UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

.

DATA SCIENCE APLICADO EN LA INDUSTRIA ACERERA

POR:

ING. MARCO ANTONIO ARCIBA MELÉNDEZ

TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERÍA EN COMPUTACIÓN

CHIHUAHUA, CHIH., MÉXICO **09/2022**

Data science aplicado a la industria acerera. Tesis presentada por Marco Antonio Arciba Meléndez como requisito parcial para obtener el grado de Maestro en Ingeniería en Computación, ha sido aprobada y aceptada por:

M.I. Fabián Vinicio Hernández Martínez
Director de la Facultad de Ingeniería

Dr. Fernando Martínez Reyes
Secretario de Investigación y Posgrado

M.S.I. Karina Rocío Requena Yáñez
Coordinador(a) Académico

Dr. Luis Carlos González Gurrola
Director(a) de Tesis

Fecha
24 de Enero del 2023

Comité:
DR. LUIS CARLOS GONZÁLEZ GURROLA
DR. RAFAEL CAMILO LOZOYA GAMEZ
DR. ALAIN MANZO MARTÍNEZ
DR. RAYMUNDO CORNEJO GARCÍA

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

20 de enero de 2023.

ING. MARCO ANTONIO ARCIBA MELÉNDEZ
Presente.

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Ingeniería en Computación, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Luis Carlos González Gurrola** para que lo desarrolle como tesis, con el título **"DATA SCIENCE APLICADO A LA INDUSTRIA ACERERA"**.

Índice de Contenido

Agradecimientos

Resumen

Índice de contenido

Índice de tablas

Índice de figuras

Índice de Ilustraciones

Capítulo 1. Introducción
- 1.1. Motivación
- 1.2. Enfoque
- 1.3. Justificación
- 1.4. Hipótesis
- 1.5. Objetivo

Capítulo 2. Marco teórico
- 2.1 Antecedentes
- 2.2 Horno de fundición por arco voltaico
- 2.3 Machine learning (ML)
- 2.4 Series de tiempo
- 2.5 Detección de anomalías

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

2.6 segmentación para series de tiempo y tiempo real

Capítulo 3. Metodología

3.1 Preparación de los datos

3.2 Aplicación sobre los métodos/algoritmos

Capítulo 4. Resultados

4.1 Isolation forest

4.2 Autoencoders

4.3 Programación dinámica (PD)

Capítulo 5. Discusión y conclusiones

5.1 Método de ML

5.2 Conclusión

5.3 Trabajo a futuro

Referencias

Glosario

Apéndice

Currículum vitae

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las investigaciones.

A T E N T A M E N T E

"Naturam subiecit aliis"

FACULTAD DE INGENIERÍA U.A.CH.

EL DIRECTOR

M.I. FABIÁN VINICIO HERNÁNDEZ MARTÍNEZ

DIRECCIÓN

SECRETARIO DE INVESTIGACIÓN Y POSGRADO

DR. FERNANDO MARTÍNEZ REYES

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

Dedicatoria

Lleno de regocijo, satisfacción y esperanza, dedico este proyecto a mis seres queridos.

A mi querida hija Madelyn, para motivarla a siempre buscar y lograr las metas que se proponga.

A mis padres, Mario Arciba y María Luis Meléndez por darme la confianza y apoyo incondicional para poder lograr todo lo que hasta hoy he logrado, así como su educación y valores que me inculcaron desde pequeño.

A mis hermanos César y Rocío, por siempre estar ahí apoyando y motivando a seguir adelante.

Agradecimientos

Primeramente agradezco a Dios por haberme dado la oportunidad de haber concluido este proyecto y haberme dado las fuerzas para superar los obstáculos que a lo largo de esta investigación se fueron presentando.

Deseo agradecer a todas las personas que de una u otra forma permitieron la realización de este trabajo de investigación, en especial a mi hija Madelyn, por su amor, cariño y paciencia en los momentos de arduo trabajo, mi pequeña hija que es mi motor de vida y lo que mas quiero en la vida, gracias.

A mis padres, Mario Arciba y María Luisa Meléndez, por infinito cariño, paciencia, comprensión y apoyo desde siempre, por haberme educado y amado siempre. Quienes sin escatimar el esfuerzo han sacrificado gran parte de su vida por mi y me han formado y educado para ser un hombre de bien, a ellos mil gracias.

De igual manera a mi asesor de tesis Luis Carlos González Gurrola que me brindó su apoyo incondicional durante todo el desarrollo de la investigación.

A todos mis maestros que a lo largo de mis estudios de maestría aportaron sus conocimientos invaluables, sugerencias y apoyo.

A la empresa AMI international que hizo que este proyecto fuera posible, dándome la confianza y brindándome las herramientas necesarias para realizar la investigación, y brindándome toda su experiencia en el tema, ¡muchas gracias!

Resumen

Este trabajo de tesis está enfocado en la aplicación de un proceso de *data science* dentro de la industria acerera, se lleva a cabo la exploración de métodos de *machine learning*, métodos de segmentación y su implementación en tiempo real, todo esto para poder mantener, en un momento dado, un estado saludable al transformador que suministra energía eléctrica a los hornos de fundición de acero por arco voltaico, reduciendo riesgos y mejorando rendimientos de todos los componentes de dicho horno.

El mayor tiempo que se llevo a cabo fue en la preparación de los datos, ya que eran datos crudos directamente extraídos de las mediciones sin preprocesar, conforme se aplicaba un método se hacía una evaluación para seleccionar si la información con la que estábamos alimentando al modelo era la correcta, por lo que se hicieron varios pasos, como eliminar variables, quitar valores nulos, y tomar máximos por día; por lo que se toma bastante tiempo en adecuar la información, debido a que si alimentamos nuestros modelos con información que no sirve los resultados obviamente no servirían de nada.

La segunda para que consumió la mayor parte del tiempo fue la experimentación, donde iniciamos aplicando métodos de detección de anomalías como lo son *Isolation Forest* y *Autoencoders,* esta experimentación resulta muy interesante ya que los modelos se comportan de manera diferente con datos históricos a con datos en tiempo real y entendimos que alcance y utilidad pueden tener estos métodos; hasta llegar al método que utilizamos para resolver uno de nuestros objetivos.

El método que nos funcionó para segmentar (que es el objetivo inicial) fue programación dinámica, este algoritmo ya se ha utilizado antes, pero, en los documentos e investigaciones que leí, nunca de la manera que se implementó en este proyecto, por lo que se desarrolló para que operara en tiempo real, con una precisión bastante alta por encima del 85%.

Uno de los problemas a resolver de este método es el consumo de tiempo para resolver problemas, a medida que la base de datos crece el método se vuelve más lento, por lo que se realizó un *batch,* que funcionara de manera automática, el cual se describe en este trabajo.

A PARTIR DEL ***ÍNDICE*** PONER EL LOGO DE LA FACULTAD EN LA PARTE SUPERIOR IZQUIERDA

Índice de Contenido

Agradecimientos 3

Resumen 4

Índice de Contenido 5

Índice de Tablas 8

Índice de Figuras 9

Capítulo 1. Introducción 10

1.1 Motivación 14

1.2 Enfoque 15

1.3 Justificación 16

1.4 Hipótesis 17

1.5 Objetivo 17

1.5.1 General 17

1.5.2 Específicos 17

Capítulo 2: Marco Teórico 18

2.1 Antecedentes 18

2.2 Horno de fundición por arco voltaico 19

2.2.1 ¿Qué son los hornos de fundición? 19

2.2.2 Funcionamiento de los hornos de arco eléctrico 20

2.2.3 Diferentes tipos de hornos 20

2.3.4 Los componentes del arco voltaico 21

2.4.5 Elaboración del Metal 21

2.5.6 Inconvenientes del horno de arco 21

2.3 Machine Learning (ML) 22

2.3.1 Aprendizaje supervisado 23

2.3.2 Aprendizaje no supervisado 23

2.3.3 Aprendizaje de refuerzo 24

2.4 Series de tiempo 24

2.5 Detección de anomalías 25
2.6 Segmentación para series de tiempo y tiempo real 26

Capítulo 3: Metodología 27
3.1 Preparación de los datos 28
3.2 Aplicación sobre los métodos/algoritmos: 29
3.2.1 Isolation forest 30
3.2.1.1 ¿Qué es? 30
3.2.1.2 ¿Cómo funciona? 32
3.2.1.3 Parámetros 33
3.2.1.4 Funcionamiento 34
3.2.2 Autoencoders 34
3.2.2.1 Parámetros 1 variable 36
3.2.2.2 Parámetros multivariable 38
3.2.3 Programación Dinámica (PD) 39
3.2.3.1 Características de la programación dinámica 40
3.2.3.2 Parámetros 1ra fase 40
3.2.3.3 Funcionamiento 1ra fase 41
3.2.3.4 Parámetros 2da fase (tiempo real) 42
3.2.3.5 Funcionamiento 2da fase 42

Capítulo 4: Resultados 43
4.1 Isolation forest 43
4.2 Autoencoders 45
4.2.1 1 variable 45
4.2.2 Multivariable: 46
4.3 Programación Dinámica (PD) 47
4.3.1 Primera fase (tiempo histórico): 47
4.3.2 segunda fase (tiempo real): 48
4.3.3 Resultados PD 49

Capítulo 5: Discusión y conclusiones 50
5.1 Métodos de ML 51
5.2 Conclusión 51
5.3 Trabajo a futuro 52

Referencias 53

Glosario 55

Curriculum Vitae 56

Índice de Tablas

Tabla 1.1: Base de datos. 29
Tabla 1.2: Librerías importadas para utilizar el método Isolation Forest en Python..... 33
Tabla 1.3: Librerías importadas y parámetros para el método Autoencoders en Python. 36
Tabla 1.4: Librerías importadas para utilizar el método Isolation forest..... 38
Tabla 1.5: Parámetro Programación Dinámica..... 41
Tabla 1.6: Score mediante PD en tiempo real. 49

Índice de Figuras

Figura 1.1 Puente Millau 10
Figura 1.2: Horno por arco voltaico. (García, 2021) 12
Figura 2.1: Serie de tiempo. 25
Figura 2.2: Representación de valor anómalo. (Tomado de Intro to anomaly detection with OpenCV, Computer Vision, and scikit-learn - PyImageSearch) 26
Figura 2.3: Ejemplo de segmentación 27
Figura 3.1: Segmentación esperada. 30
Figura 3.2: Observación difícil de aislar. 31
Figura 3.3: Observación fácil de aislar. 32
Figura 4.1: Isolation Forest. 44
Figura 4.2: *Autoencoders* en Monóxido de Carbono. 45
Figura 4.3: *Autoencoders* en multivariado (6 gases). 46

Capítulo 1. Introducción

La industria siempre ha tenido avances importantes y de grandes aportes en tema de utilización de la ciencia para mejorar procesos, mejorar la eficiencia de los recursos y generar mejores resultados en todos los aspectos.

La industria acerera es de gran importancia a nivel mundial en el año 2019 la producción mundial de acero alcanzó 1869 millones de toneladas, con un aumento del 3.4% respecto del año 2018 y México quedó en el 15vo. Lugar como productor de este material a nivel mundial según la *World Steel Association.* (Bardahl Industria Blog, 2020)

Por mencionar algunas de las principales aplicaciones podemos observar las construcciones de grandes puentes como el puente de Millau, en Francia, que observamos en la figura 1.1, instrumental médico, edificaciones, y en la industria automotriz entre otros.

