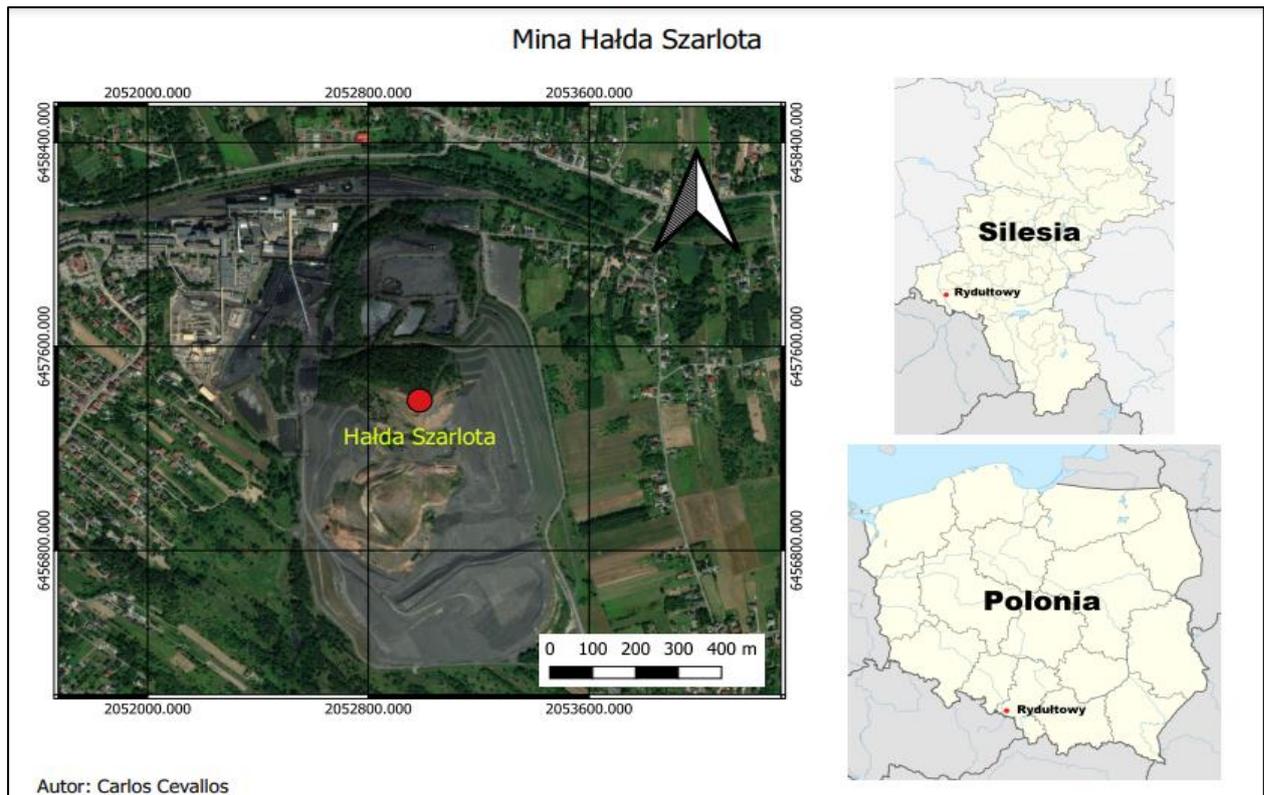
### **1.3.2 Objetivos Específicos**

1. Implementar el método autoencoders, PCA, ICA y KPCA para el análisis de los datos de los sensores.
2. Generar estadísticos de control para el monitoreo de datos errados de la red de sensores, en la detección y reconstrucción.
3. Reconstruir los datos errados para estimar los valores reales de los sensores a partir del entrenamiento de estos.

## **1.4 Marco teórico**

### **1.4.1 Descripción del sitio**

El proyecto minero objeto de estudio es la concesión minera Hałda Szarlota, se encuentra ubicado dentro de ciudad de Rydułtowy, ubicada en la provincia de Silesia, sur de Polonia. En la Figura 1.1 se observa la ubicación de la mina.



**Figura 1.1 Ubicación de la concesión minera.**

De acuerdo con información (MINDAT, 2022), la zona donde está situada la concesión minera está compuesta principalmente por esquistos negros, buchita, clinker, carbón, porcelanita, parabasalto, escoria de metacarbón y metamudstone.

#### **1.4.2 Método de explotación**

El método de explotación empleado en la mina de carbón en la que se recolectaron los datos es el “Longwall”, aplicado en minería subterránea, siendo uno de los mejores para minería de carbón. Longwall o también conocido como minado por lonjas o tajos largos, es un método que se aplica depósitos con potencia uniforme, habitualmente en ocurrencias de grandes extensiones, debido a que el área en el que se trabaja debe ser bien soportada (Quispe Gutiérrez mis Hermanos Karen & Jean Pool Muriel Quispe, 2016).

### **1.4.3 Deep Learning**

El método de aprendizaje profundo o Deep Learning en inglés es uno de los factores principales que permiten lograr un sistema de inteligencia artificial de nivel humano y en ocasiones sobre humano. Además, este método posee grandes conexiones con nuestras funciones cerebrales y con distintos hallazgos neurocientíficos (Matsuo et al., 2022).

Este aprendizaje permite que los distintos modelos computacionales (los cuales se componen de múltiples capas de procesamiento) aprendan a realizar presentaciones de datos con varios niveles de abstracción. Además, el Deep learning descubre estructuras intrincadas en grandes conjuntos de datos, esto lo realiza mediante el uso de algoritmos de retropropagación con el fin de indicar como una maquina cambia sus parámetros internos que se usan para calcular la representación en cada capa a partir de la representación en la capa anterior (Lecun et al., 2015).

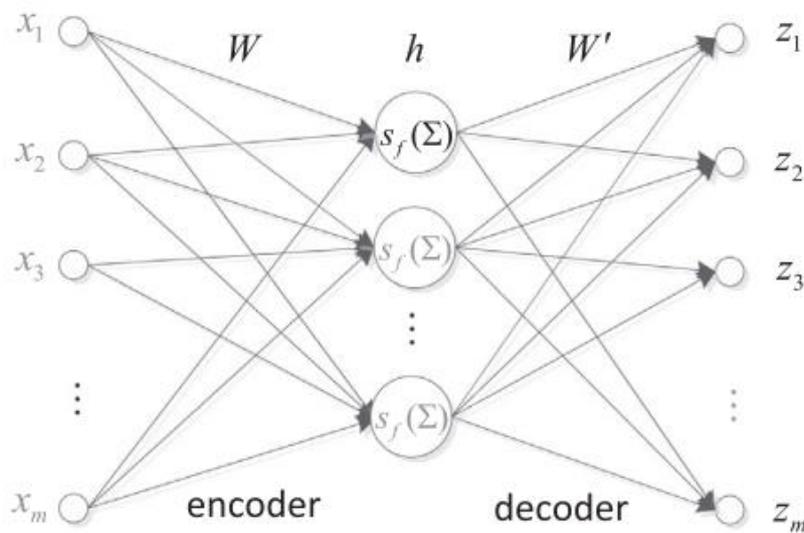
### **1.4.4 Validación de sensores**

La validación de sensores es una implementación necesaria en la aplicación de la inferencia de sensores, modelo de control predictivo y la optimización de procesos industriales. La validación está basada en modelos estadísticos que se guían en el uso de procesos normales para la construcción de modelos. Los modelos de análisis de componentes principales o PCA, son utilizado para la extracción de la correlación variable de los datos (Yoo et al., 2006).

Al momento de realizar la validación de los sensores, primero se realiza la optimización del estado estacionario, además de encontrar valores objetivos óptimos para las variables controladas y las varianzas manipuladas. Si los sensores se encuentran defectuosos, los valores objetivos óptimos no son válidos. Por ello, se requiere un enfoque de validación efectivo para la detección e identificación de sensores defectuosos en línea. A partir de la detección de los sensores defectuosos se estima la magnitud de la falla y se reemplaza con la mejor reconstrucción para mantener el sistema de control en línea. Nos referimos a detección, identificación y reconstrucción de fallas del sensor como validación de sensores (Qin & Li, 1999).

### 1.4.5 Autoencoders (AE)

Los codificadores automáticos o autoencoders son redes neuronales que no están supervisadas, tiene como estructura un codificador y una parte decodificadora. El codificador realiza la codificación datos de entrada a una representación dimensional más baja. Por otra parte, el decodificador tiene el trabajo de la reconstrucción de los datos de entrada (Abiri et al., 2019).



**Figura 1.2 Arquitectura de la red neuronal AE. Fuente: Yan et al, 2016.**

El codificador tiene la función de encontrar una representación comprimida para la entrada de datos dada. Según (Yan et al., 2016) los autoencoders se le asigna una función  $f(x)$  esta función asigna una entrada  $x \in R^d$  a una representación oculta  $h \in R^{d'}$  de tal forma que la Ecuación 1.1:

$$h = f(x) = s_f(Wx + b) \quad (1.1)$$

Donde  $W$  es una matriz  $d' \times d$  y  $b \in R^{d'}$  es un vector de sesgo,  $s_f$  es una función de activación no lineal, una función logística sigmoide  $s_f(x) = \frac{1}{1+e^{-x}}$ .

En otra parte el decodificador es un reflejo del codificador, lo que obliga que el AE a la elección de una representación comprimida más rica que tenga el objetivo de la

reconstrucción de la entrada lo más real posible. En este caso la función  $g$  mapea la reconstrucción oculta y  $h$  vuelve a la reconstrucción  $z$ , la Ecuación 1.2 lo demuestra.

$$z = g(h) = s_g(W'h + b') \quad (1.2)$$

Donde  $s_g$  es la función de activación para el decodificador y también es una función sigmoide. Los parámetros del vector son un vector de polarización  $b' \in R^d$  y una matriz  $dx d'$ , además  $W' = W^T$ .

Para la reconstrucción, se da un conjunto de datos de entrenamiento  $D = \{(x|x \in R^d)\}$ , la función objetivo del AE es la suma de todos los errores de reconstrucción entre la entrada  $x$  y su reconstrucción  $z$ . La función objetivo se optimiza para la obtención de los parámetros  $\theta = \{W, b, W', b'\}$ , la función objetivo se describe en la Ecuación 1.3.

$$J_{AE}(\theta) = \sum_{x \in D} L(x, g(f(x))) \quad (1.3)$$

Donde  $L$  es la reconstrucción del error, usualmente algunas opciones incluyen el error al cuadrado Ecuación 1.4

$$L(x, z) = \|x - z\|^2 = \|x - g(f(x))\|^2 = \|x - s_g(W'(s_f(Wx + b) + b'))\|^2 \quad (1.4)$$

#### 1.4.6 Análisis de componentes principales-PCA

EL PCA es uno de los métodos estadísticos que permite la reducción de la complejidad de espacios muestrales que poseen muchas dimensiones y además conservan su información más relevante. Los PCA pertenecen a las técnicas llamadas unsupervised learning. Estas técnicas tienen el objetivo predecir una variable  $Y$  a partir de predictores, para ellos se debe disponer de  $p$  características  $(X_1, X_2, \dots, X_p)$  y de la variable  $Y$  medidas en  $n$  observaciones. Esta técnica tiene como objetivo extraer información utilizando predictores sin la respuesta de la variable  $Y$  (Amat Rodrigo, 2017).

# CAPÍTULO 2

## 2. METODOLOGÍA

Para la validación de sensores se necesitó de 2 etapas: (1) Detección de datos - Pretratamiento de datos, y (2) Reconstrucción de los datos - Validación. Ambas etapas fueron desarrolladas en la herramienta Matlab. La Figura 2.1 presenta las fases de la metodología. La Figura 2.2 representa el monitoreo del Autoencoder aplicado en este proyecto.

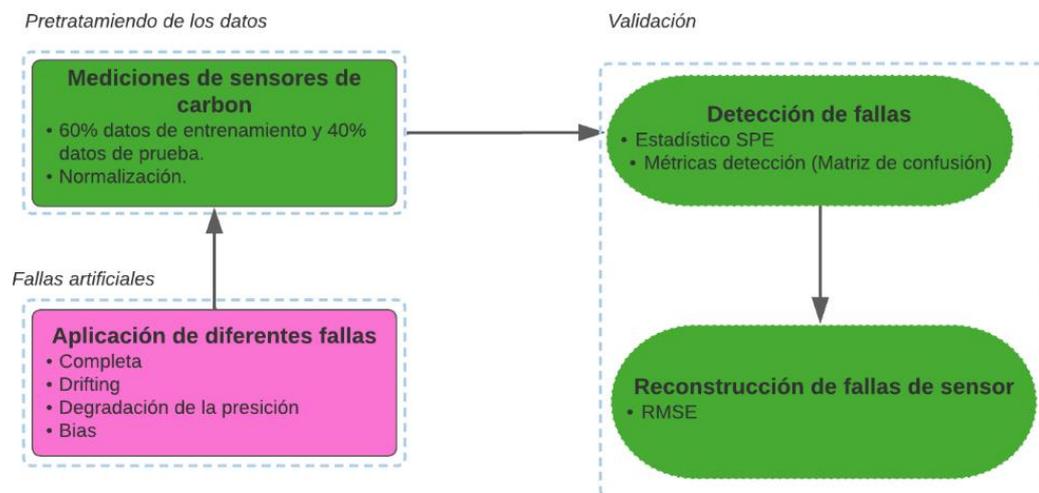


Figura 2.1 Metodología del proyecto

### 2.1 Etapa 1: Detección de sensores

La Etapa 1 consistió en la detección de los datos de cada uno de los sensores y esta se dividió en fases.