Figura 1.1 Puente Millau (Los 10 puentes más famosos del mundo, 2022)

En México la industria del acero representa el 2% del Producto Interno Bruto (PIB), el 6.9% del PIB industrial y el 12.9% del PIB manufacturero, por lo que es de suma importancia no exista disminución en la producción de este metal, ya que afectaría

directamente el crecimiento económico del país. En el año 2019 México reportó 18.6 millones de toneladas y esperan que para el año 2020 suba un 1.7%. (Bardahl Industria Blog, 2020)

Esta actividad es importante para otras actividades, que son imprescindibles en la construcción y modernidad del país, desde ferrocarriles hasta los más grandes rascacielos, vehículos automotores, aviones, instrumentos médicos, entre otros. (Millán, 2014)

El acero es clave y estratégico para el desarrollo económico del país, por ser el insumo básico de diferentes sectores industriales de alta importancia relativa en la economía mexicana; como el automotriz, la industria eléctrica y electrónica, entre otras, pero también es de los principales consumidores de la minería de metales metálicos, la generación y distribución de energía eléctrica, gas natural y otros combustibles. (Millán, 2014)

Esta industria ha implementado avances importantes para reducir riesgos y costos, así como aumentar la productividad y rendimiento de todo lo que se utiliza en torno a la industria, por lo que hoy en día gracias a los grandes avances tecnológicos y de la ciencia han desarrollado avances significativos para esta industria mediante *machine learning* (ML).

Existen varios tipos de implementos que se utilizan en la industria acerera, pero esta investigación se enfocará en los hornos de fundición por arco voltaico, que son los que se utilizan en las plantas que se analizan.

En la siguiente ilustración se muestran los principales componentes de un horno de fundición por arco voltaico.

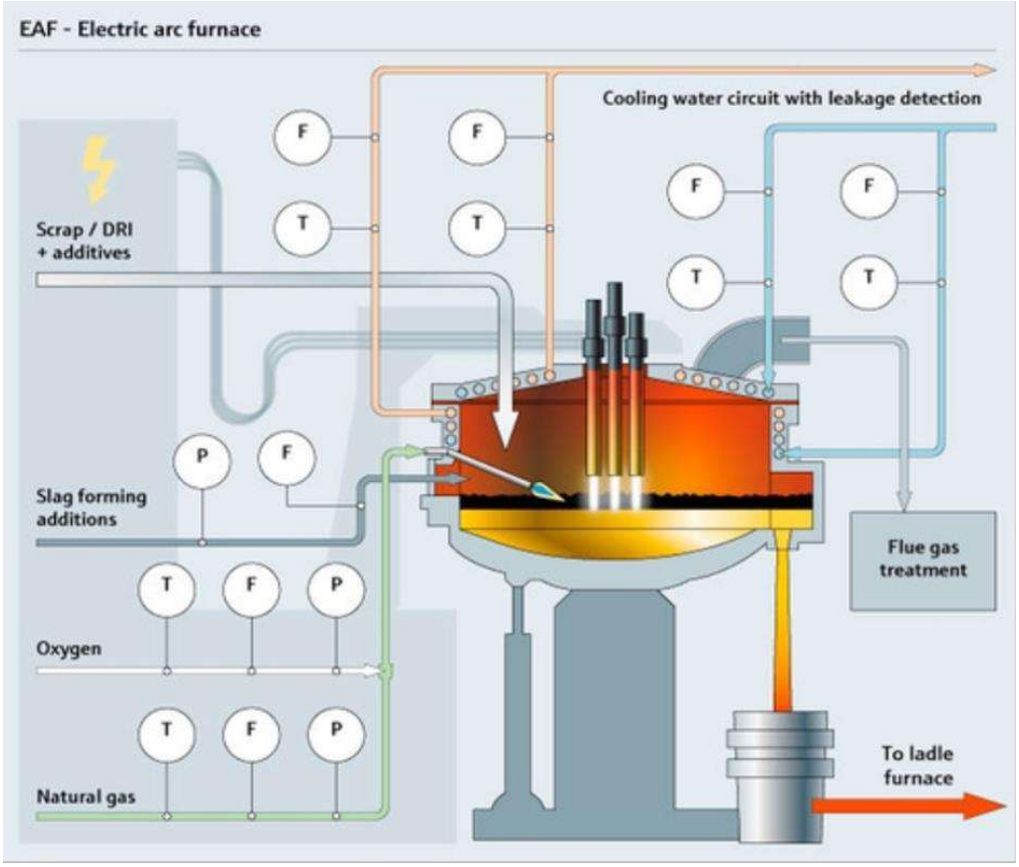

Figura 1.2: Horno por arco voltaico. (García, 2021)

Hay dos partes en las que dividimos los componentes y de donde se extrajo las bases de datos con las que trabajamos en esta tesis; el transformador que constantemente adapta la energía eléctrica para el proceso de fundición que conecta con los electrodos (regularmente de grafito); y el horno, que contiene la bóveda, la cuba (parte inferior por donde se suministra la chatarra) y donde los electrodos son introducidos (parte superior) para el proceso de fundición, el cual es el siguiente:

El ciclo comienza cuando la chatarra es introducida al horno por medio de la cuba (un tipo plato de entre 5 a 8 metros de diámetro) cuya capacidad varía entre 0 y 300 toneladas (comúnmente se utilizan coladas de 60 ton), la bóveda del horno se cierra y se precalienta con unos quemadores (o lanzas) que funcionan con gas, posteriormente se insertan los electrodos, los cuales son operados por unos brazos mecánicos que están interconectados a

un transformador que se encarga de adaptar la energía eléctrica a dichos electrodos; se inicia la corriente eléctrica y con esto el proceso de fusión, al terminar la colada se vacía; AMI tiene sensores en varios puntos de la bóveda del horno y estos sensores están conectados a un dispositivo que se encarga de guardar las mediciones que toman a una red local, éstos datos los conocemos como "CCR" y del mismo modo en el transformador cuenta con sensores y otro dispositivo de almacenamiento para guardarlos en la red local, éstos datos se conocen como "Gas online".

Se puede observar el funcionamiento de los hornos, el cual fue la fuente de estudio de este trabajo. Se elaboró este proyecto para una empresa que se centra en tener sistemas que miden los gases dentro de estos hornos, así como muchas otras variables para tenerlos en control. Dependiendo de cada planta, las mediciones o toma de muestras pueden ser de diferentes intervalos de tiempo: cada 30 minutos, 45 minutos, una hora y hasta cada 24 horas, por lo que es importante interpretar correctamente los datos que nos envía dicho sistema (pre procesamiento). AMI (viene de "*automation*") es una compañía que se dedica a la automatización internacional y soluciones de control y optimización, especializándose en hacer equipo que mantenga bajo control otros equipos con los que opera la planta en cuestión. Dicha empresa tiene sus propios equipos que se dedican al desarrollo e implementación de software en las diferentes plantas, son sistemas que mediante sensores toman mediciones y las pasan a una base de datos; estos datos son con los que trabajamos y que mencionamos anteriormente.

El control de procesos, así como predecir eventos son temas que siempre generan seguridad entre las personas a cargo de las herramientas e instrumentos que utilizan en su trabajo de día a día, por lo que se busca el estado saludable del transformador mediante el desarrollo de este proyecto.

1.1 Motivación

La empresa pretende conseguir un método automático que, en caso de que algún horno o transformador presente cambios que pongan en riesgo la operación o seguridad de la planta, le proporcione la(s) posible(s) causa(s) que originó ese cambio e inclusive poder tomar acciones antes de que lleguen a situaciones críticas; así como mejorar el rendimiento de todos los materiales y aditamentos que se utilizan en el proceso de fundición (específicamente el ahorro en consumo de electrodos).

El objetivo particular de la empresa es lograr el ahorro en consumo de electrodos en un 5%, lo cual traducido a recurso material son millones de pesos al año que estarían ahorrando para cada planta bajo su gestión.

Este trabajo permitirá a las empresas fundidoras de acero maximizar rendimientos y productividad, reducir costos y evitar accidentes dentro de las plantas; esto a su vez generará estabilidad a la empresa e impulsará la economía del país en el que ésta opere.

En lo personal, desde inicio del posgrado tuve mucho interés en aprender y aplicar técnicas de ML, y que mejor oportunidad que con datos reales de este proyecto en la industria acerera, por lo que lo tomé con gran entusiasmo y motivación en cuanto me platicaron del proyecto en cuestión.

1.2 Enfoque

La presente tesis se enfocó en aplicar un método que ayude al control y predicción de los gases que se generan dentro del transformador que adapta la energía eléctrica a la necesidad del horno de fundición de acero, los sensores que nos arrojan la información son capaces de monitorear 10 gases, entre otros factores como lo son temperatura ambiente, pero nuestro objeto de estudio son los gases conocidos como gases combustibles (Acetileno, Hidrógeno, Etileno, Monóxido de Carbono, Etano y Metano); esto debido a que por recomendación de la empresa son los primeros a estudiar y analizar si, mediante desarrollo y aplicación de un método aplicado en tiempo real, detecte y arroje, en caso de variación en el(los) gas(es), la(s) causa(s) principal(es) a tomar en cuenta para su control.

Cabe mencionar que los hornos de fundición de acero por arco voltaico son los que son más utilizados en la actualidad, en la industria acerera, debido a sus características que menciono más adelante.

La primera opción para este trabajo fue modelos de *machine learning*, por lo que utilizamos métodos clasificadores, principalmente métodos no supervisados ya que el objetivo es dejar un modelo automatizado que funcione en tiempo real, por lo que es importante que no necesite, al menos en tanta frecuencia, de algún operador manual, y conforme fuimos realizando pruebas fuimos modificando el modelo, mas no el objetivo.

1.3 Justificación

Este proyecto nace de la necesidad de las empresas para modelar e implementar nuevos métodos de control y monitoreo a los hornos de fundición de las plantas acereras, para lograr mejores rendimientos en general en todo el proceso.

En la actualidad se ha desarrollado y mejorado procesos relacionados con la industria, así como otros rubros (medicina, mercadotecnia, financiera, entre otros), con ayuda de ML con resultados favorables. No existe hasta el momento un trabajo con el mismo enfoque que el que estamos por desarrollar, por lo que este proyecto es pionero en este aspecto (mejorar el control y medición de los hornos de fundición en tiempo real). Está plenamente justificado para incrementar la rentabilidad de la empresa, la seguridad en las plantas acereras, y mejorar la productividad de este sector tan importante tanto a nivel nacional como mundial, que es la industria acerera.

La productividad de la empresa se asegura al momento de no tener que realizar paros emergentes de la planta, por algún riesgo dentro de los hornos, un mantenimiento correctivo, y mediante la conservación de los insumos de una manera más eficiente alargando sus ciclos de vida. Lo que se convierte en ahorro en gastos y generación de mayor producción.

En seguridad se beneficiará mucho, ya que ha ocurrido en diversas plantas la explosión de los hornos y/o transformadores con que operan estas plantas generando pérdidas humanas, daños materiales por millones de dólares y paro de operaciones de la planta por un periodo determinado de tiempo, por lo que al implementar este proyecto mejoraría en todos los aspectos a dichas plantas.

1.4 Hipótesis

Es posible implementar un modelo/algoritmo informático para detectar y definir el comportamiento de los gases dentro del transformador, encargado de suministrar energía eléctrica a los hornos de fundición de acero por arco voltaico con una efectividad superior al 85%.

1.5 Objetivo

1.5.1 General

Elaborar un sistema que pueda detectar el comportamiento de los gases, dentro de un transformador en tiempo real, que suministran de energía a los hornos de fundición de acero por arco voltaico.

1.5.2 Específicos

1. Detectar segmentos en los gases que se generan dentro de los transformadores.
2. Aplicar el modelo en tiempo real para prevenir fallas o accidentes lo antes posible.

Capítulo 2: Marco Teórico

En esta sección se describen los conceptos y procesos necesarios para el completo entendimiento de este documento; cada sección hablará de un tema diferente e incluirá su definición, uso y especificaciones.

2.1 Antecedentes

El proyecto es interesante ya que no se encuentra información en literatura (al menos hasta el momento) que aborde este tema en particular. Encontré un artículo (Ge, Song, Ding, & Huang, & & Dogo, 2017) que habla sobre la mayoría de las técnicas que utiliza la minería de datos en el sector industrial en general, el cual es un artículo muy completo de todas las metodologías, alcances y porcentajes de participación en procesos específicos y qué tan precisos son en comparación con otros métodos. Me parece muy importante esta aportación ya que al analizar similitudes con lo que yo quiero investigar, puedo tomar como base los modelos que ahí ya hayan detectado como más funcionales para cada operación.

Los análisis y la minería de datos han jugado un papel importante para el descubrimiento de conocimiento y es lo que pretendo hacer con este proyecto aumentar el conocimiento.

El trabajo presentado por Nkonyanaun y otros (2019) es interesante ya que aplican procesos de minería de datos para clasificar y predecir cantidades de posibles defectos antes de la fundición, esto con el fin de mejorar la precisión de la calidad del producto terminado según las especificaciones que requieran. En este estudio utilizan técnicas como *random forest*, redes neuronales y *Support Vector Machine* (SVM), técnicas propias de *Machine Learning* (ML) y hacen comparaciones entre sí, resultando en este estudio *random forest* la de mayor precisión con un 77.8%.

El problema se origina cuando los tanques salen de sus valores “normales” y tienen que ser monitoreados por expertos en el área para definir si se le tiene que hacer alguna intervención, al incrementarse el número de tanques (o clientes para AMI) es complicado mantener sobre observación a todos; por lo que se pretende solucionar esto mediante un modelo automatizado.

El problema inicia con la segmentación de las series de tiempo lo cual tenemos muchos trabajos relacionados al respecto, donde Mörgen (2006) nos explica que todos los métodos de segmentación necesitan que definamos los números de segmentos esperados o, en su defecto, el umbral para que se detecte automáticamente, por lo que exploramos varios métodos (detallados en el capítulo de metodología) para lograr la solución adecuada.

2.2 Horno de fundición por arco voltaico

2.2.1 ¿Qué son los hornos de fundición?

Son hornos que se utilizan para reducción de óxidos a metales, matificación por fusión de minerales sulfurosos y aún para descomposición de carbonatos (calcinación). Los materiales a tratar se cargan por la parte superior del horno y los productos se extraen por la parte más baja. (*Home*, 2022)

Horno de fundición por arco voltaico: Disponen de electrodos de carbón por los cuales circula una corriente eléctrica, generando entre los electrodos y el metal a fundir un arco voltaico. Esta energía se transforma en calor. Estos hornos se emplean básicamente para la fundición de aceros. (*Home*, 2022)

2.2.2 Funcionamiento de los hornos de arco eléctrico

"Los hornos de arco eléctrico (u hornos de arco) son hornos de alta temperatura que utilizan corrientes eléctricas de alto voltaje como su elemento de calentamiento principal. Estos hornos son una parte crucial de las operaciones de reciclaje de hierro y acero. Aparecen en "mini-molinos" que recicla la chatarra de hierro para su reutilización. Los hornos de arco también se utilizan en la producción de acero. El diseño de estos hornos los hace lo suficientemente simples como para que los aficionados también los construyan en casa" (www.areametalurgia.com, 2020).

2.2.3 Diferentes tipos de hornos

"Hay cuatro tipos principales de hornos que se utilizan en las fundiciones en la actualidad: hornos de cubilote, hornos de inducción, hornos de crisol y hornos de arco. Cada uno de estos utiliza métodos diferentes para completar la misma tarea general: derretir metal o chatarra con cantidades increíbles de calor para así poder moldearlo y usarlo para todo, desde la construcción hasta la electrónica y también en los artículos domésticos comunes. Los usos específicos de cada horno dependen de la forma en que funciona, así como de cuál es su principal fuente de calor" (www.areametalurgia.com, 2020).

"Los hornos de arco eléctrico actúan como una especie de término medio. Al utilizar electricidad y, opcional/ocasionalmente, combustible sólido como fuentes de calor, pueden procesar tanto la chatarra como el mineral de hierro para fabricar acero" (www.areametalurgia.com, 2020).

2.3.4 Los componentes del arco voltaico

"Un horno de arco eléctrico es esencialmente una tetera gigante resistente al calor que funciona con tres picos de grafito. El horno tiene una tapa removible enfriada por agua que sostiene los picos de grafito y está conectado a grandes líneas eléctricas que actúan como electrodos. Cuando se levanta la tapa, el horno se puede cargar con cualquier combinación de chatarra de hierro, mineral de hierro, fundente y combustible sólido, y cuando se cierra y se asegura firmemente, los electrodos se pueden bajar a la chatarra para comenzar el proceso de fusión" (www.areametalurgia.com, 2020).

2.4.5 Elaboración del Metal

"Para la elaboración del metal, los hornos de arco derriten la chatarra y el mineral mediante el uso de sus electrodos de grafito. Cuando se alimenta electricidad al horno, la energía salta entre los dos electrodos energizados y al electrodo neutro conectado a tierra. Los arcos eléctricos de alto voltaje creados por estos picos de grafito emiten grandes cantidades de calor directo y radiante que derriten el contenido del horno. Si también se han colocado combustibles sólidos en el horno, el calor de los electrodos se transfiere al combustible y lo enciende, aumentando el calor general. Cuando el metal fundido está listo para su uso, se puede drenar a través de un puerto especial en el horno para fundir y forjar. Todo el proceso puede durar tan solo una hora" (www.areametalurgia.com, 2020).

2.5.6 Inconvenientes del horno de arco

"Aunque los hornos de arco son útiles y pueden procesar grandes cantidades de chatarra en un período de tiempo relativamente corto, sus características les dan algunas desventajas que deben considerarse antes de usarlos. Más que cualquier otra

cosa, los hornos de arco usan una cantidad increíblemente grande de electricidad, hasta el punto que se sabe que el uso de hornos de arco industriales hace que la energía parpadee en el área y los tiempos de uso generalmente se ajustan a períodos en los que la electricidad no se usa ampliamente" (www.areametalurgia.com, 2020).

2.3 Machine Learning (ML)

ML es una forma de la IA que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita. Conforme el algoritmo se alimenta con datos de entrenamiento, es posible producir modelos más precisos fundamentados en información procesada y preparada; y como menciona IBM en su página:

> "Un modelo de ML es la salida de información que se genera cuando se entrena su algoritmo con datos. Después del entrenamiento, al proporcionar un modelo con una entrada, se le dará una salida. Por ejemplo, un algoritmo predictivo creará un modelo predictivo. A continuación, cuando proporcione el modelo predictivo con datos, recibirá un pronóstico basado en los datos que entrenaron al modelo" (www.ibm.com, 2021).

> "ML permite modelos a entrenar con conjuntos de datos antes de ser implementados. Algunos modelos están online y son continuos. Este proceso iterativo de modelos online conduce a una mejora en los tipos de asociaciones hechas entre los elementos de datos. Debido a su complejidad y tamaño, estos patrones y asociaciones podrían haber sido fácilmente pasados por alto por la observación humana. Después de que un modelo ha sido entrenado, se puede utilizar en tiempo real para aprender de los datos. Las mejoras en la precisión son el resultado del proceso de entrenamiento y la automatización que forman parte del machine learning" (www.ibm.com, 2021).

"Las técnicas de ML son necesarias para mejorar la precisión de los modelos predictivos. Dependiendo de la naturaleza del problema que se está atendiendo, existen diferentes enfoques basados en el tipo y volumen de los datos" (www.ibm.com, 2021).

2.3.1 Aprendizaje supervisado

"El aprendizaje supervisado comienza típicamente con un conjunto establecido de datos y una cierta comprensión de cómo se clasifican estos datos. El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso de analítica. Estos datos tienen características etiquetadas que definen el significado de los datos" (www.ibm.com, 2021).

En otras palabras, se identifica los valores con alguna etiqueta predefinida, esto para indicarle al modelo con que va a relacionar cada que se encuentre un valor de ese tipo, y al aplicar el método pueda clasificar en una u otra etiqueta según lo que haya "aprendido" o con lo que se haya entrenado dicho modelo.

2.3.2 Aprendizaje no supervisado

"El aprendizaje no supervisado se utiliza cuando el problema requiere una cantidad masiva de datos sin etiquetar. Por ejemplo, las aplicaciones de redes sociales, tales como Twitter, Instagram y Snapchat, tienen grandes cantidades de datos sin etiquetar. La comprensión del significado detrás de estos datos requiere algoritmos que clasifican los datos con base en los patrones o clústeres que encuentran. El aprendizaje no supervisado lleva a cabo un proceso iterativo, analizando los datos sin intervención humana" (www.ibm.com, 2021).

Este tipo de aprendizaje es diferente al supervisado precisamente por el tema de etiquetar datos con anterioridad, en este tipo de aprendizaje se etiqueta después de ser analizado y clasificado, según sus características a algún clúster con similitud entre todos sus componentes.

2.3.3 Aprendizaje de refuerzo

> "El aprendizaje de refuerzo es un modelo de aprendizaje conductual. El algoritmo recibe retroalimentación del análisis de datos, conduciendo al usuario hacia el mejor resultado. El aprendizaje de refuerzo difiere de otros tipos de aprendizaje supervisado, porque el sistema no está entrenado con el conjunto de datos de ejemplo. Más bien, el sistema aprende a través de la prueba y el error. Por lo tanto, una secuencia de decisiones exitosas conduce al fortalecimiento del proceso, porque es el que resuelve el problema de manera más efectiva" (www.ibm.com, 2021).

Como la definición dice, este método se forja en base de prueba y error, en la recompensa o castigo que pueda afectar a la decisión final tomada por el algoritmo; este tipo de algoritmos se utiliza mucho en robótica y para resolver videojuegos de una manera mas eficiente, incluso que un humano, pero eso es otro tema.

2.4 Series de tiempo

"Llamamos serie de tiempo a un conjunto ordenado de valores *y(t)* que están asociados biunívocamente a puntos consecutivos y equiespaciados de la variable de dominio *t*. El valor del dominio siempre crece…, aunque la variable de dominio no es necesariamente el tiempo, el paradigma de ese tipo de series es el de muestras tomadas regularmente en el tiempo, de ahí su nombre: series de tiempo." (Nava, 2015).

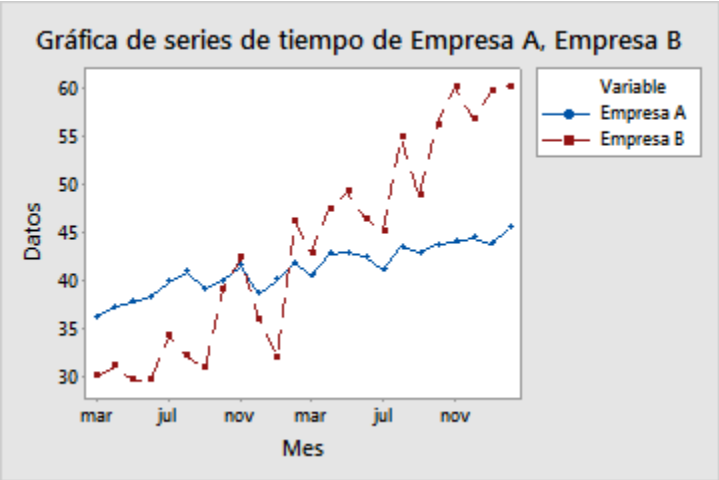

Figura 2.1: Serie de tiempo. (Revisión general de Gráfica de series de tiempo, 2022)

2.5 Detección de anomalías

Como Johnson (1992) definió a la anomalía “valor atípico como observación en un conjunto de datos que parece ser inconsistente con el resto de ese conjunto de datos”, o de manera similar Hawkins (1980) dice que “valor atípico como una observación que se desvía tanto de otras observaciones como para despertar sospechas de que se generó por un mecanismo diferente”. Es un valor fuera de la normalidad que la mayoría de los datos trae y por lo tanto se debe analizar que sucede con ese dato o datos.