#### 2.1.1 Datos de entrada

En esta fase los datos fueron recopilados por los sensores con los diferentes parámetros de concentraciones tales como: metano, temperatura, presión y humedad. El conjunto de datos que se recolectó de la mina de carbón corresponde a información de sensores ubicados en varios lugares alrededor de la mina. El mapa del área de interés de la mina se presenta en la Figura 2.3. Los sensores, el tipo de acción que los activa y los umbrales de advertencia se presentan en las Tablas 2.1, 2.2 y 2.3.

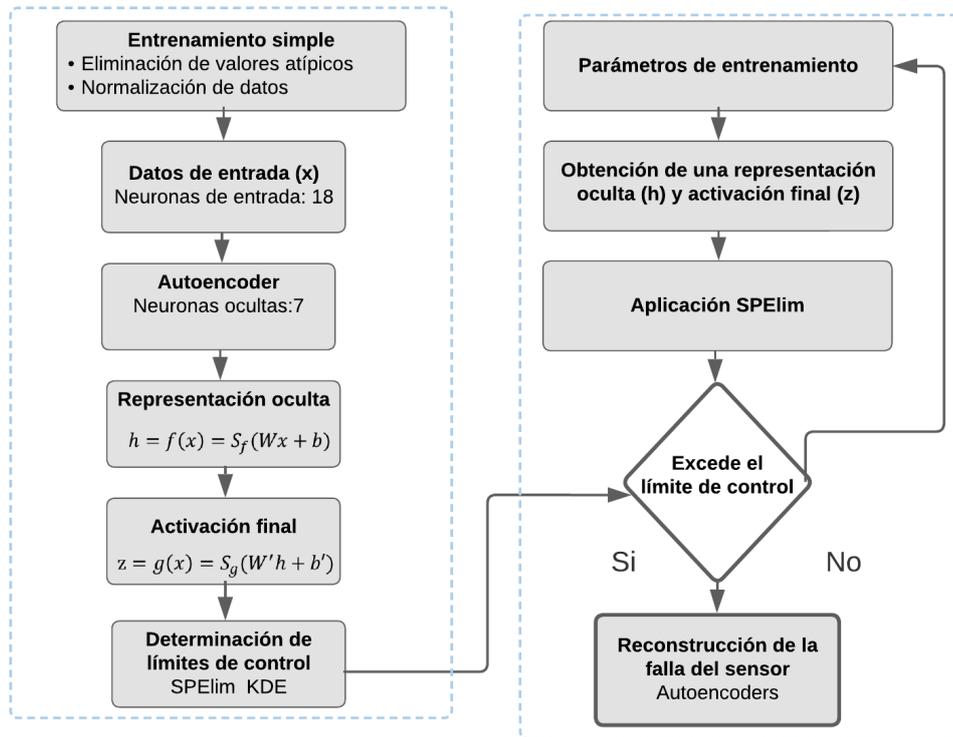


Figura 2.2 Monitoreo de un Autoencoder.

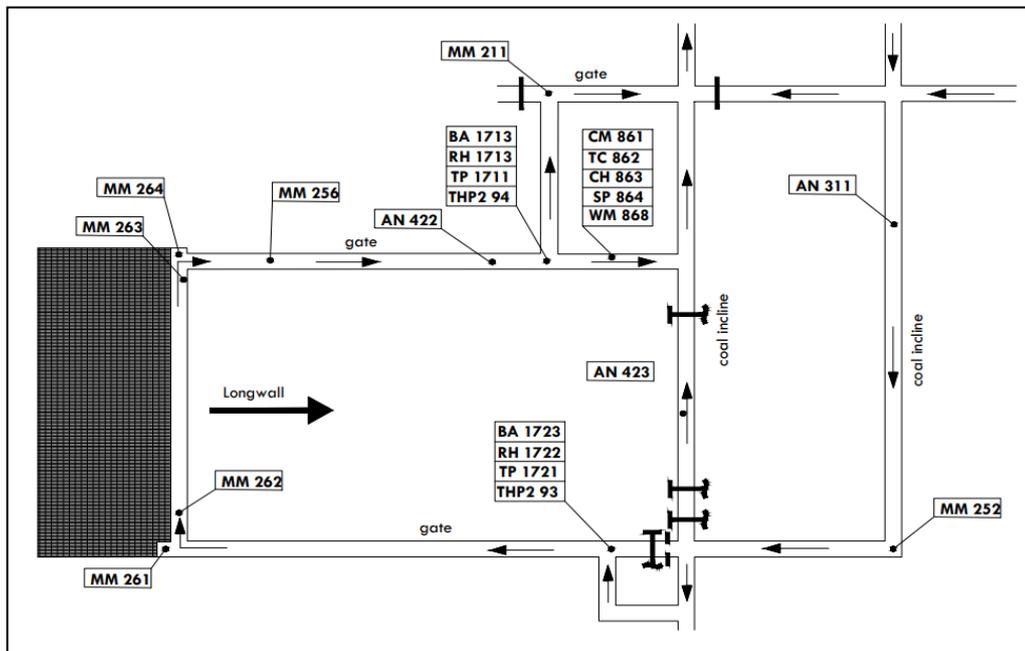


Figura 2.3 Mapa de la mina que contiene la ubicación de cada sensor.

**Tabla 2.1 Características de los sensores, tipos de datos que recopilan.**

<i>AN311</i>	Anemómetro [m/s] Tipo de sensor: anemómetro [-5, 5] Tipo: alarmante
<i>AN422</i>	Anemómetro [m/s] Tipo de sensor: anemómetro [-5, 5] Tipo: desconectar
<i>AN423</i>	Anemómetro [m/s] Tipo de sensor: anemómetro [-5, 5] Tipo: desconectar
<i>TP1721</i>	Temperatura [°C] Tipo de sensor: temperatura THP (sensor de tres componentes THP2/93) Tipo: registro
<i>RH1722</i>	Humedad [%HR] Tipo de sensor: humedad THP (sensor de tres componentes THP2/93) Tipo: registro
<i>BA1723</i>	Barómetro [hPa] Tipo de sensor: barómetro THP (sensor de tres componentes THP2/93) Tipo: registro
<i>TP1711</i>	Temperatura [°C] Tipo de sensor: temperatura THP (sensor de tres componentes THP2/94) Tipo: registro
<i>RH1712</i>	Humedad [%HR] Tipo de sensor: humedad THP (sensor de tres componentes THP2/94) Tipo: registro
<i>RH1712</i>	Barómetro [hPa] Tipo de sensor: barómetro THP (sensor de tres componentes THP2/94) Tipo: registro

**Tabla 2.2 Características de los sensores, tipos de datos que recopilan**

<i>MM252</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk Tipo: apagado                  Valor del umbral A (alarma): 2,0% valor del umbral W (aviso): 1,5%</p>
<i>MM261</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk Tipo: apagado                  Valor del umbral A: 1,5 %                  Valor del umbral W: 1,0 %</p>
<i>MM262</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk Tipo: apagado                  Valor del umbral A: 1,0 %                  Valor del umbral W: 0,6 %</p>
<i>MM263</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk                  Tipo: apagado                  Valor del umbral A: 1,5 %                  Valor del umbral W: 1,0 %</p>
<i>MM264</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk Tipo: apagado                  Valor del umbral A: 1,5 %                  Valor del umbral W: 1,0 %</p>
<i>MM256</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk                  Tipo: apagar                  Valor del umbral A: 1,5 %                  Valor del umbral W: 1,0 %</p>
<i>MM211</i>	<p>Medidor de metano [%CH4]                  Tipo de sensor: medidor de metano MM-2PWk Tipo: apagado                  Valor del umbral A: 2,0%                  Valor del umbral W: 1,5%</p>
<i>CM861</i>	<p>Medidor de metano de alta concentración [%CH4] Tipo de sensor: medidor de metano [0, 100]                  Tipo: registro</p>

**Tabla 2.3 Características de los sensores, tipos de datos que recopilan.**

CR863	Sensor para diferencia de presión en la brida de Drenaje de metano [Pa] Tipo de sensor: diferencia de presión [0, 250] Tipo: registro
P_864	Presión dentro de la tubería de drenaje de metano [kPa] Tipo de sensor: presión [0, 110] Tipo: registro
TC862	Temperatura dentro de la tubería [°C] Tipo de sensor: temperatura [10, 40] Tipo: registro
WM868	Entrega de metano calculada según CM, CR, P, TC [m3/min] Tipo de sensor: suministro de metano [0, 50] Tipo: registro

Los datos recopilados por los sensores son importantes para el análisis del proyecto, además de que cada uno de los sensores registraba atributos diferentes se tiene los mínimos, máximos, media, desviación estándar y mediana que tiene cada uno de los sensores, los cuales están representados en la Tabla 2.4.

Cada uno de los 18 sensores con los que se trabajó tiene 2185 datos de lectura, el tiempo en el que se recolectó cada dato fue de cada hora, estos 18 sensores poseen una medición distribuida entre sensores de humedad, temperatura, presión y metano.

**Tabla 2.4 Parámetros estadísticos de cada sensor.**

Sensor	Mínimo	Máximo	Media	Desviación estándar	Mediana
AN311	-266	5	3.484	0.611	3.6
AN422	0	2.4	1.655	0.128	1.6
AN423	-2.4	5.3	1.498	0.33	1.
TP1721	0	27.9	25.477	0.932	25.4
RH1722	0	71	49.283	6.143	48
BA1723	0	1131.7	1106.161	7.625	1105.9
TP1711	0	31.2	28.894	0.757	28.8

RH1712	0	86	68.687	7.268	69
BA1713	0	1130.9	1105.597	7.617	1105.3
MM252	-0.1	30	0.038	0.121	0
MM261	0	30	0.049	0.125	0
MM262	-0.2	30	0.051	0.136	0
MM263	-2	30	0.248	0.197	0.2
MM264	-2	40	0.327	0.206	0.3
MM256	0	30	0.43	0.204	0.4
MM211	-2	30	0.7	0.151	0.7
CM861	-0.2	67.7	32.92	21.395	43.7
CR863	-8	258	75.081	55.161	78
P_864	0	435.4	86.967	29.158	94.2
TC862	0	40.5	29.898	9.898	32.9
WM868	0	6.39	1.803	1.32	2.2

### 2.1.2 Contaminación de datos

Se realizó la contaminación de los datos, es decir que del 100% de los datos se contaminaron el 40%, con el objetivo de trabajar con esos datos para la validación. Con el número de datos de cada sensor que fueron 2185, se obtuvo un número de 874 que representan los datos de prueba y 1311 datos normales. Para esto, se realizó un código el cual se presenta en el Anexo 1. El total de los datos contaminados fueron 365.

A partir de la contaminación de los datos seleccionados anteriormente, del 40% está distribuido entre 4 sensores los cuales fueron:

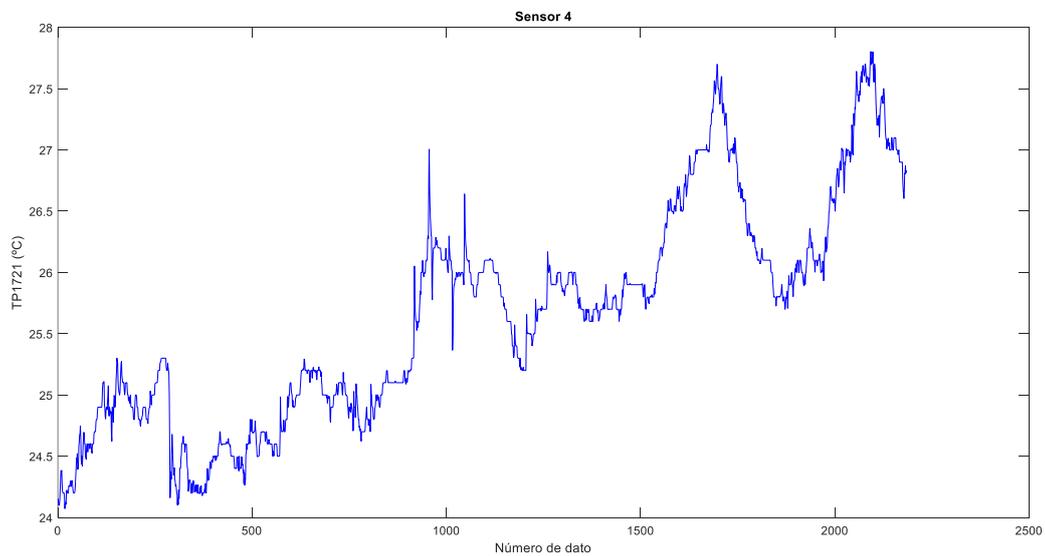
- Sensor #TP1721, sensor de temperatura.
- Sensor #RH1712, sensor de humedad.
- Sensor # BA1713, sensor de presión.
- Sensor # MM261, sensor de metano.

Cada uno de los sensores fue contaminado en rangos de datos (esto mencionará más adelante). Además, cada uno de estos sensores fueron sometidos a distintas fallas, para simular la contaminación que podrían tener en cada de los sensores en las condiciones que se encuentran. Cada una de las aplicaciones de las fallas se las realizó mediante un código que se muestra en el Anexo 2.

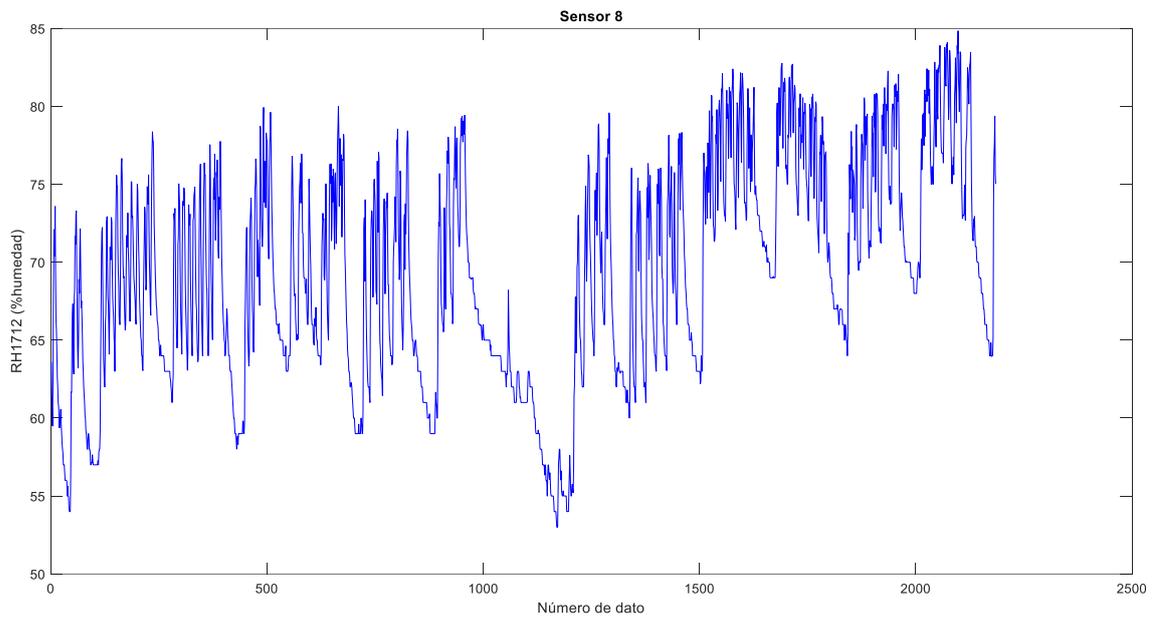
**Tabla 2.5 Parámetros de sensores seleccionado para la contaminación.**

Sensor	Mínimo	Máximo	Media	Desviación estándar	Mediana
TP1721	0	27.90	25.48	0.93	25.40
RH1712	0	86	68.69	7.268	69
BA1713	0	1130.90	1105.59	7.62	1105.30
MM261	0	30	0.049	0.13	0

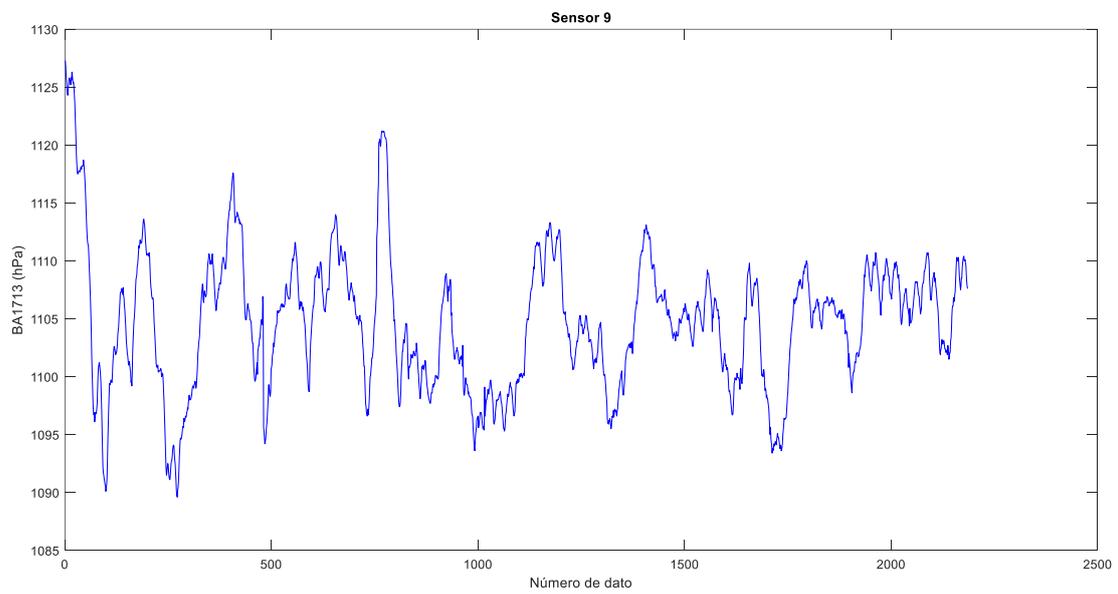
En las Figuras 2.4 a 2.7 se muestra la tendencia de los datos antes de la contaminación.



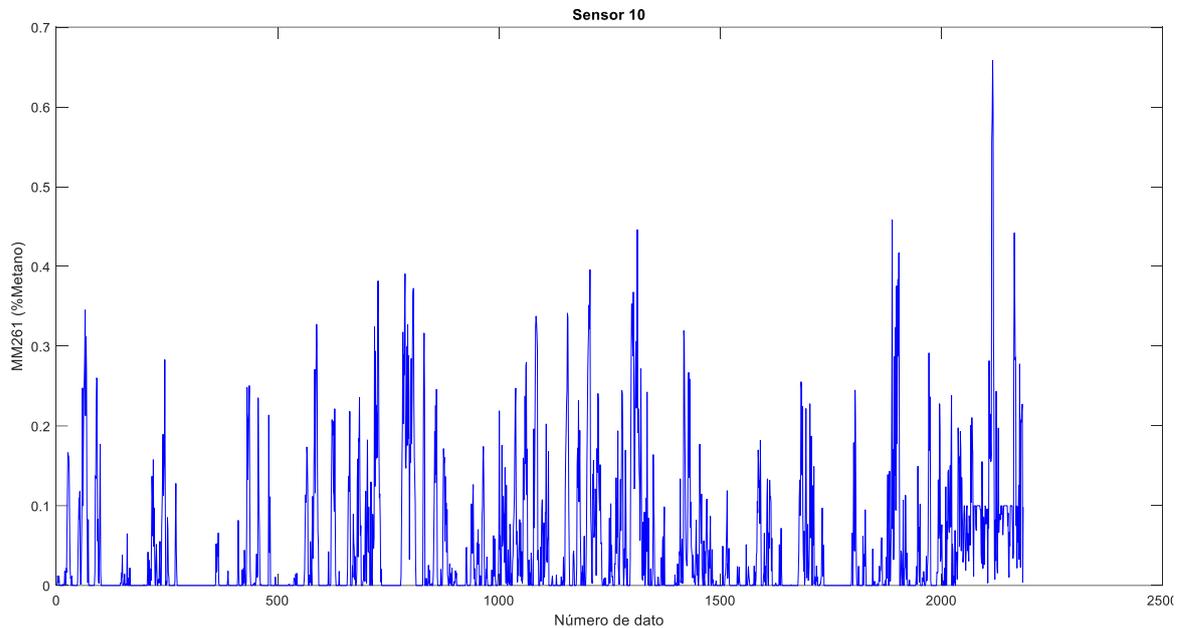
**Figura 2.4 Tendencia del sensor #4 TP1721.**



**Figura 2.5 Tendencia del sensor #8 RH1712.**



**Figura 2.6 Tendencia del sensor #9 BA1713.**



**Figura 2.7 Tendencia del sensor #10 MM261.**

### **2.1.2.1 Sensor TP1721**

En este sensor se realizó una falla completa, además el número de datos que se alteró en este sensor fue de 70 datos, los datos alterados fueron desde el dato 50 hasta el 120.

### **2.1.2.2 Sensor RH1712**

En este sensor se realizó una falla tipo drifting, es decir que desde el rango de datos que va a fallar mantienen la misma lectura, pero se le agrega 0.25 a cada valor. El número de datos que se alteró a este sensor fue de 75, los datos alterados fueron desde el dato 250 hasta 325.

### **2.1.2.3 Sensor BA1713**

Para este sensor se realizó una falla tipo degradación de la precisión, la cual nos dan valores aleatorios en los valores donde está fallando. En este sensor el número de datos que se alteró fue de 120, los datos alterados fueron desde el dato 430 hasta 550.

#### 2.1.2.4 Sensor MM261

Para este sensor se realizó una falla tipo Bias, es una falla tipo sesgo de magnitud 2 positivo que sube todos los valores en el rango de alteración, es decir los incrementa en 2 de magnitud. Se alteraron 100 datos para este sensor, los datos alterados fueron desde el dato 700 hasta el 800.

#### 2.1.3 Extracción de atributos

Se realizó la extracción de atributos mediante los métodos de Autoencoders, PCA, ICA y KPCA. Pero antes de aplicar la metodología se trabajó con datos normalizados. Se aplicó una normalización a los datos, pero de los datos no alterados para tener la información real.

Para la normalización se dice que  $X$  tiene una distribución normal estándar si  $\mu = 0$  y  $\sigma = 1$ , es decir media cero y desviación estándar 1 (Wasserman, 2003). La Ecuación 2.1 para la normalización se la presenta a continuación:

$$Z = \frac{X - \mu}{\sigma} \quad (2.1)$$

Donde  $Z$  es la variable aleatoria estándar,  $\mu$ , es la media de los datos y  $\sigma$  la desviación estándar de los datos.

La normalización de los datos se realizó mediante un código, el cual está especificado en el Anexo 3.

Una vez que se normalizaron los datos se procedió a aplicar cada uno de los métodos AE, PCA, KPCA y ICA para la extracción de atributos para la validación de los sensores. Todos los métodos permitieron obtener los coeficientes de los componentes principales de las variables utilizadas, que son conocidas como cargas de la matriz de datos. Para la aplicación de este método se toma un porcentaje de los datos, para este proyecto se usó un porcentaje del 85%.

En el caso de los AE esta extracción de atributos la metodología aplica un entrenamiento de los datos con el fin de encontrar la mejor reconstrucción de datos posibles, dicho entrenamiento tiene un rango de 10000 iteraciones. Las neuronas de entrada para el AE fueron de 18 y en la representación oculta se ocuparon 7, las cuales son los componentes principales del PCA.

## 2.1.4 SPE y KDE

Una vez se obtuvo los valores de los componentes principales, con estos datos se obtuvo el Squared prediction error (SPE), esta metodología se aplicó a la matriz de los componentes principales y para los sensores que están fallando, además permitió reconocer los límites que tiene los valores antes de ser considerados una falla. En el Anexo 4, se observa la elaboración del código para la aplicación de esta metodología.

Usando el valor del (SPE), se obtuvo el Kernel density estimator (KDE), el cual es una forma no paramétrica de estimación empleada en estadística, permite estimar la función de densidad de probabilidad de variables aleatorias (Wasserman, 2003). Para las variables mencionadas anteriormente se aplicó el KDE, para estimar la densidad de cada una de estas variables, con el fin de obtener el valor límite que deben tener los sensores para ser determinado como falla. El código de la elaboración del KDE, se encuentra en el Anexo 5.

### 2.1.4.1 Métricas de detección

Una vez realizada la detección y los límites, se procedió a realizar métricas de detección para evaluar cada uno de los métodos, para ello se los evaluó con la matriz de confusión, la cual permite la visualización del desempeño de un algoritmo de aprendizaje supervisado, en el que las columnas representan el número de predicciones del método y las filas los valores reales (Barrios, 2019). Además, que los parámetros de la matriz permitieron obtener las métricas, las cuales son la exactitud, precisión, sensibilidad y especificidad.



Figura 2.8 Matriz de confusión y sus componentes. Fuente: Barrios, 2019.

## 2.2 Etapa 2: Reconstrucción de datos

En la etapa de reconstrucción, mediante la reconstrucción para cada uno de los métodos se obtendrán los valores y las gráficas de la reconstrucción de cada uno de los métodos explicados en la extracción de atributos, que se explicó en la etapa anterior. Esta reconstrucción viene de la mano en el caso de los AE por el entrenamiento de la data en la etapa anterior. En este caso de la reconstrucción las neuronas de salida deben ser las mismas que la entrada, entonces las neuronas son 18. En el Anexo 6 se muestra el código de la reconstrucción.

### 2.2.1 Métricas de reconstrucción

Para la evaluación de la reconstrucción se realizaron las métricas de reconstrucción, para este caso en concreto se usó el estadístico del RMSE el cual es la desviación estándar de los residuos (errores de predicción), los residuos representan la medida de que tan lejos se encuentran los puntos de datos en la línea de regresión (Calasan et al., 2020). La Ecuación 2.2 representa el cálculo del RMS. Donde,  $N$  representa el número de datos medidos,  $x_i$  los datos medidos y  $\hat{x}_i$  los datos de la predicción o la reconstrucción.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (2.2)$$

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

### 3.1 Etapa 1: Detección de sensores

#### 3.1.1 Datos de entrada

La Tabla 2.4 muestra los datos recolectados por los sensores con los que se trabajó para la validación. Cabe recalcar que se trabajaron con 18 sensores.