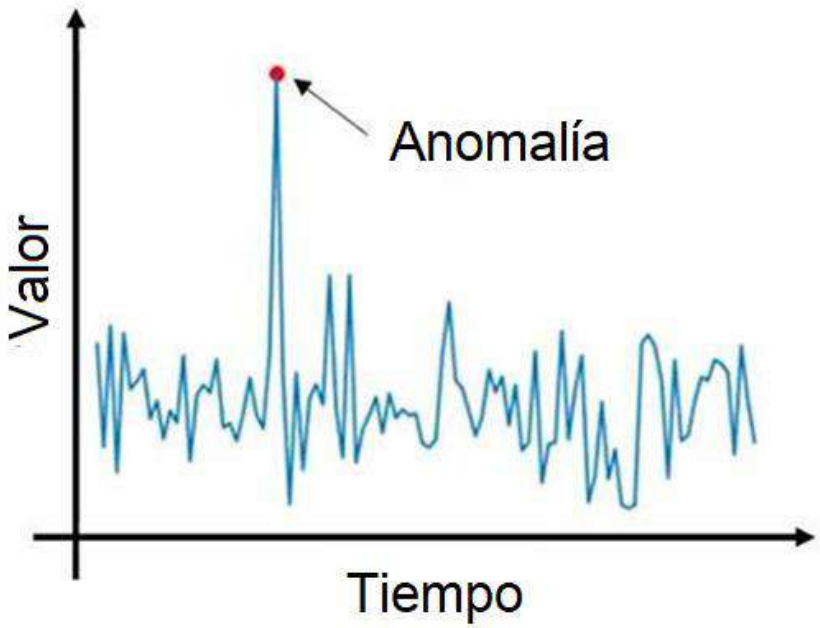

Figura 2.2: Representación de valor anómalo. (What is an isolation forest, 2022)

2.6 Segmentación para series de tiempo y tiempo real

Las series de tiempo llamadas también series cronológicas o series históricas son un conjunto de datos numéricos que se obtienen en períodos regulares y específicos a través del tiempo, los tiempos pueden ser en años, meses, semanas, días u otra unidad adecuada al problema que se esté trabajando.

La segmentación es identificar puntos que mantienen cierta tendencia similar para poder agruparlos en un subconjunto de datos y poder analizar dicho subconjunto de datos para poder identificar qué condiciones o variables lograron mantener de ese modo ese comportamiento.

El tiempo real es lo que se conoce como *streaming* y es más que conforme aparezcan o se guarden los datos nuevos se genere el análisis correspondiente para que casi al instante

podamos dar solución o interpretación a lo que sucede en dicho momento y no esperar horas o días en detectarlo.

Es importante mencionar que en este ámbito no existe trabajo relacionado específicamente en la industria acerera, por lo que es la mayor aplicación que tendrá este trabajo de tesis.

En la figura 2.3 se muestra un ejemplo de la segmentación de una serie de tiempo.

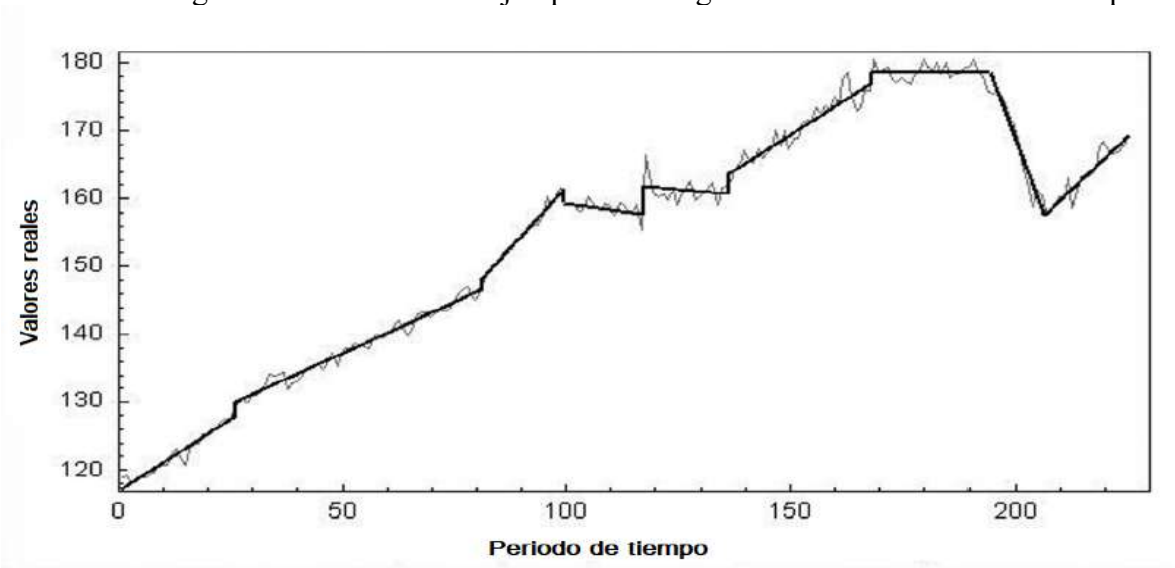

Figura 2.3: Ejemplo de segmentación

Capítulo 3: Metodología

En esta tesis buscamos identificar los segmentos de los gases en el transformador, así como las variables del horno en tiempo real. Para esto es importante preparar los datos que tenemos para alimentar nuestro modelo.

3.1 Preparación de los datos

La metodología es la siguiente:

3.1.1 Adquisición de los datos: Como mencioné antes, la empresa AMI tiene los recursos necesarios para la obtención de mediciones y almacenar éstas en sistemas internos a su organización, dentro de la bóveda del horno, como también en los transformadores correspondientes. Estos datos provenientes directamente de los hornos los conocemos como CCR, y los provenientes del transformador como Gas online, lo que nos enfocaremos solo a Gas online.

3.1.2 Pre-procesamiento de los datos:

a. Utilizamos el lenguaje de Python y la librería pandas para procesamiento de los datos, los cuales limpiaremos eliminando valores nulos o vacíos,

b. Quitamos o removimos variables que no aportan información relevante para el análisis previamente consultada con expertos en el área y se eliminaron valores que se registraron menos de 50 minutos ya que no debe haber valores por menos de ese tiempo,

c. Normalizamos los valores de los gases respecto su valor límite,

d. Los valores de fechas fueron convertidos a un valor secuencial, tomando como base los días y fracción de ellos para representar horas y minutos.

e. Se tomó el valor máximo por día para ajustar la frecuencia de muestreo y el umbral funcionara de manera igualitaria para todos los gases.

3.1.3 Proceso:

a. Se hicieron evaluaciones en tres métodos diferentes, que más adelante serán descritos en su totalidad.

b. Elegimos el "gas clave" de cada punto para enfocarnos a dicho(s) gas(es) ("el término gas clave es propiedad intelectual de la empresa y no tengo permiso de divulgar o dar información del proceso de obtención de dicho gas, pero puede ser de 1 hasta 4 gases clave")

Quedó la siguiente base de datos:

Tabla 1.1: Base de datos.

Nombre	Datos	Formato
Gas online	13,968	csv

La cual utilizaremos para extraer los datos de cada gas y procesarlos por separado para elaborar sus segmentos.

3.2 Aplicación sobre los métodos/algoritmos:

Ya teniendo preparados los datos para su análisis, procedimos a aplicar los diferentes métodos, es importante recordar que nuestro objetivo es definir segmentos que estén marcados por un evento y/o un cambio de tendencia sobre la serie de tiempo.

En esta sección se describe todo el proceso que se tomó con cada uno de los métodos: descripción del método, parámetros utilizados, descripción de la operacionalidad del método.

La evaluación de cada método fue en base a los resultados esperados previamente detectados por los expertos en el tema (en este caso las personas (ingenieros) de AMI) y son como se muestran en la siguiente figura:

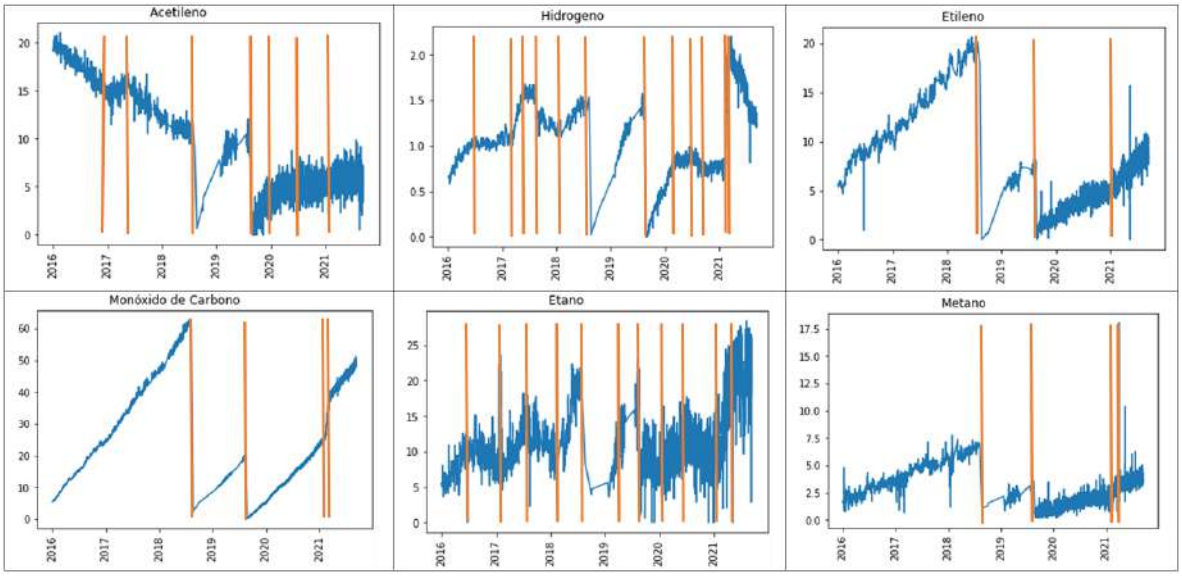

Figura 3.1: Segmentación esperada.

3.2.1 Isolation forest

Como ya mencionamos antes este método no supervisado de ML ha tomado popularidad y es muy utilizado para la detección de anomalías, por lo que fue el primer método elegido para evaluar su desempeño.

3.2.1.1 ¿Qué es?

Isolation Forest es un algoritmo de ML para la detección de anomalías, entendiendo por anomalía cualquier valor que salga del rango que consideramos normal. Los casos, que tienen una longitud de ruta promedio más corta en el bosque de aislamiento entrenado, se clasifican como puntos anómalos. (Bai, 2021)

Es un árbol de decisión, en donde se van aislando los elementos, entre mas fácil se su aislamiento significa que es fácil de clasificar como valor anómalo asi como se muestra en la figura 3.2.

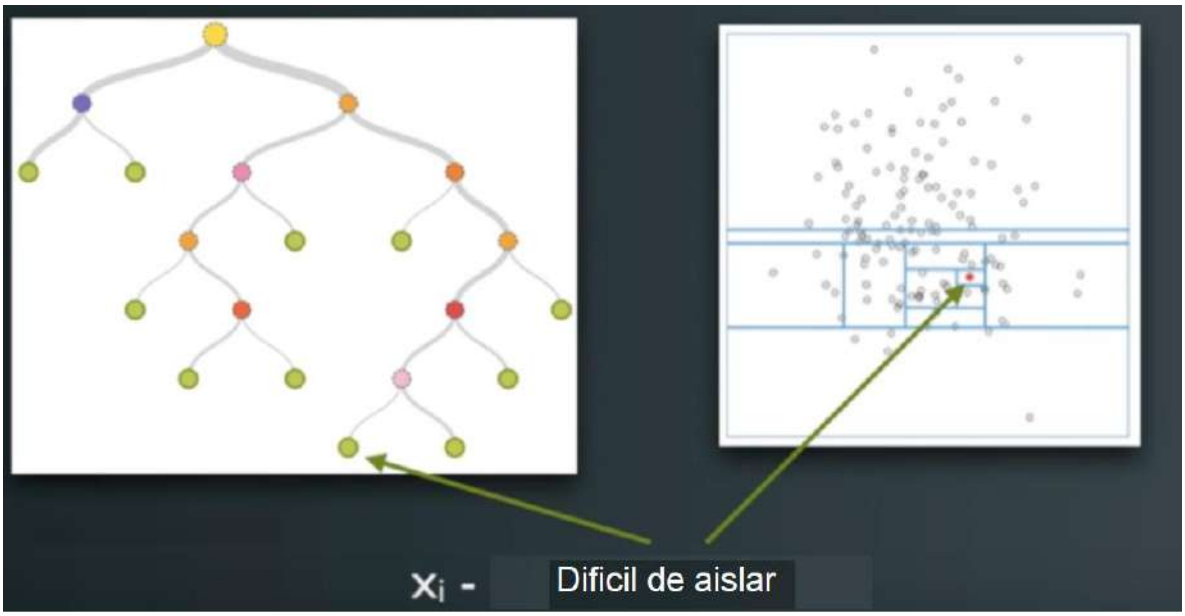

Figura 3.2: Observación difícil de aislar, porque es muy común. A la izquierda el árbol y a la derecha las subdivisiones (en 2D) que las decisiones van haciendo sobre el conjunto de datos hasta que un valor queda aislado. (Inteligencia artificial deteccion de anomalias y su aplicacion al covid 20, 2022)

Por otro lado, en la figura 3.3 observamos a que se refiere un valor anómalo o fácil de aislar.