#### 3.1.2 Contaminación de los datos

Se contaminó el 40% de los datos, esta contaminación se repartió entre 4 sensores, los cuales se explicó en el capítulo anterior. De los 18 sensores cada uno de estos tuvo 2185 datos de medida de sensor cada uno, se aplicó el 40% de los datos se obtuvo que cada sensor de la parte contaminada tiene 874 datos cada sensor.

Luego de la selección del 40% se procedió a la contaminación de este, el cual se repartió en 4 sensores en diferente rango y se aplicó diferentes tipos de fallas a cada uno de los sensores. Estas fallas se usaron para realizar todos los métodos.

#### **Sensor #TP1721, sensor de temperatura.**

Este sensor se contaminó los datos en el rango de 50 a 120, aplicando una falla completa, en la cual se reemplazó el valor de los datos originales por un valor de 20. A continuación se observa la primera falla en el sensor #4. En la Figura 3.1 se muestra el código de la falla.

```

%Falla Completa-- Variable (Sensor)#TP1721 TEMPERATURE--VARIABLE #4- Valor==>20
%Falla desde muestra: 50--120

for i=50:120
    X_test(i,4)=20;
end

```

**Figura 3.1 Falla completa al sensor #4, código.**

**Sensor #RH1712, sensor de humedad.**

Para este sensor #8 se elaboró una falla del tipo drifting, es decir que el primer valor tendrá un aumento de +0.25 en su medición y los siguientes tendrán el anterior +0.25, es decir +0.50 y así hasta el final. El rango de falla para este sensor fue del valor 250 al 325, este sensor tiene unidades de °C. En la Figura 3.2 se muestra el código de la falla.

```

% Frailure Drifting-- Sensor de humedad--RH1712--Variable #8
% Drifting magnitud .25°C--en el rango 250-325

d(1)=0.25;
for i=2:76
    d(i)=0.25+d(i-1);
end

X_test(250:325,8) = X_test(250:325,8)+d';

```

**Figura 3.2 Falla tipo drifting al sensor #8, código.**

**Sensor # BA1713, sensor de presión.**

Para el sensor #9 se aplicó una falla la cual es la degradación de la precisión, esta falla lo que hizo fue generar números aleatorios entre 0 y 1, los cuales se multiplicaron por 5 y se le agrega al valor de cada dato. Esta falla se aplicó desde el dato 430 al 550 del sensor. En la Figura 3.3 se muestra el código de la falla.

```

%Degradación de la precisión --Sensor BA1713--Variable #9 430--550
%mean_BA1713= mean(X_train(:,9));
for i=430:550
    X_test(i,9)=5*rand+X_test(i,9);
end

```

**Figura 3.3 Falla tipo Degradación de la precisión al sensor #9, código.**

#### **Sensor # MM261, sensor de metano.**

En el sensor #10 se aplicó una falla tipo Bias o tipo sesgo de magnitud 2 positivo que subió todos los valores en el rango de alteración, es decir los incrementó en 2 de magnitud los valores, para este sensor se trabajó con 100 datos, se aplicó la falla a los datos del 700 al 800. En la Figura 3.4 se muestra el código de la falla.

```

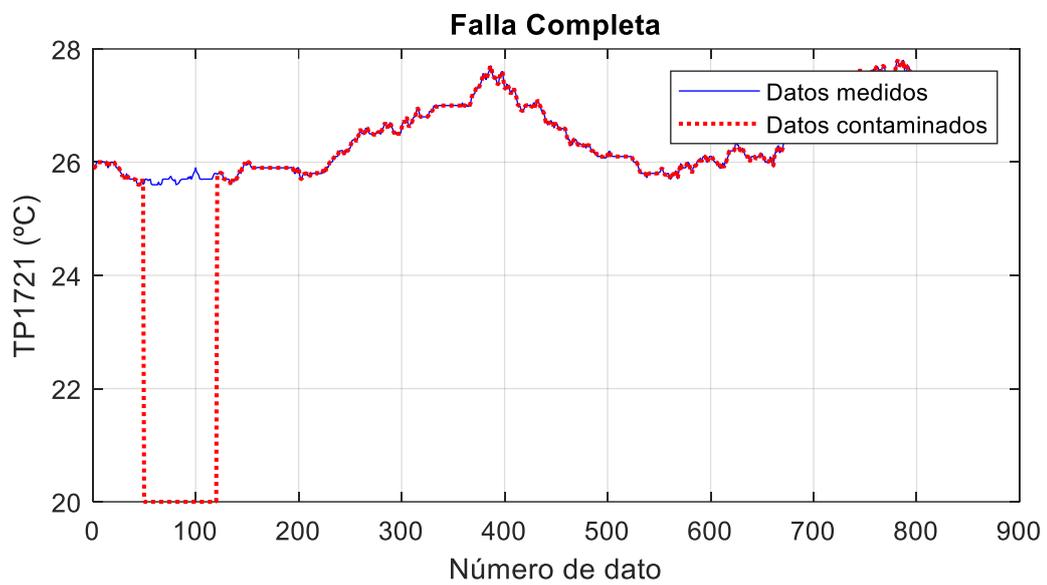
%Bias--Methane--Sensor MM261 700--800

mean_MM252=mean(X_train(:,10));
b=mean_MM252*2;
for i=700:800
    X_test(i,10) = X_test(i,10)+b;
end

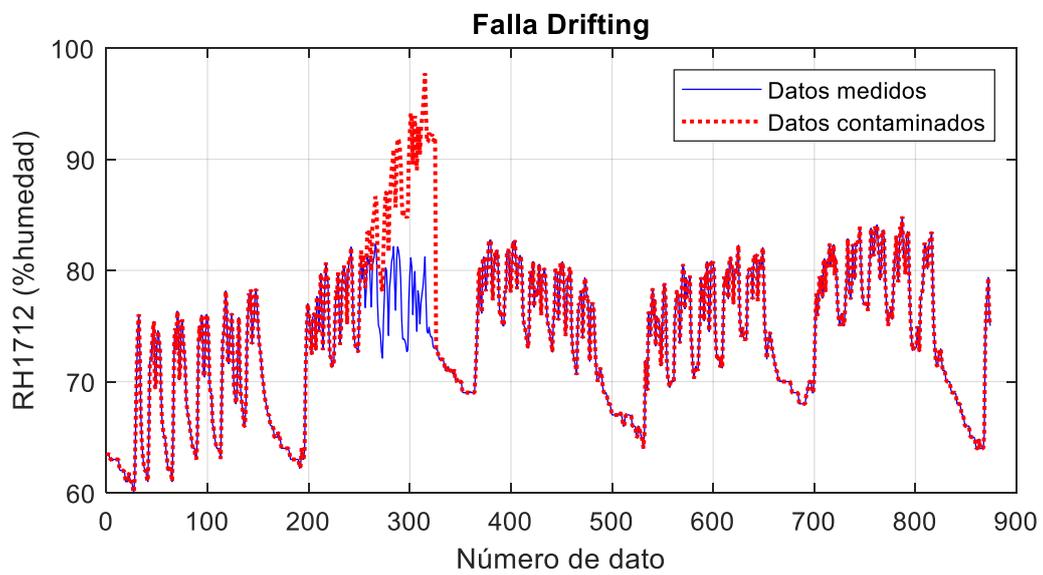
```

**Figura 3.4 Falla tipo Bias al sensor #10, código.**

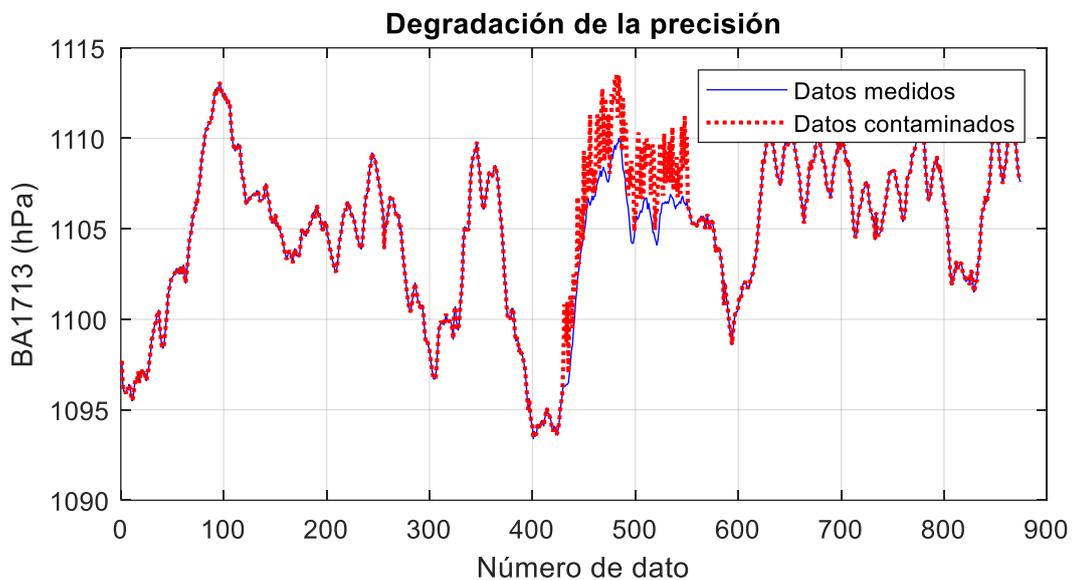
La aplicación de las fallas o contaminación de datos a cada uno de los sensores se puede ver gráficamente en las siguientes Figuras 3.5, 3.6, 3.7 y 3.8. Además, se pudo observar la aplicación de las fallas en los rangos establecidos anteriormente y se puede denotar el cambio en esos valores.



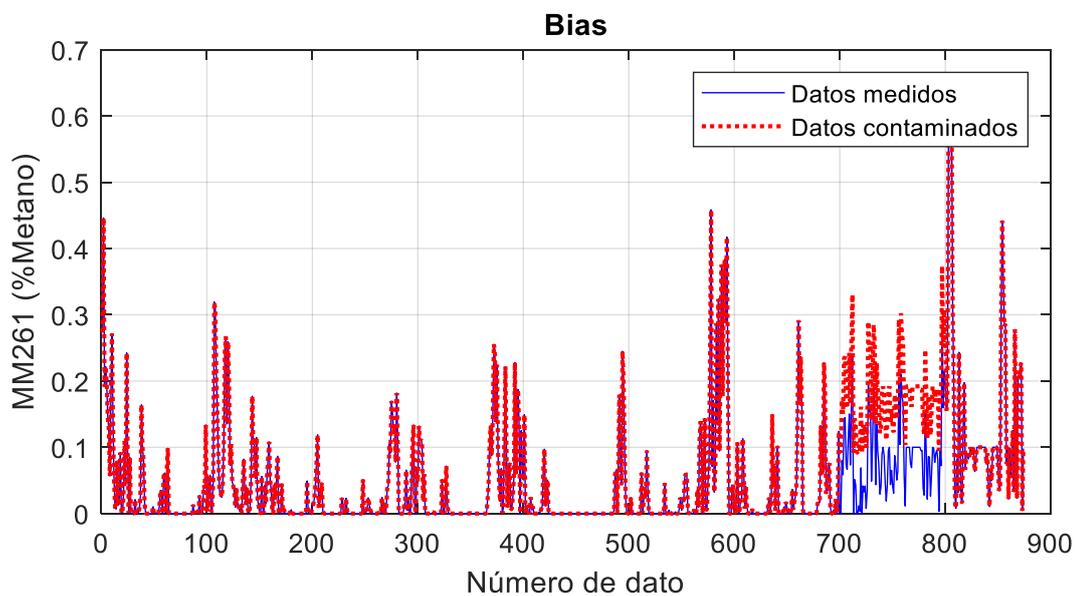
**Figura 3.5 Falla completa, sensor #4.**



**Figura 3.6 Falla tipo drifting, sensor #8.**



**Figura 3.7** Falla degradación de la precisión, sensor #9.



**Figura 3.8** Falla tipo Bias, sensor #10.

### 3.1.3 Extracción de atributos

Para esta fase se realizaron los diferentes métodos que mencionados anteriormente. Como primer punto se realizó la normalización de los datos.

## Normalización

Se aplicó la normalización para la aplicación del PCA, para la reducción de los datos. Además, se debe realizó esta normalización a los datos que no se les aplicó la falla, para la validación, la cual se realizó más adelante.

### 3.1.3.1 Extracción de atributos mediante los autoencoder y obtención de los componentes principales.

Para la obtención de los componentes principales, los cuales permitieron conocer las nuevas dimensiones con las que se trabajará en la reconstrucción de los datos. En el Anexo se encuentra el código con el que se realizó el entrenamiento de atributos para los AE.

Las dimensiones nuevas que se obtuvieron fue  $k$ , el cual tenía un valor de 7, es decir que las componentes principales sería 7 y además son la reducción de dimensiones mediante el método AE

#### 3.1.3.1.1 SPE Y KDE

Con los componentes principales obtenidos se procedió a utilizarlos para el SPE y a partir del SPE se obtiene el KDE, este valor da el límite para que los datos sean considerados una falla. En la Figura 3.9 se observa el SPE límite del método AE.

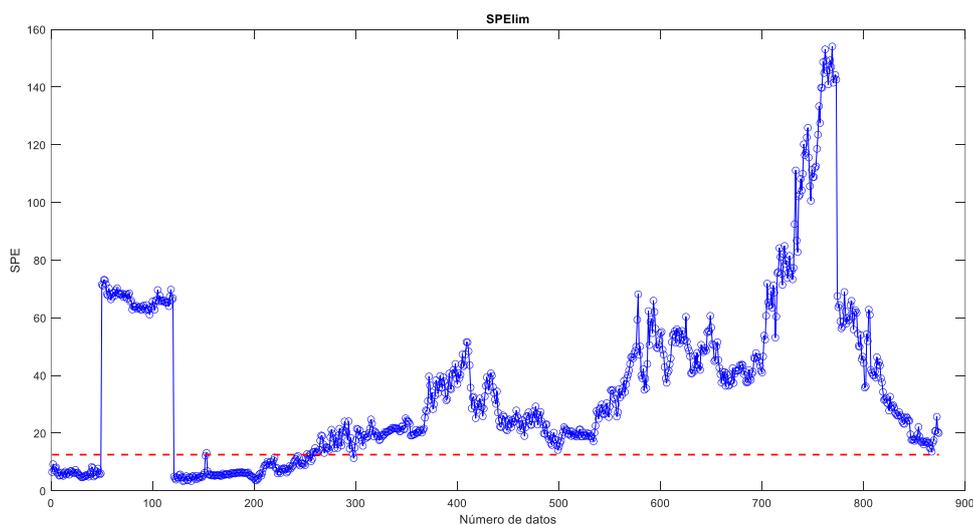


Figura 3.9 SPE límite del AE

El valor del SPE límite obtenido por el KDE para los AE, obtuvo un valor de 12.51. en la Tabla 3.1 se muestra los valores del SPE límite para cada uno de los métodos.

**Tabla 3.1 SPE límite de todos los métodos aplicados.**

<b>Método</b>	<b>SPE</b>
<i>Autoencoder</i>	12.51
<i>PCA</i>	5.01
<i>ICA</i>	$3.31 \times 10^6$
<i>KPCA</i>	6.04

### 3.1.3.1.2 Métricas de detección

Las Tablas 3.2, 3.3, 3.4 y 3.5 representan la matriz de confusión de cada uno de los métodos aplicados.

**Tabla 3.2 Matriz de confusión AE.**

<b>Matriz confusión Autoencoders</b>	
177	328
7	362

**Tabla 3.3 Matriz de confusión PCA.**

<b>Matriz confusión PCA</b>	
172	333
50	319

**Tabla 3.4 Matriz de confusión KPCA.**

<b>Matriz confusión KPCA</b>	
291	214
133	236

**Tabla 3.5 Matriz de confusión ICA.**

<b>Matriz confusión ICA</b>	
116	389
107	262

En la detección, el número de datos contaminados fueron de 365 y en el caso de los AE se obtuvo una detección de 362 casos contaminados, teniendo un porcentaje de acierto del 99% en los datos contaminados, seguido por el PCA que obtuvo 319 casos detectados y un 87%, el ICA con un número total de 262 y un 72%, y por último el KPCA con un número total del 236 y un 65%. Además, la Tabla 3.6 presenta las métricas de la matriz de confusión.

**Tabla 3.6 Métricas de la matriz de confusión.**

<b>MÉTODO</b>	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>Especificidad</b>
<i>PCA</i>	0.56	0.34	0.77	0.49
<i>ICA</i>	0.43	0.23	0.52	0.40
<i>KPCA</i>	0.60	0.58	0.69	0.52
<i>Autoencoders</i>	0.62	0.35	0.96	0.53

En la Tabla 3.6 se obtuvo resultados en la que se representa la exactitud, precisión, sensibilidad y especificidad de cada uno de los métodos, en lo que el AE es el método con más exactitud con 0.62, esta medición permite saber lo cerca que se encuentra el resultado del valor real, seguido por el PCA, KPCA e ICA. Para la precisión el método más exacto fue el ICA, seguido por PCA, AE y KPCA, la precisión nos permite

reconocer la dispersión del conjunto de datos a partir de las mediciones repetidas de una misma magnitud.

En la sensibilidad los AE obtuvieron el valor más alto con un 0,9, seguido por KPCA, PCA e ICA, esto representa la tasa de valores positivos detectados por el método. Y por último la especificidad, el valor más alto fue en el AE, seguido por KPCA, PCA e ICA, esto es la medición de la tasa de valores negativos que fueron detectado por el método.

### 3.2 Reconstrucción de datos

La reconstrucción nos permitió usar los datos contaminados y las variables estadísticas obtenidas para reconstruir los datos y hacerlos trabajar de forma estable.

En el caso de los AE esto se lo realizo mediante el entrenamiento previo para obtención de la mejor reconstrucción posible por el método. En las siguientes Figuras se presenta la reconstrucción de datos por el método de AE.

#### 3.2.1 AE Reconstrucción

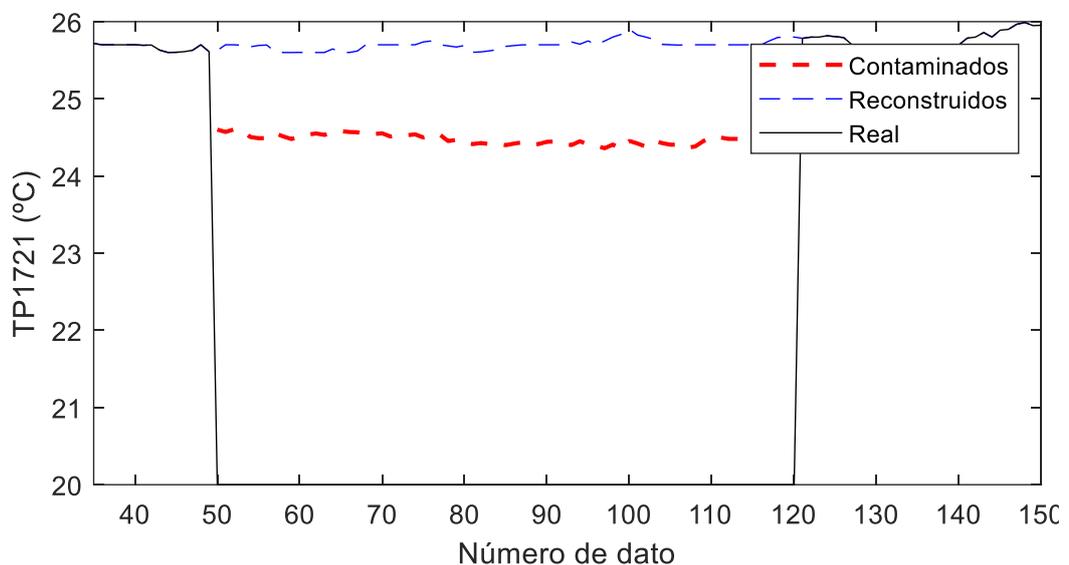
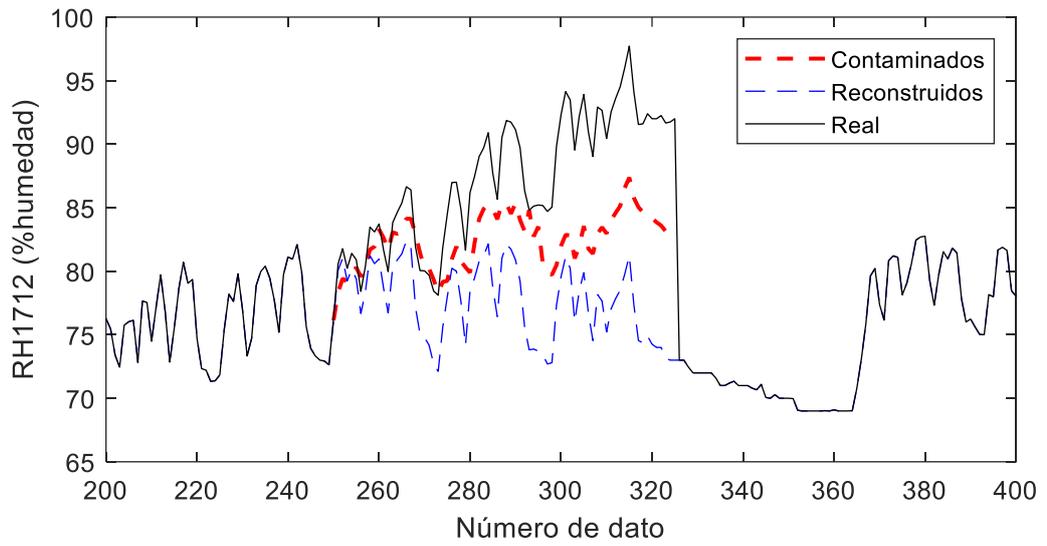
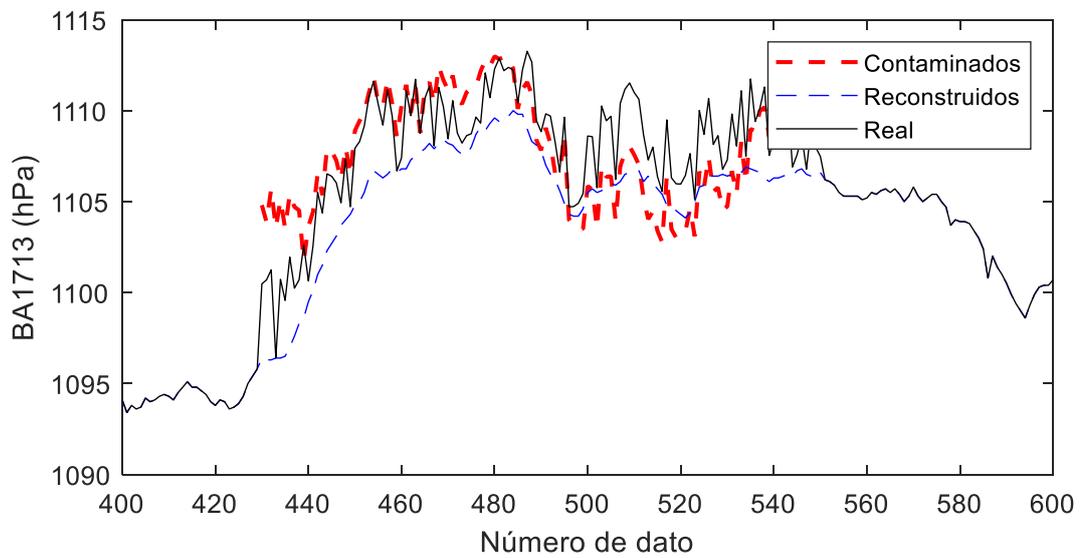


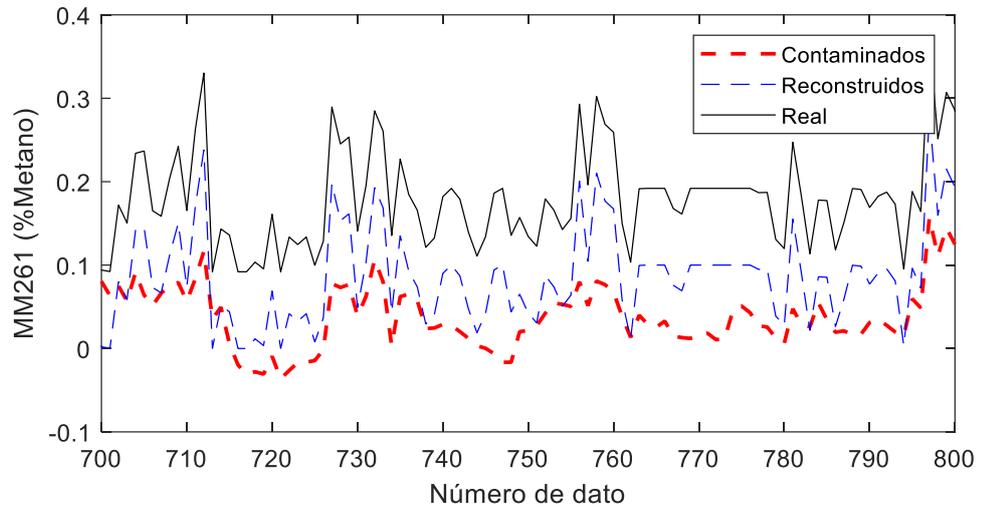
Figura 3.10 Reconstrucción AE, Sensor #4 TP1721