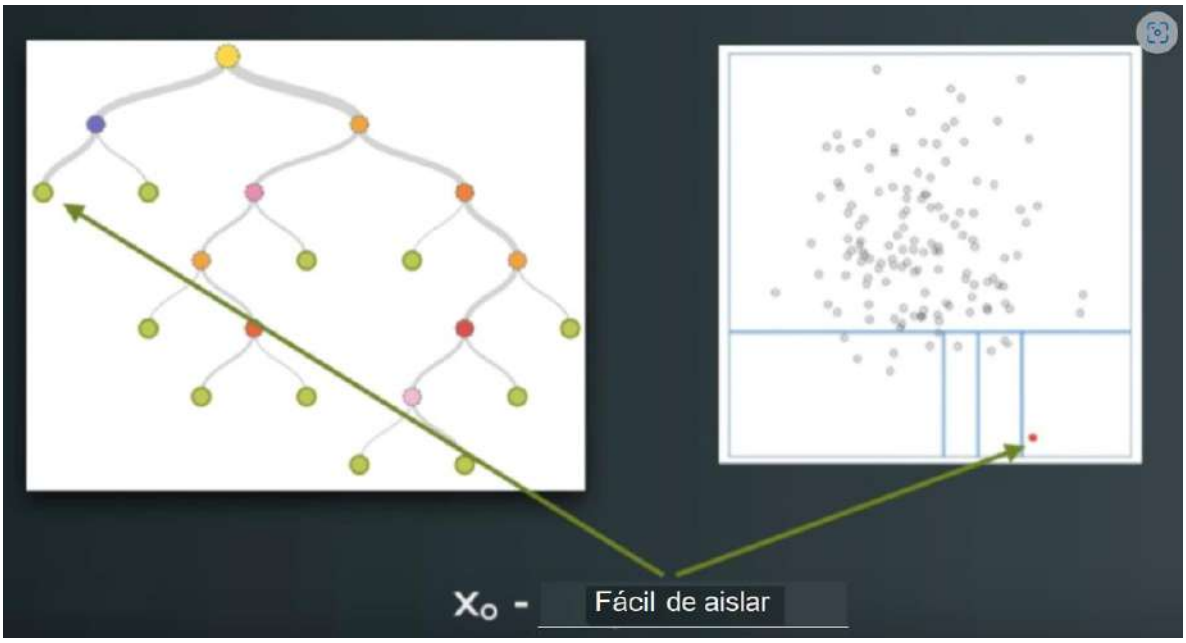

Figura 3.3: Observación fácil de aislar, porque es fuera de la regularidad de los demás datos. (Inteligencia artificial deteccion de anomalias y su aplicacion al covid 20, 2022)

3.2.1.2 ¿Cómo funciona?

"Para entender cómo funciona el Isolation Forest, tenemos que ver cómo un árbol de decisión es capaz de llegar a la conclusión de que un punto es anómalo. Los pasos que realiza un árbol consisten en:" (medium.com, 2019)

1. Elegir un registro dentro del conjunto de datos y sus diferentes variables.
2. Elegir un valor aleatorio que esté dentro del mínimo y el máximo de cada variable.
3. Crear nodo o ramificación: si el valor del registro al que estamos mirando es mayor o menor que el valor aleatorio anterior, repetiremos el ejercicio de evaluar nuestro punto con el mínimo y máximo de nuestro intervalo, esta vez más acotado, siendo el nuevo máximo o el mínimo el punto de corte de la rama creada.
4. Realizar el tercer paso las veces que sea necesario hasta que no se pueda ramificar más y el punto esté aislado.

"Así, cuantas menos ramas haya necesitado el árbol para aislar al punto, más anómalo será. Y si hacemos, por ejemplo, 100 árboles y calculamos la media del número de ramas por cada valor, tendremos una aproximación bastante robusta del grado de anomalía de cada observación o registro de nuestro conjunto de datos… Así es, sin entrar en términos ni formulaciones matemáticas, cómo funciona el Isolation Forest. Una gran ventaja de este algoritmo en comparación a otros métodos es que no utiliza medidas de distancia, similitud o densidad del conjunto de datos, que suele ser computacionalmente muy costoso. El *Isolation Forest* tiene una complejidad que crece linealmente gracias a las bondades del sub-muestreo: computa árboles por subpartes del conjunto de datos. Así, tiene la capacidad de escalar en *datasets* grandes y con muchas variables irrelevantes" (medium.com, 2019).

3.2.1.3 Parámetros

Se utilizó la librería de scikit-learn de Python para el método y un estandarizador que normaliza los datos un poco más para que su detección sea más exacta; mediante la siguiente línea de código:

Tabla 1.2: Librerías importadas para utilizar el método Isolation Forest en Python

from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import IsolationForest

Definimos el *contamination_rate* del método en .05, que es el umbral para la detección de anomalías, lo que significa que le indicamos al método que alrededor del 5% de los valores pueden ser anómalos, debido a la cantidad de datos que manejamos, es importante no generar muchas alarmas ya que en una planta real es posible atender tan seguido un solo horno.

3.2.1.4 Funcionamiento

El método como tal es muy rápido, para la cantidad de datos que se opera, el análisis de los gases es de manera individual y aunque en detección de anomalías puede ser muy útil, en cuanto a segmentación de la serie de tiempo es muy poco preciso.

3.2.2 Autoencoders

Menciona Jesús Martinez en la pagina de datasmarts que:

> "Típicamente, las redes neuronales profundas son utilizadas en problemas de aprendizaje supervisado. Es decir, a partir de data etiquetada, bien sea con valores continuos (como el precio de un inmueble) o discretos (categorías, como gato o perro), buscamos hallar o aproximar la función que correlaciona entradas y salidas. De manera más sencilla, queremos saber para cada X, cuál es su y correspondiente" (datasmarts.net, 2019)

> "No obstante, las redes neuronales también albergan el potencial de brillar en contextos donde el conjunto de datos no está etiquetado (aprendizaje no supervisado)" (datasmarts.net, 2019)"

> "Unas de las aplicaciones más fascinantes, por no decir útiles, en este respecto es la compresión de información. Para ello, utilizamos un tipo especial de redes neuronales conocidas como *autoencoders"* (datasmarts.net, 2019).

> "Los *autoencoders* en vez de correlacionar **X** e **y**, lo que buscan es, a partir de una representación reducida de la entrada, reconstruirla. Para ello, cuentan con tres piezas específicas" (datasmarts.net, 2019) (figura 3.4):

- “Codificador (*Encoder*): Es la porción de la red neuronal encargada de comprimir los datos de entrada, típicamente en forma de vector, conocido como *bottleneck feature*.
- *Bottleneck feature*: Una especie de cuello de botella es la capa media que divide al *encoder* y al *decoder*. Es la representación comprimida de la entrada, usualmente bastante más pequeña en cuanto a dimensiones e información.
- Decodificador (*Decoder*): Es la segunda parte de la red neuronal, a cargo de reconstruir los datos de entrada originales a partir de la representación comprimida resultante del *feature* cuello de botella” (datasmarts.net, 2019).

“En teoría, los *autoencoders* deberían ser capaces de reproducir la data original sin pérdida de información, pero no suele ser el caso en la práctica, puesto que al convertir la entrada al bottleneck feature es inevitable que prescindamos de información que es difícil de reconstruir por el decoder” (datasmarts.net, 2019).

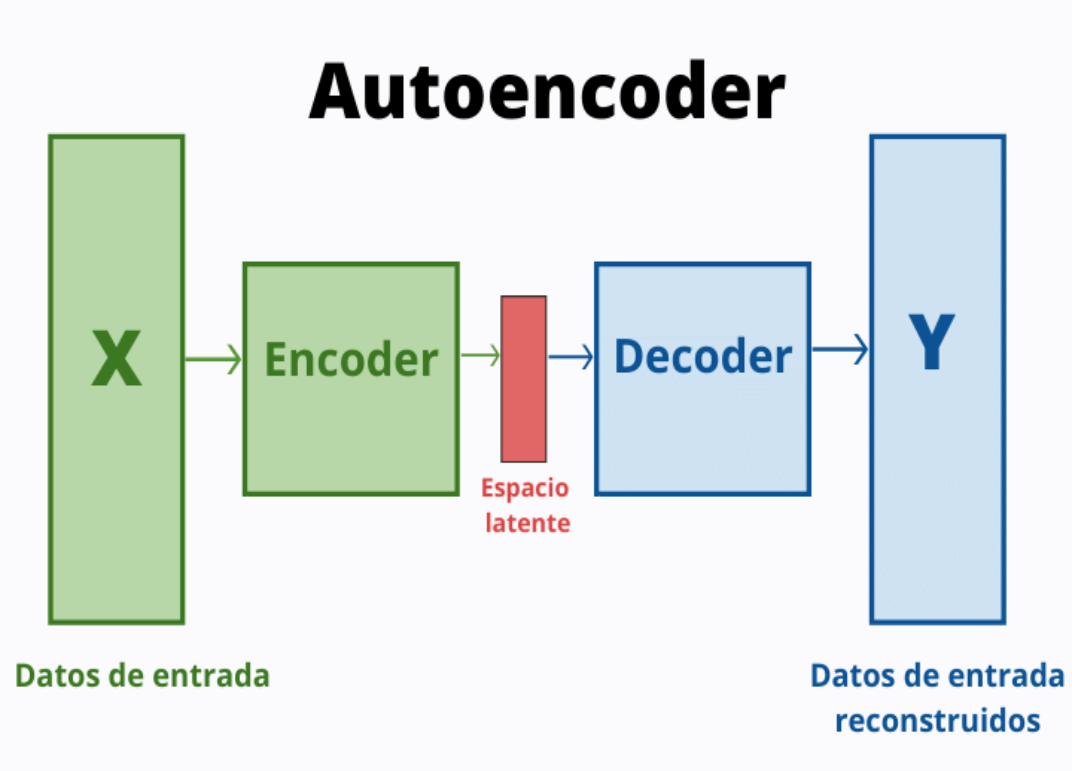

Figura 3.4: Diagrama Autoencoder. (datasmarts.net, 2022)

Con este método utilizamos un modelo multivariado para la detección de anomalías en nuestras series de tiempo de cada uno de los hornos que tenemos en las diferentes plantas, permitiendo un análisis más completo y no tan a simple vista como normalmente se realiza.

Este es un método similar al anterior en funcionamiento, pero la diferencia radica en que este brinda la oportunidad de llevar a cabo un análisis multivariado, es decir que todos los gases forman parte del análisis y dentro de sus virtudes es que muestra información que no es posible ver a simple vista, este método de *deep learning* se puede utilizar desde 1 variable hasta n número de variables.

3.2.2.1 Parámetros 1 variable

Como mencioné el método acepta de 1 a "n" variables y se hizo el experimento tomando en cuenta cada gas individualmente (debido al comportamiento del algoritmo y viendo que no es útil de este modo se puso ejemplo de un solo gas (CO)) y tomando todos los gases a la vez; primero importamos las herramientas necesarias de la siguiente manera:

Tabla 1.3: Librerías importadas y parámetros para el método Autoencoders en Python.