**Figura 3.11 Reconstrucción AE, Sensor #8 RH1712**



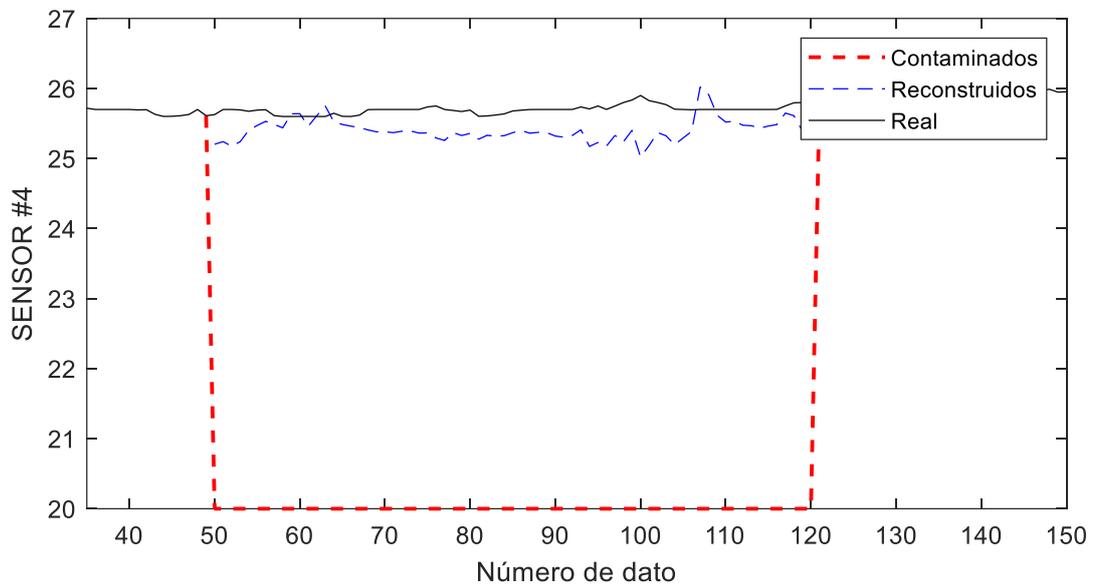
**Figura 3.12 Reconstrucción AE, Sensor #9 BA1713**



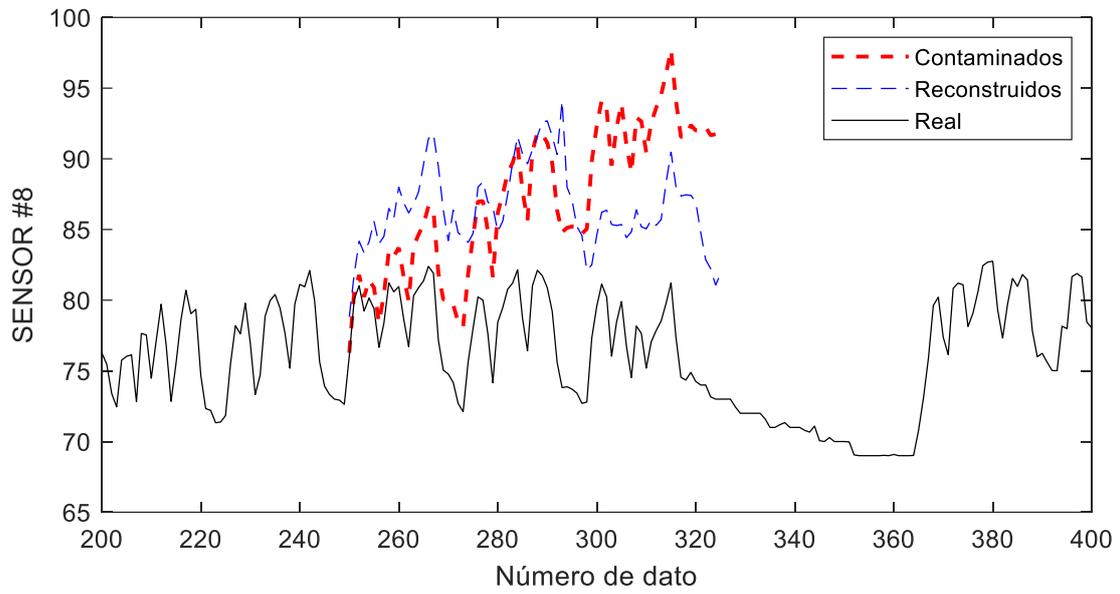
**Figura 3.13 Reconstrucción AE, Sensor #10 MM261**

### 3.2.2 PCA Reconstrucción

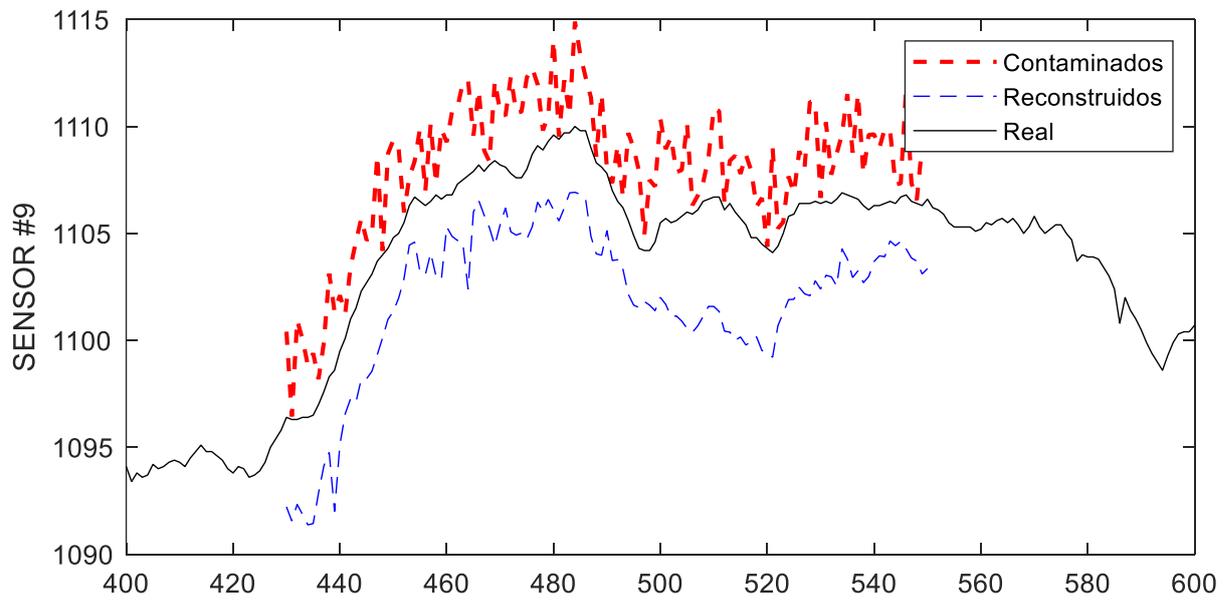
En las siguientes Figuras se presenta la reconstrucción de datos por el método de PCA.



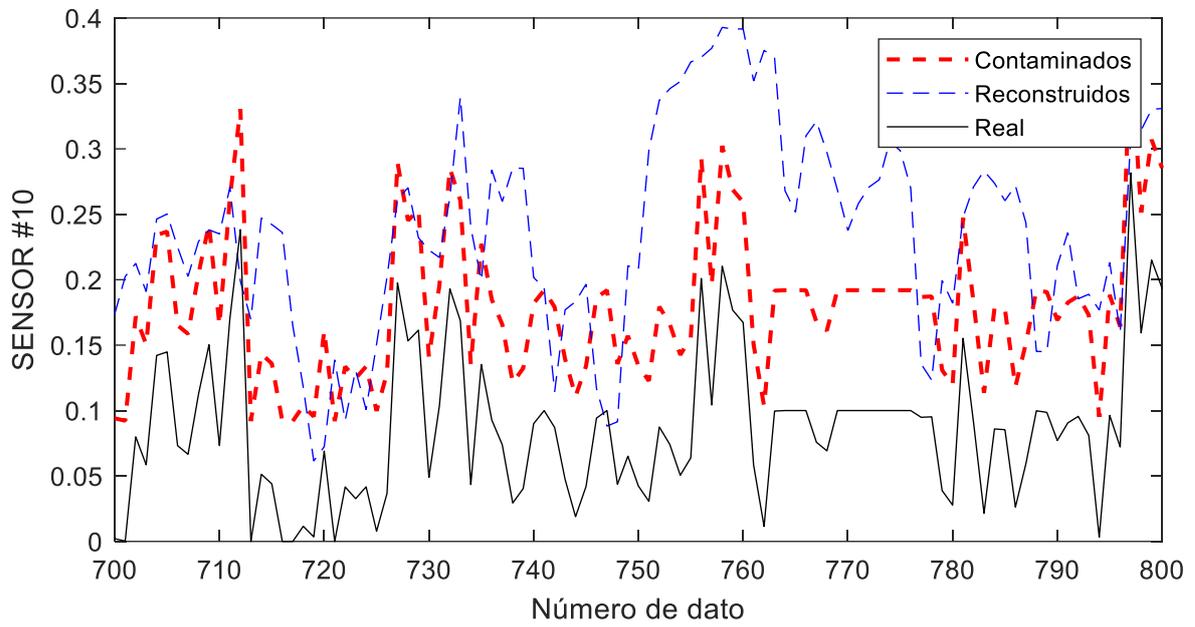
**Figura 3.14 Reconstrucción PCA, Sensor #4 TP1721**



**Figura 3.15 Reconstrucción PCA, Sensor #8 RH1712**

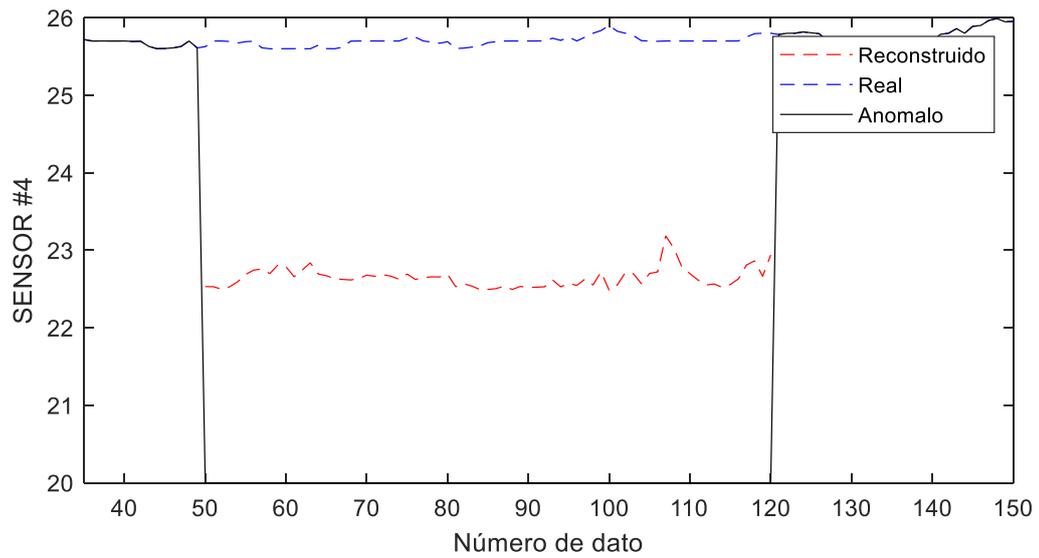


**Figura 3.16 Reconstrucción PCA, Sensor #9 BA1713**

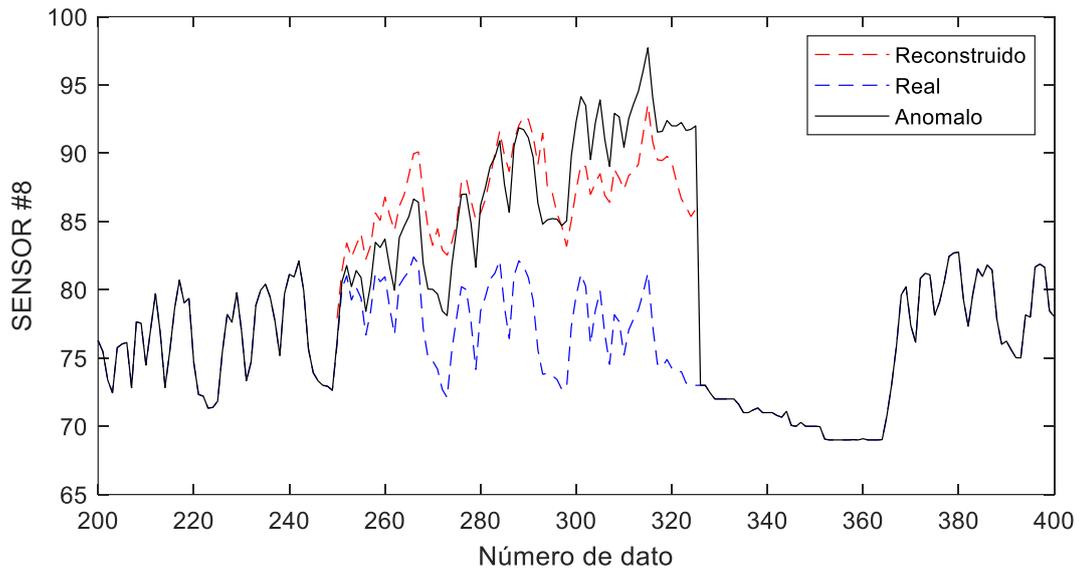


**Figura 3.17 Reconstrucción PCA, Sensor #10 MM261**

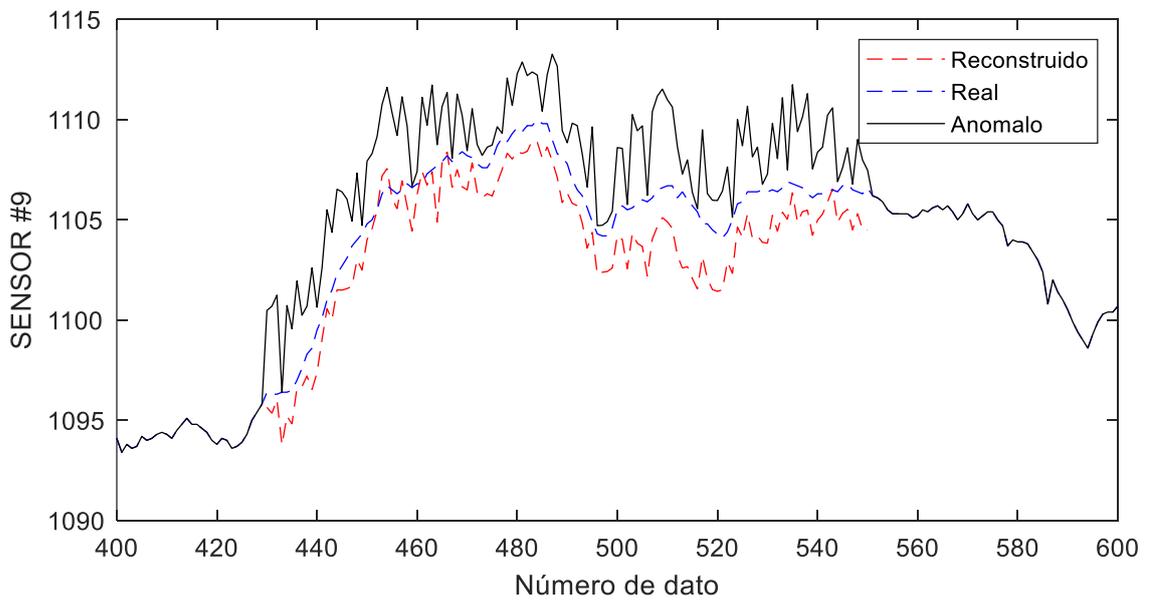
### 3.2.3 ICA Reconstrucción



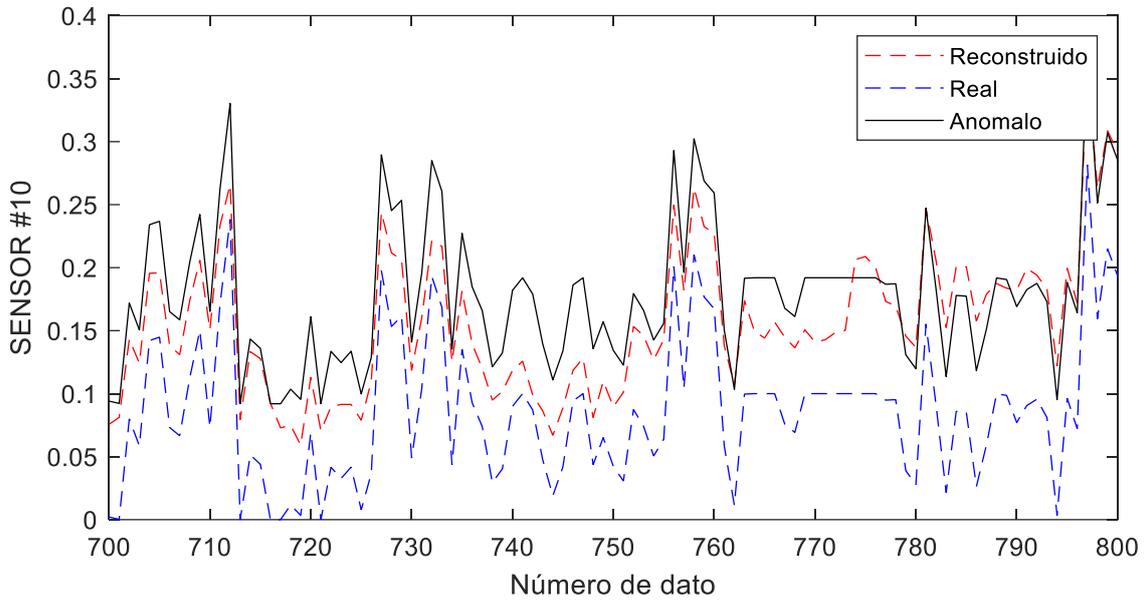
**Figura 3.18 Reconstrucción ICA, Sensor #4 TP1721**



**Figura 3.19 Reconstrucción ICA, Sensor #8 RH1712**

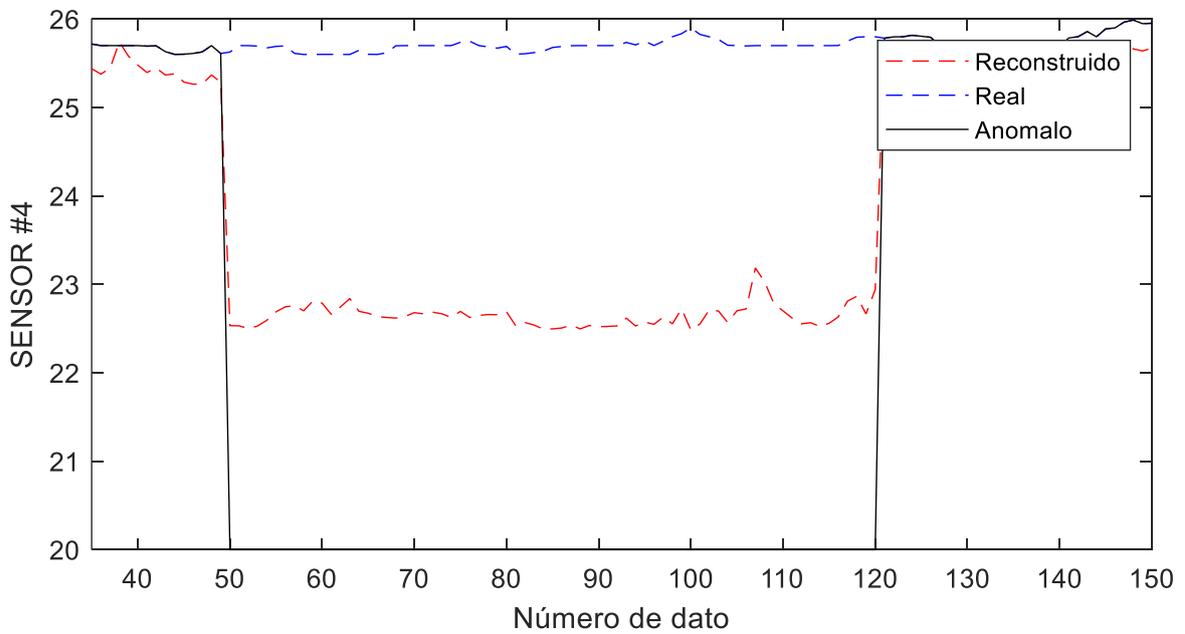


**Figura 3.20 Reconstrucción ICA, Sensor #9 BA1713**

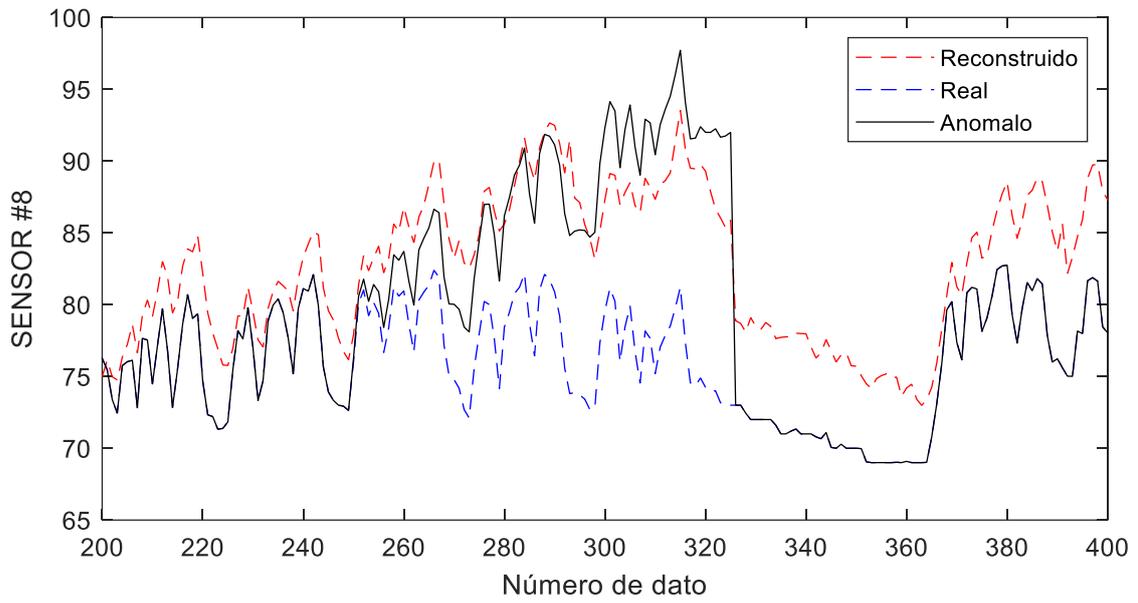


**Figura 3.21 Reconstrucción ICA, Sensor #10 MM261**

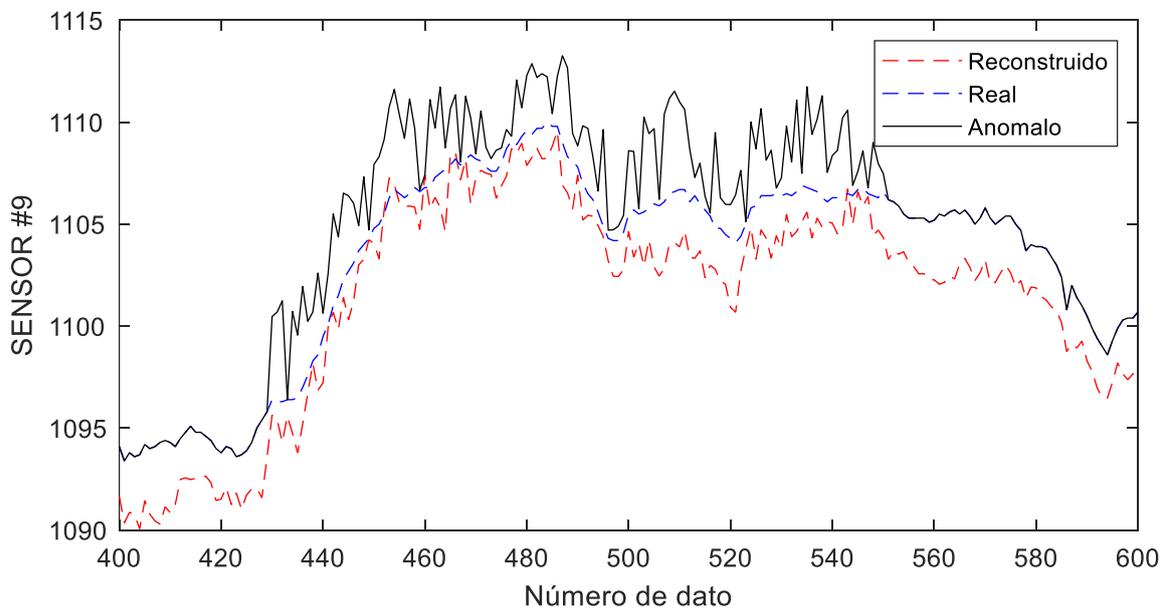
### 3.2.4 KPCA Reconstrucción



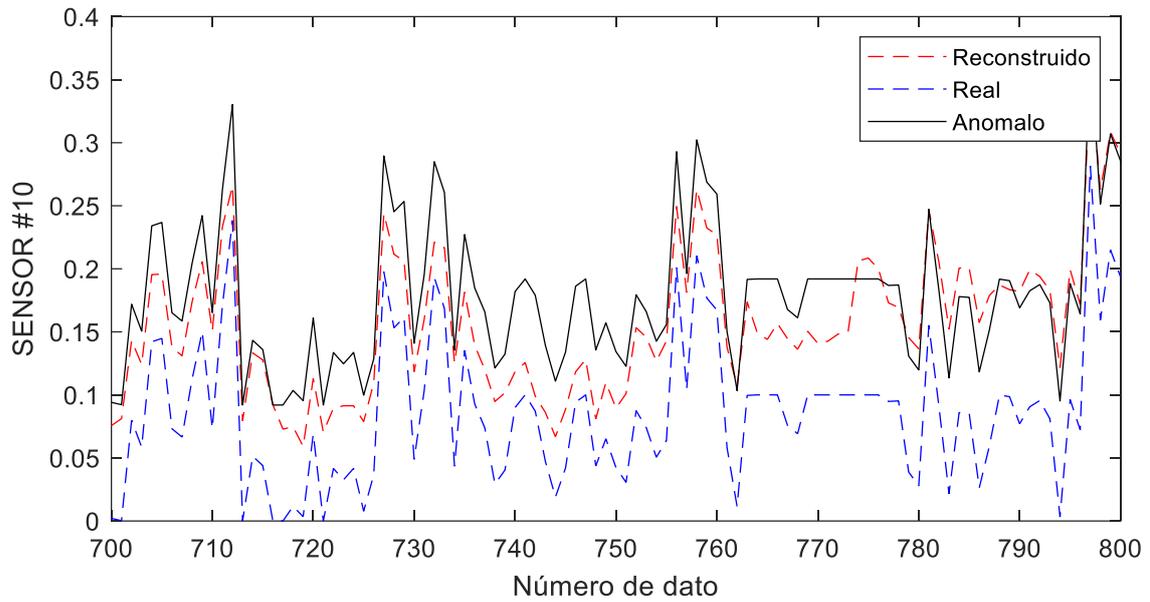
**Figura 3.22 Reconstrucción KPCA, Sensor #4 TP1721**



**Figura 3.23 Reconstrucción KPCA, Sensor #8 RH1712**



**Figura 3.24 Reconstrucción KPCA, Sensor #9 BA1713**



**Figura 3.25 Reconstrucción KPCA, Sensor #10 MM261**

### 3.2.5 Métricas de reconstrucción

Para las métricas de la reconstrucción se empleó el RMSE, para cada uno de los métodos. En la Tabla 3.7 se muestra las métricas de reconstrucción con cada uno de sus métodos. Se obtuvo las métricas de reconstrucción en las que el método de los AE presentó menos error con respecto a los diferentes métodos con un error de con un valor de 2.65 y una incertidumbre de  $\pm 2.54$ , esto permitió reconocer el error que hay entre los datos normales y la reconstrucción, siendo menor que los otros 3 métodos.