Librerías
from pyod.models.auto_encoder import AutoEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
Parámetros:

hidden_neurons = [25,1,1,25]
n_features=1
contamination=0.01(1%)
training data = ⅔
testing data = ⅓

Después de entrenar el modelo y calcular su score, procedo a analizar la distribución del score de los datos con un histograma el cual queda de la siguiente manera:

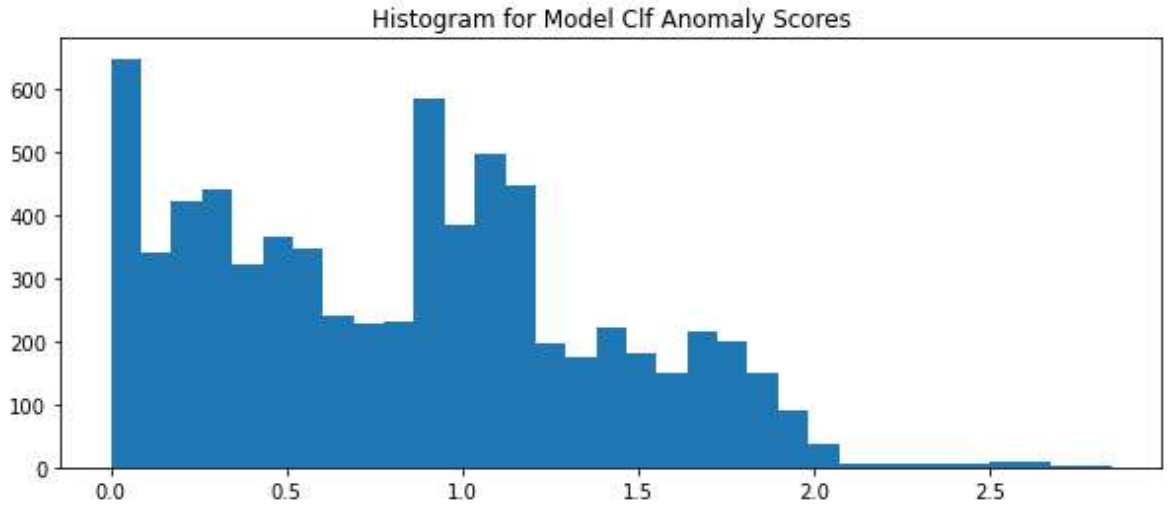

Figura 3.5: Histograma Monóxido de carbono.

De aquí obtenemos el criterio de tomar como anomalía aquellos puntos que en su score excedan 2.1 de valor aproximadamente y de aquí procedemos directamente a probar el modelo en los valores de *test,* donde los resultados los presentamos en la sección de resultados.

3.2.2.2 Parámetros multivariable

Tabla 1.4: Librerías importadas para utilizar el método Isolation forest.

Parámetros
hidden_neurons = [25,2,2,25]
n_features = 6
el resto quedó igual al de 1 variable

Gases:

Acetileno (C2H2), Hidrógeno (H2), Etileno (C2H4), Monóxido de Carbono (CO), Etano (C2H6) y Metano (CH4).

Mismo proceso de histograma del score de anomalías, para indicarle cuáles valores son tomados como tal:

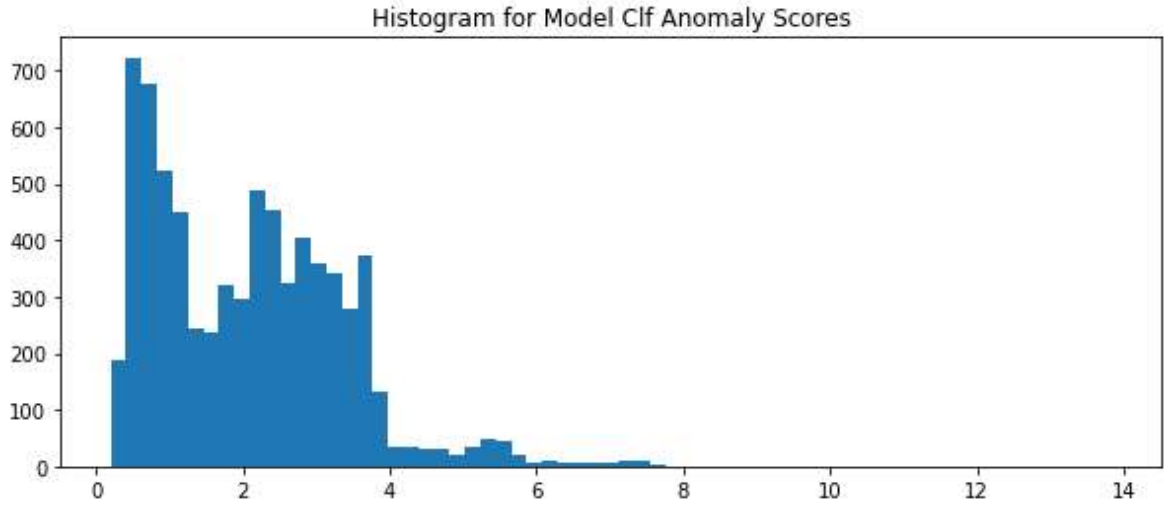

Figura 3.6: Histograma multivariable.

Podemos observar que a partir del 7 hay muy pocos puntos por lo que ese será nuestro criterio para marcar anomalías, y posible segmentación.

3.2.3 Programación Dinámica (PD)

La programación dinámica es una técnica de solvencia, que puede ayudar a simplificar procesos que contienen múltiples subproblemas. Debido a que la programación dinámica puede ayudar a optimizar el proceso de codificación para muchas aplicaciones informáticas; los profesionales en análisis de datos, programación y desarrollo de software a menudo aplican este proceso para agilizar su trabajo.

Su fundamento proviene de la ecuación de Bellman la cual es la siguiente:

$$OPT(j) = \begin{cases} 0 & \text{if } j = 0 \\ \min\limits_{1 \le i \le j} \{ e_{ij} + c + OPT(i-1) \} & \text{if } j > 0 \end{cases}$$

Figura 3.7: *Ecuación Bellman* (Bellman, 2005)

Donde OPT(j) es el costo mínimo entre cada punto y e_{ij} es la suma de los errores al cuadrado que existe entre cada punto; de aquí que nuestro umbral, es el que define hasta que nivel se puede considerar un punto nuevo, parte del mismo segmento y, cual es punto de cambio para un segmento nuevo.

A lo largo de la historia existen métodos ya definidos para segmentación en base a programación dinámica como lo son *bottom-up*, *sliding window* y *top-down*. Como lo menciona Lovrić en su artículo "*algorithmic methods for segmentation of time" (2014)*, este proyecto toma una semejanza al método SWAB descrito en el artículo, ya que es una

combinación entre *bottom-up* y *sliding window* pero con la diferencia que es completamente en tiempo real y no necesita 5 o 6 posibles segmentos para operar, más detalles vienen en el capítulo de metodología, que es donde se describe este algoritmo.

3.2.3.1 Características de la programación dinámica

La programación dinámica adquiere dos características importantes, que la convierten en una herramienta viable y eficaz para reducir el tiempo de programación y aumentar la funcionalidad y eficiencia del programa. (Indeed, n.d.)

Para este proyecto utilizaremos la programación dinámica para segmentar las series de tiempo de los gases, y definir rangos para establecer correlaciones y causalidad.

Como ya mencioné anteriormente, este algoritmo parte de la idea de problemas grandes reducirlos a pequeños para su solución, aquí podemos ver su desempeño.

Este algoritmo tiene dos fases de aplicación, la primera en datos históricos y la segunda en tiempo real. Se decidió probar este algoritmo ya que en el libro de "*Algorithm Design*" de (Kleinberg & Tardos, 2005, 25-30) resuelven el problema de mínimos cuadrados segmentados con esta metodología, y la dinámica para resolver ese problema es muy acorde a la necesidad de este proyecto.

3.2.3.2 Parámetros 1ra fase

En cuanto a parámetros solo necesita un archivo de texto que contenga la información separada por tabulador y cada renglón contiene el punto secuencial o *index* y el valor del gas; como comentamos anteriormente la serie de tiempo se ajustó en el eje x, de fechas a días y

su fracción para representar horas y minutos, tomando el primer valor como 0 y a partir de ahí se le suman los días transcurridos.

El algoritmo admite 2 parámetros: *penalty_desired* y *max_num_seg*; el primero corresponde a los grados de libertad para considerar que un punto ya no está dentro de la misma tendencia de los demás puntos, generando un segmento nuevo; y el segundo es para indicar cuántos segmentos quieres obtener del archivo que le estás proporcionando.
En este caso necesité que el algoritmo me detectara automáticamente por lo que solo utilicé el "*penalty_desired"*.

Tabla 1.5: Parámetro Programación Dinámica.

Parámetros
penalty_desired = 62

3.2.3.3 Funcionamiento 1ra fase

El algoritmo se ajusta muy bien en algunos gases, pero no en todos fue bueno, por lo que hubo que ajustar el umbral, este método, entre más cantidad de datos su tiempo de ejecución aumenta considerablemente; por lo que la reducción de los datos al máximo por día mejoró en gran medida la rapidez del algoritmo y su efectividad se mantuvo; primero observaremos cómo quedan los segmentos que brinda nuestro algoritmo en bases de datos históricas (no tiempo real) como se muestra en el capítulo de resultados.

3.2.3.4 Parámetros 2da fase (tiempo real)

Los parámetros son los mismos que la 1ra fase, solo que a este procedimiento le agregue una simulación de obtener datos en tiempo real con otro módulo conocido como *time.sleep* y en el algoritmo se agregó una animación para ver cómo se recorre la información en tiempo real con la opción *animation* de matplotlib:

Para efectos prácticos el *time.sleep* le di espacio de 1 segundo y la animación de la gráfica que la leyera cada segundo (1000 milisegundos) (también es importante tener a consideración que en la vida real los datos estarían llegando con diferentes frecuencias que van en posibilidad de cada 20 minutos, cada 30 minutos, cada hora, cada 3 horas, 6 horas, 12 horas, hasta 24 horas, según sea la necesidad del horno)

3.2.3.5 Funcionamiento 2da fase

Con la experiencia del modo histórico (o fase 1) sabíamos que conforme la base de datos fuera creciendo el algoritmo se iría haciendo más lento, por lo que se adoptó una alternativa para que no perdiera su efectividad, la alternativa consiste en al detectar un punto para segmentación, almacenar la base de datos y volver a correr el algoritmo desde el punto inmediato posterior iniciando un nuevo análisis, sin perder la información; esto nos da doble beneficio ya que aparte de lo ya mencionado, a la hora de hacer un análisis posterior (correlación) ya estará dividida la información (que es parte del trabajo a futuro).

Capítulo 4: Resultados

En este capítulo podremos ver los resultados obtenidos de cada uno de los métodos y algoritmos implementados para este proyecto, recordemos que la intención final de este proyecto es lograr controlar los gases dentro de un transformador que ajusta energía eléctrica para hornos de fundición de acero por arco voltaico; por lo que los resultados van enfocados a la regulación de esos gases.

Es importante mencionar que la empresa que necesita este proyecto, esperaba soluciones con ML, por lo que se buscaron opciones de detección de anomalías que son por lo regular métodos no supervisados de ML.

Se mostrarán uno por uno los métodos y se compartirán los resultados obtenidos, si es útil para nuestro resultado final y por qué.