**Tabla 3.7 Métricas de reconstrucción.**

<b>Método</b>	<b>RMSE</b>
<i>PCA</i>	$3.45 \pm 4.36$
<i>ICA</i>	$3.73 \pm 4.40$
<i>KPCA</i>	$3.73 \pm 4.40$
<i>Autoencoders</i>	$2.65 \pm 2.54$

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Se realizó la validación de sensores para monitoreo de gas metano en minas de carbón, mediante la implementación de autoencoders no lineales. La aplicación del método de autoencoders no lineales arrojó un mejor desempeño con respecto a los métodos tradicionales como el PCA, KPCA e ICA. Además, este método permitió la reconstrucción de los datos a partir de la extracción de características no lineales.
- El método de AE obtuvo buenos resultados en las métricas de detección, ya que este fue el método más exacto, es decir que la reconstrucción estuvo más cerca de los otros métodos de los valores iniciales con un valor de 0.62, seguido por PCA = 0.56, KPCA = 0.60 e ICA = 0.63.
  - La precisión del método AE no fue la esperada, dando un valor de 0.35 en la dispersión, encontrándose en tercer lugar (por debajo del ICA = 0.23 y PCA = 0.34) y por último está el KPCA = 0.58; mientras menor sea la dispersión mayor es la precisión.
  - En la sensibilidad, los AE arrojaron valores más altos con un 0.96 (esto corresponde al número de datos no contaminados que pudo detectar el método), seguido por el PCA = 0.77, KPCA = 0.69 e ICA = 0.52.
  - La especificidad los AE fue la más alta con un valor de 0.53, es decir esto representa el número de datos contaminados que el método ha podido clasificar, seguido por el KPCA = 0.52, PCA = 0.49 e ICA = 0.40.
- En las métricas de la detección se obtuvo el número de datos contaminados de 365, los porcentajes de acierto que tuvieron los métodos para detectar la falla de un dato fueron:
  - Los AE de los 365 obtuvo una detección de 362 casos contaminados, teniendo un porcentaje de acierto del 99% en los datos contaminados.
  - El PCA que obtuvo 319 casos detectados, lo que representa un 87%.

- ICA con un número total de 262 representa un 72%.
  - Por último, el KPCA con un número total del 236 representa el 65%.
- Los métodos aplicados se evaluaron mediante métricas, en las que el autoencoder obtuvo menor error ( $2.65 \pm 2.54$ ), seguido del método PCA ( $3.45 \pm 4.36$ ). Los métodos ICA y KPCA presentaron un error de ( $3.73 \pm 4.40$ ) cada uno. Estos resultados demuestran que los métodos lineales arrojan un menor error que los métodos tradicionales o con funciones lineales.

## 4.2 Recomendaciones

- Aplicar el método con un rango más considerado de datos en la contaminación, para detectar la diferencia entre la reconstrucción en un rango más grande.
- Comparar a fondo los métodos PCA y AE, ya que sus resultados no fueron tan distanciados de lo esperado.

# BIBLIOGRAFÍA

- Abiri, N., Linse, B., Edén, P., & Ohlsson, M. (2019). Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems. *Neurocomputing*, 365, 137–146. <https://doi.org/10.1016/j.neucom.2019.07.065>
- Amat Rodrigo, J. (2017). *Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE*. <https://github.com/JoaquinAmatRodrigo/Estadistica-con-R>
- Barrios, J. (2019, July 26). *La matriz de confusión y sus métricas*. Healt Big Data. <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
- Ćalasan, M., Abdel Aleem, S. H. E., & Zobia, A. F. (2020). On the root mean square error (RMSE) calculation for parameter estimation of photovoltaic models: A novel exact analytical solution based on Lambert W function. *Energy Conversion and Management*, 210. <https://doi.org/10.1016/j.enconman.2020.112716>
- Global Methane Initiative. (2011). *Metano de las Minas de Carbón: Reducción de las Emisiones, Avance de las Oportunidades de Recuperación y Utilización*. <http://www2.ergweb>.
- Gómez, C., Garcés, K., Chavarriaga, J., Universidad de los Andes, & Alfonso, I. (2021). *Diseño y Construcción de una Red Inalámbrica de Sensores para el Monitoreo de Gases en Minas Subterráneas de Carbón*.
- Kozielski, M., Sikora, M., & Wróbel, Ł. (2021). Data on methane concentration collected by underground coal mine sensors. *Data in Brief*, 39, 112–133. <https://doi.org/10.17632/yd7vw4c5mk.1>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. <https://doi.org/10.1038/nature14539>
- Matsuo, Y., LeCun, Y., Sahani, M., Precup, D., Silver, D., Sugiyama, M., Uchibe, E., & Morimoto, J. (2022). Deep learning, reinforcement learning, and world models. *Neural Networks*, 152, 267–275. <https://doi.org/10.1016/j.neunet.2022.03.037>
- MINDAT. (2022, July 3). *Rydułtowy heap (Szarlota heap), ROW Ruch Rydułtowy Mine, Rydułtowy, Wodzisław County, Silesian Voivodeship, Poland*.

- Qin, S. J., & Li, W. (1999). *Detection, Identification, and Reconstruction of Faulty Sensors with Maximized Sensitivity*.
- Quispe Gutierrez mis Hermanos Karen, N. A., & Jean Pool Muriel Quispe, J. (2016). *APLICACIÓN DEL MÉTODO DE EXPLOTACIÓN LONG WALL EN LA VETA CABANA 3H, NIVEL 2770, MINERA AURIFERA RETAMAS S.A. PATAZ LA LIBERTAD 2016*.
- Romero Acero, Á., Marín Cano, A., Jiménez Builes, J., & Universidad Nacional de Colombia. (2013). *Red de sensores inalámbricos para el monitoreo de alertas tempranas en minas subterráneas: una solución a la problemática de atmósferas explosivas en la minería de carbón en Colombia*.
- Wasserman, L. (2003). All of Statistics A Concise Course in Statistical Inference by Larry Wasserman. *Springer*, 20–45.
- Yan, W., Guo, P., Gong, L., & Li, Z. (2016). Nonlinear and robust statistical process monitoring based on variant autoencoders. *Chemometrics and Intelligent Laboratory Systems*, 158, 31–40. <https://doi.org/10.1016/j.chemolab.2016.08.007>
- Yoo, C. K., Villez, K., Lee, I. B., van Hulle, S., & Vanrolleghem, P. A. (2006). Sensor validation and reconciliation for a partial nitrification process. *Water Science and Technology*, 53(4–5), 513–521. <https://doi.org/10.2166/wst.2006.155>

## 5. ANEXOS

### 5.1 Anexo 1: script

```
1
2 clear;
3 clc;
4 close all;
5
6 load methanedata.mat
7
8
9 %%
10 data=methanedatafinal; %Nueva variable de los datos
11 data=data(:,7:24);
12
13 L=length(data);
14
15 n_train=round(L*0.60);
16 n_test=L-n_train;
17
18 %%
19 X_train=data(1:n_train,:); %Datos en condiciones normales (60%)
20
21 X_test=data(n_train+1:end,:);%Datos contaminados (40%)
22 XT=X_test;
23
24
```

## 5.2 Anexo 2

```
1 %% CONTAMINACIÓN DE DATOS
2
3 %Falla Completa-- Variable (Sensor)#TP1721 TEMPERATURE--VARIABLE #4-
4 Valor==>20
5 %Falla desde muestra: 50--120
6
7 for i=50:120
8     X_test(i,4)=20;
9 end
10
11
12 % Frailure Drifting-- Sensor de humedad--RH1712--Variable #8
13 % Drifting magnitud .25°C--en el rango 250-325
14
15 d(1)=0.25;
16
17 for i=2:76
18     d(i)=0.25+d(i-1);
19 end
20
21 X_test(250:325,8) = X_test(250:325,8)+d';
22
23
24
25 %Degradación de la precisión --Sensor BA1713--Variable #9 430--550
26 %mean_BA1713= mean(X_train(:,9));
27
28 for i=430:550
29     X_test(i,9)=5*rand+X_test(i,9);
30 end
31
32 %Bias--Methane--Sensor MM261 700--800
33
34
35 mean_MM252=mean(X_train(:,10));
36 b=mean_MM252*2;
37
38 for i=700:800
39     X_test(i,10) = X_test(i,10)+b;
40 end
```

## 5.3 ANEXO 3

```
1 %% NORMALIZACIÓN
2 %Parámetros para desnormalizar
3 m= mean(X_train); %MEDIA
4 s=std(X_train); %DESVIACIÓN ESTANDAR
5
6 for i=1:n_train
7     for j=1:18
8         X(i,j)=(X_train(i,j)- m(j))/s(j);
9     end
10 end
11
12 for i=1:length(X_test)
13     for j=1:18
14         X_faulty(i,j)=(X_test(i,j)-m(j))/s(j);
15     end
16 end
17
18 XT=X_faulty;
```

## 5.4 ANEXO 4

```
1 %k=COMPONENTES PRINCIPALES-----K=7
2 %%
3 P=COEFF(:,1:k); %P>Loading vector
4 for i=1:size(X,1)
5     t2(i)=X(i,:)*P*inv(diag(LATENT(1:k)))*P'*X(i,:); %HOTELLING
6     SPE(i)=X(i,:)*COEFF*(X(i,:)*COEFF)'-X(i,:)*P*(X(i,:)*P)'; %SQUARED
7     PREDICTION ERROR
8 end
9
10 %Cálculo del SPE y t2 PARA SENSORES CON FALLAS
11 for i=1:size(XT,1);
12     XTt2(i)=XT(i,:)*P*inv(diag(LATENT(1:k)))*P'*XT(i,:);
13     XTSPE(i)=XT(i,:)*COEFF*(XT(i,:)*COEFF)'-XT(i,:)*P*(XT(i,:)*P)';
14 end
```

## 5.5 ANEXO 5

```
1 %% KERNEL DENSITY ESTIMATOR (Function --kde.m)
2 %Se lo hace en los T2/SPE de X (TRAINING DATA)
3
4 [bandwidth,density,xmesh,cdf]= kde(t2);
5 r=0.95;
6 for i=1:size(cdf,1),
7     if cdf(i,1)>=r,
8         break;
9     end;
10 end;
11
12 T2limit=xmesh(i);
13
14 [bandwidth,density,xmesh,cdf]=kde(SPE);
15 r=0.95;
16 for i=1:size(cdf,1),
17     if cdf(i,1)>=r,
18         break;
19     end;
20 end;
21
22 SPElimit=xmesh(i);
```

## 5.6 ANEXO 6

```
1 ##### First network - AE (3 layers) #####
2 input_dim = X_train.shape[1] #18
3 encoding_dim = 7 #Let's select 7... obeying PCA
4 #~ red neuronal --> autoencoder
5
6 def build_autoencoder():
7     input_img = Input(shape=(input_dim,))
8
9     encoded = Dense(encoding_dim,activation='elu')(input_img)
10
11    decoded = Dense(input_dim, activation = 'linear')(encoded)
12
13    autoencoder = Model(input_img, decoded)
14
15    autoencoder.compile(optimizer='adam',loss='mse')
16
17    print(autoencoder.summary())
18    return autoencoder
19
20 if Train == 1:
21     autoencoder = build_autoencoder()
22     history = autoencoder.fit(X_train,X_train,
23                             epochs = 10000,
24                             batch_size = 16,
25                             shuffle = False,
26                             validation_split = 0.30,
27                             verbose = 1,
28                             callbacks = my_callbacks)
29     autoencoder = load_model("./models/AE.h5")
30     # Plotting loss
31     plt.figure(dpi=250)
32     plt.plot(history.history['loss'],'b')
33     plt.plot(history.history['val_loss'],'r--')
34     plt.title('Model training vs Validation Loss')
35     plt.ylabel('Loss')
36     plt.xlabel('epoch')
37     plt.legend(['Training', 'Val_loss'])
38     plt.show()
39
40 elif Train == 0:
41     autoencoder = load_model("./models/AE.h5")
42
43
44 X_reconstruction = autoencoder.predict(X_test)
45 X_train_rec = autoencoder.predict(X_train)
46
47 #EXPORTA DATOS A .MAT
48 sio.savemat("./Reconstruction_AE.mat", {"Reconstruccion_AE":
49 X_reconstruction,
50                                     "Reconstruccion_AE_train":
51 X_train_rec})
```