4.1 Isolation forest

Este método es muy popular dentro de *anomaly detection* por lo que pensamos que sería ajustable a nuestras series de tiempo de cada gas; a continuación, muestro las gráficas con los resultados de los gases iniciales obtenidas a partir de este método:

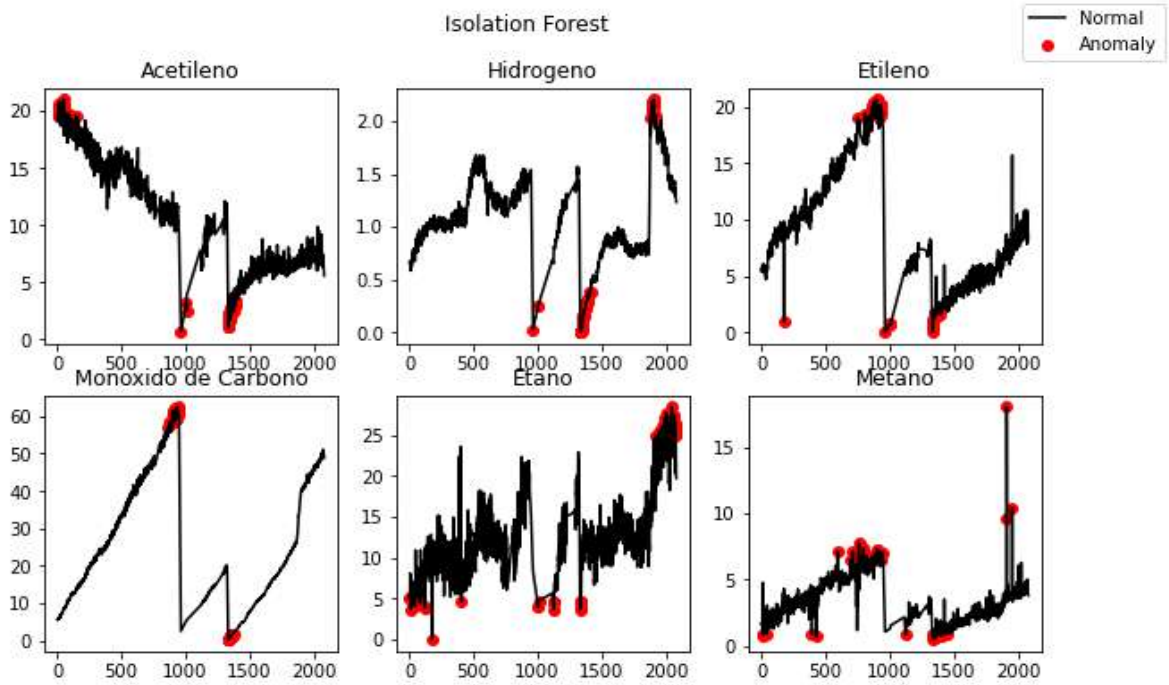

Figura 4.1: Isolation Forest.

En la figura 4.1 es importante agregar que, aunque el límite de puntos para detectar anomalías (*contamination_rate*), es el mismo en todos los gases no en todos detecta la misma cantidad de puntos, y esto es debido al comportamiento de cada gas, pero de igual manera no es útil para la segmentación requerida.

Comparando los valores esperados con los que arroja el método y podemos concluir lo siguiente: si detecta muchos de los puntos que necesitamos, pero también nos genera muchos falsos positivos, por lo que no nos permitiría la segmentación adecuada para avanzar al siguiente paso en la investigación; también podemos observar puntos que no fueron detectados por lo que tendríamos que complementar el método lo que bajaría la rapidez del algoritmo, lo cual no es adecuado y pasamos al siguiente método.

4.2 Autoencoders

El método de *autoencoders*, similar a *Isolation forest* es muy útil detectando anomalías, sobre todo en el modelo multivariado, que es donde se notó una mejor aproximación de lo que necesitaba, ya que es sumamente importante, como ya he mencionado antes, la correcta segmentación para así poder dar información de calidad y que realmente sirva a tomar una decisión sobre las acciones a realizar.

Vamos a dar un vistazo a los resultados obtenidos tanto para 1 solo gas, como el análisis multivariado:

4.2.1 1 variable

El modelo tarda alrededor de 1-2 minutos en entrenarse y hacer todo el proceso, pero esto es dependiendo a la cantidad de datos que tengamos, por lo que es un tema a consideración ya que, de ser muchos datos de entrenamiento, éste se vería afectado en efectividad y por ese tema ya sería descartado el modelo; enseguida podemos ver el resultado del gas monóxido de carbono (CO) es el siguiente:

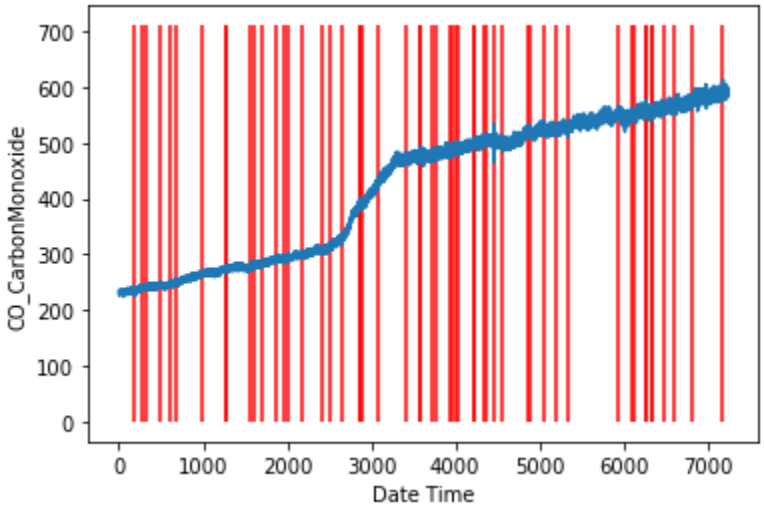

Figura 4.2: *Autoencoders* en Monóxido de Carbono.

Como podemos observar arroja demasiadas alarmas, y solo lo estamos evaluando en una parte no en toda la serie de tiempo como el *Isolation forest*; para segmentar de esta manera no nos fue útil y en todos los gases arroja valores similares por lo que resultó, en general, menos útil que el método anterior y avanzamos al análisis multivariado.

4.2.2 Multivariable:

Con todos los parámetros definidos (mencionados en el capítulo de metodología) procedemos a calcular cómo opera en todos los gases para lo que muestro la siguiente figura:

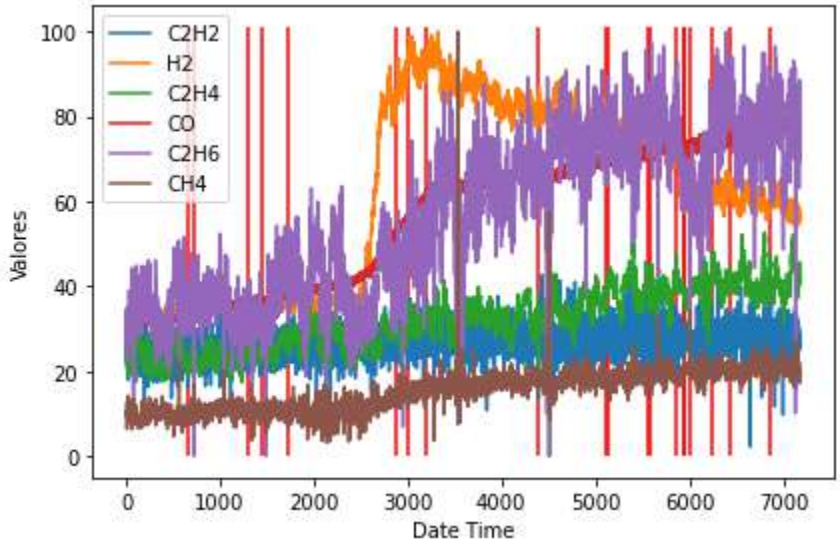

Figura 4.3: *Autoencoders* en multivariado (6 gases).

El modelo funciona con la misma rapidez que una sola variable e incluso disminuyen en gran cantidad las marcas, aunque hay puntos que si se ajustan a la necesidad se puede observar poca precisión y para segmentar gas por gas no puede ser de la misma manera.

Este algoritmo multivariado, ajustando los parámetros un poco más podría indicar puntos importantes que debería tomar en cuenta para hacer algún análisis a futuro, pero por

el momento no es lo que necesito, que es lograr la segmentación certera; por lo que procedí a buscar otro algoritmo/método que me realizara esa parte.

Es importante mencionar que este algoritmo cabe la posibilidad de implementarlo en tiempo real, su desempeño es muy similar al histórico, por lo tanto, tampoco fue de utilidad; podría decirse que estos métodos de ML son especialistas en *anomaly detection* más no son muy útiles en segmentación.

4.3 Programación Dinámica (PD)

4.3.1 Primera fase (tiempo histórico):

Esta primera fase consistió en realizar algunas pruebas para revisar que el algoritmo podía cumplir las expectativas que tenía hasta ahora, por lo que vi que la segmentación se ajustó en gran medida a lo esperado, como prueba está la siguiente figura que muestra su desempeño:

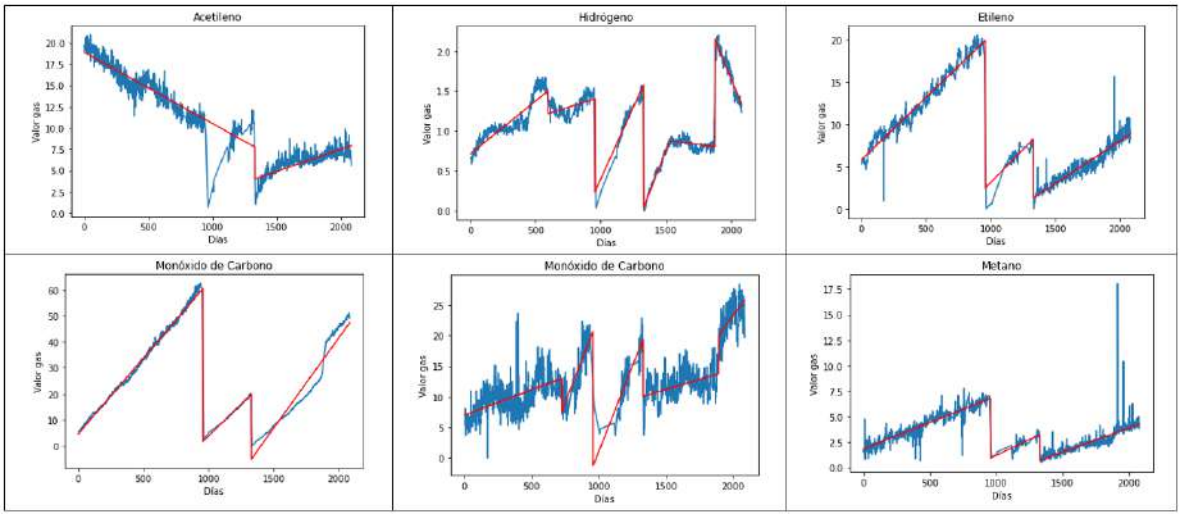

Figura 4.4: Segmentación por programación dinámica en datos históricos.

Luego de comparar los puntos note que era el algoritmo adecuado, pero quedaba otro reto aún mayor, migrar a tiempo real y que este funcionara de la misma manera o incluso mejor, y ajustar el umbral.

Pasaremos al tiempo real y ver cómo se comporta, lo que es la segunda fase, para así poder tomar una decisión y enfocarnos más al tiempo real que es otro de nuestros objetivos.

4.3.2 segunda fase (tiempo real):

A continuación, en la figura 4.2 muestro los puntos donde se segmenta con el algoritmo de PD en tiempo real:

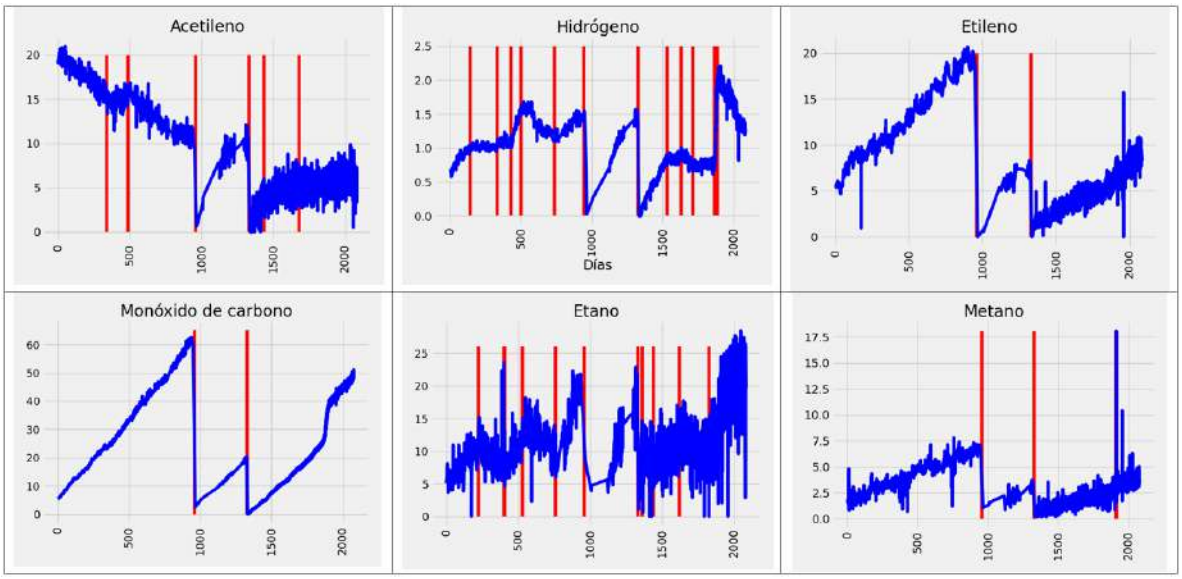

Figura 4.5: Segmentación por programación dinámica en tiempo real.

A simple vista podemos observar que se ajusta muy bien a lo que estábamos esperando, por lo que procedemos a hacer un score para validar su efectividad que presentaremos más adelante.

El algoritmo funciona muy bien, aunque ahora en esta modalidad de tiempo real faltó por detectar algunos puntos importantes, pero en comparación a los otros métodos es mucho más efectivo y preciso; queda trabajar por limpiar los parámetros y recordar que este es un método automático que estará ejecutándose cada que se actualice la base de datos, por lo que podría considerar que en ese momento el algoritmo consideró que el comportamiento y el histórico de ese gas presentó un cambio, por lo que habría que analizar dichos puntos.

4.3.3 Resultados PD

El algoritmo cumple los requerimientos por lo que procedemos a realizar los análisis para determinar las posibles causas que detonan su variabilidad, en la tabla 1 muestro el score obtenido para cada gas:

Tabla 1.6: Score mediante PD en tiempo real.

Score acetileno: 0.857	Score hidrógeno: 0.928	Score Etileno: 0.667
Score Monóxido: 0.50	Score Etano: 0.916	Score Metano: 0.75

Tenemos un promedio total de 0.769 que corresponde a un 76.9% de efectividad, por lo que procedimos a ajustar el umbral para mejorar este score.

Capítulo 5: Discusión y conclusiones

El principal problema en este trabajo fue la segmentación de las series de tiempo ya que la base de datos original viene con diferentes frecuencias de muestreo, es decir, había muestras cada 3 días, muestras por día y hasta 24 muestras por día, esto debido al sistema que toma las mediciones y cambia automáticamente su frecuencia de muestreo debido a la necesidad del horno, si considera que hay un riesgo en algún horno la frecuencia de muestreo incrementa, pero si el horno está "estable" la frecuencia de muestreo disminuye.

Según la literatura revisada y como lo comentan la mayoría de autores en base al tema de segmentación, en los algoritmos que se implementan para este tema, no existe un algoritmo completamente autónomo que se autoajuste a las necesidades de cada serie de tiempo, y para obtener segmentos se necesita especificar el número de segmentos deseado o el umbral para que detecte segmentos automáticamente; sin embargo, para nuestro proyecto de tesis logramos homogeneizar las series de tiempo de cada gas para que funcionara con el mismo umbral.

Hay dos puntos clave en el desarrollo de esta tesis, uno es la segmentación como ya lo comenté anteriormente, y el preprocesamiento de los datos; dado que esta tesis fue enfocada a la implementación de algoritmos en datos reales me fui topando con muchos problemas con los datos que me fueron proporcionados, por lo que personalmente recomiendo, antes de cualquier proyecto, comprender y revisar a detalle cada uno de los datos y verificar que la información sea coherente y funcional para desarrollar el proyecto, de lo contrario ajustarla para poder aplicar ya sea el método o algoritmo correspondiente, si tenemos datos incorrectos de inicio nuestros resultados serán incorrectos.

5.1 Métodos de ML

Los métodos que utilizamos como *Isolation forest* y *autoencoders* son excelentes métodos de detección de anomalías, pero no sirven para generar segmentos como tal, por lo que alguno de estos métodos podría complementar la segmentación por PD y generar alarmas donde ocurra una eventualidad, con la intención de vigilar que no salga de control el gas y mantenerlo controlado.

5.2 Conclusión

Este trabajo nos acerca a lograr el objetivo inicial de predecir y controlar los gases dentro del transformador, el algoritmo desarrollado de PD es muy parecido al algoritmo SWAB (*Sliding Window And Bottom-up)* que mencionan en su artículo (Lovrić, Milanović, & Stamenković, 2014). El cual es una combinación de *Bottom-up* y *Sliding Window,* pero la diferencia radica en que se realiza como forma de *batch* para mantener una eficiencia rápida en nuestro algoritmo.

Después de la experimentación realizada podemos concluir que el el modelo se ajusta a la segmentación esperada en cada gas por encima del 80% de precisión (como pudimos observar en el score del capítulo de resultados).

El trabajo realizado en esta tesis se aplicará dentro de la empresa AMI *automation* ®, lo que resta por realizar lo mencionó en trabajo a futuro, por lo que me siento satisfecho con los resultados obtenidos, que como ya mencioné son la base para concluir dicho proyecto, y quiero resaltar la importancia del científico de datos o analista de datos, de poder identificar y categorizar la información correcta y oportunamente, pues en base a eso dependerá que el modelo/algoritmo funcione o no.

5.3 Trabajo a futuro

Queda pendiente realizar las correlaciones correspondientes de cada segmento generado y arrojar las variables que provocan la fluctuación de cada gas y todo esto implementarlo para todas las plantas que operan de la misma manera, que por su posición geográfica, cultura e infraestructura, son diferentes entre sí. La intención es que el algoritmo funcione de manera uniforme para todas.

Por otro lado, las nuevas tecnologías y desarrollo de nuevos algoritmos son cada vez mayor, y podría desarrollarse un algoritmo automático en un futuro que pasaría a sustituir el implementado en este trabajo por lo que mejoraría la eficacia y eficiencia del actual, pero por el momento esta es la mejor opción.

Referencias

https://www.areametalurgia.com/post/c%C3%B3mo-funciona-un-horno-de-arco-el%C3%A9ctrico

www.ibm.com. (2021). *IBM*. Obtenido de https://www.ibm.com/mx-es/analytics/machine-learning

Bai, B. (2021, October 19). *What is Isolation Forest?* Data Science World. Retrieved September 28, 2022, from https://dsworld.org/what-is-an-isolation-forest/

Bardahl. (2020). *La producción de acero en México: estimación 2020 - Bardahl Industria*. Bardahl Industria. Retrieved September 28, 2022, from https://www.bardahlindustria.com/la-produccion-de-acero-en-mexico-estimacion-2020/

Ge, Z., Song, Z., Ding, S. X., & Huang, B. (2017). Data Mining and Analytics in the process industry: The role of machine learning. *IEEE Access*.

Home. (2022, September 27). YouTube. Retrieved September 28, 2022, from https://mamisymolonas.com/secretos-culinarios/que-son-los-hornos-de-fundicion.html

Indeed. (n.d.). *Dynamic Programming: Characteristics, Methods and Examples*. Indeed. Retrieved September 28, 2022, from https://www.indeed.com/career-advice/career-development/dynamic-programming

medium.com. (26 de junio de 2019). *medium*. Obtenido de Keepler: https://medium.com/@keepler.io/isolation-forest-el-algoritmo-estrella-para-detecci%C3%B3n-de-anomal%C3%ADas-416bb5892f10

Kleinberg, J., & Tardos, E. (2005). *Algorithm Design*. Pearson.

datasmarts.net. (19 de Enero de 2019). *datasmarts.net*. Obtenido de https://datasmarts.net/es/autoencoders/

Millán, J. A. (2014, January 28). *Finanzas - Análisis. México está construido de acero*. El Universal. Retrieved September 28, 2022, from https://archivo.eluniversal.com.mx/finanzas-cartera/2014/impreso/mexico-esta-construido-de-acero-107727.html

Nkonyana, T., Sun, Y., Twala, B., & Dogo, E. (2019). Performance evaluation of data mining techniques in steel manufacturing industry. *Sciencedirect*.

Raschka, S., & Mirjalili, V. (2019). *Aprendizaje automatico con python.* España: Marcombo.

Lovrić, M., Milanović, M., & Stamenković, M. (2014). *ALGORITHMIC METHODS FOR SEGMENTATION OF TIME SERIES: AN OVERVIEW*.

Glosario

1. – Animation: Modulo de la librería matplotlib en el lenguaje Python.
2. – Autoencoders: Método no supervisado de aprendizaje automático.
3. – Automation: Automatización.
4. – Batch: Lote; particiones que se hacen de la base de datos original.
5. - Bottleneck feature: Cuello de botella.
6. - Contamination_rate: rango de contaminación (umbral de los métodos de detección de anomalías)
7. - Data science: Ciencia de datos, Proceso completo de analizar e interpretar grandes cantidades de información.
8. - Deep learning: Aprendizaje profundo.
9. - Electric Arc Furnace: Horno de arco eléctrico.
10. – Electrodo: Es un conductor eléctrico utilizado para hacer contacto con una parte no metálica de un circuito.
11. - Gas online: Base de datos que se obtiene del transformador.
12. - Isolation Forest: Bosque de aislamiento. Método de aprendizaje automático.
13. - Machine learning: Aprendizaje automático.
14. - Penalty_desired: Penalidad deseada (umbral para método de programación dinámica)
15. – Python: Lenguaje de programación.
16. - Random forest: Bosque aleatorio. Modelo de aprendizaje automático.
17. - Redes neuronales: Técnica o modelo de aprendizaje automático.
18. – Streaming: Recolección y transmisión de datos en vivo.
19. - Support Vector Machine: Máquina de soporte de vector. Modelo de aprendizaje automático.
20. - Time.sleep: Modulo para similar el tiempo real en Python.
21. - World Steel Association: Asociación mundial del acero.

MARCO A. ARCIBA MELÉNDEZ

INGENIERO MATEMATICO

Soy una persona proactiva, excelente compañero de equipo, organizado y responsable. Disfruto mucho aprender cosas nuevas, por lo que mi objetivo es un puesto desafiante y dinámico donde poder compartir mi experiencia y a su vez sumar nuevos conocimientos.

CONTACTO

p207345@uach.mx
marcko_03@hotmail.com

EDUCACIÓN

Universidad Autónoma de Chihuahua

Maestría en Ing en Computación, 2022

• Promedio académico 9.4

Universidad Autonoma de Chihuahua

Ing. en matemáticas, 2010

Promedio 8.4

CURSOS / IDIOMAS

Curso de Python avanzado

Pandas, scikit-learn, pytorch, entre otros
Udemy 2020

Idioma Inglés

Nivel Oral 85%.
Nivel Escrito 85%.

EXPERIENCIA LABORAL

Data scientist

AMI, Feb 2020 - Actual

Encargado de aplicar proyecto de predicción y control de los gases dentro del transformador que suministra energía a los hornos de fundición de acero por arco voltaico. Desarrollo y aplicación

Supervisor de almacén

Sonolife, Octubre 2018 - Jul 2020

Recepción y envío de mercancía. Inventarios mensuales. Inventarios cíclicos. Generación de guías para envíos nacionales e internacionales. A cargo del personal del almacén.

Gerente de Plaza

Sixty, Abr 2019 - Sep 2019

Monitoreo de 11 tiendas. Arqueos (auditorias). Check list mensual. Control y mantenimiento de las sucursales. Contacto directo con demás áreas para elaborar y dar seguimiento a estrategias (mercadotecnia, compra, etc.).

Gerente ventas

Coppel, Dic 2014 - Mar 2018

Atención al cliente. Acomodo por lay-out. Inventarios cíclicos y semestrales. Alcance de metas y objetivos de ventas. arqueos y cortes de caja. capacitación a empleados. Manejo de AFORE.