



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

***MEJORA DE LA CALIDAD DE UN PROCESO
MEDIANTE LA DETECCIÓN DE ANOMALÍAS BASADA
EN INTELIGENCIA COMPUTACIONAL***

Autor:

Curiel Alonso, Jon

Tutor:

De La Fuente Aparicio, María Jesús

Departamento de Ingeniería de Sistemas y
Automática

Valladolid, Mayo 2022.



Universidad de Valladolid

MEJORA DE LA CALIDAD DE UN PROCESO MEDIANTE LA DETECCIÓN DE ANOMALÍAS BASADA EN INTELIGENCIA COMPUTACIONAL



ESCUELA DE INGENIERÍAS INDUSTRIALES



Índice de contenidos

RESUMEN	11
ABSTRACT	12
CAPITULO I: INTRODUCCIÓN	13
1. INTRODUCCIÓN AL PROYECTO	15
2. OBJETIVOS DEL PROYECTO	17
3. ORGANIZACIÓN DE LA MEMORIA	18
CAPITULO II: ESTUDIO TEORICO	19
1. CONTROL DE CALIDAD	21
2. DETECCION DE FALLOS POR CONTROL ESTADISTICO DE PROCESOS	23
2.1 VARIABILIDAD EN EL PROCESO DE PRODUCCION	24
2.2 CAUSAS DE LA VARIABILIDAD	25
2.3 GRÁFICAS DE CONTROL	26
2.4 ANALISIS DE COMPONENTES PRINCIPALES (PCA)	29
2.4.1 ESTADISTICOS PARA LA DETECCION DE FALLOS	33
3. DETECCIÓN DE FALLOS CON TÉCNICAS DE MACHINE LEARNING	36
3.1 REDES NEURONALES ARTIFICIALES (ANN)	37
3.1.1 REDES NEURONALES DE PROPAGACION HACIA DELANTE	39
3.1.2 REDES NEURONALES RECURRENTES	41
3.1.3 REDES NEURONALES GRU	43
3.1.4 AUTOENCODERS	44
3.1.4.1 AUTOENCODERS VARIACIONALES	47
CAPITULO III: PLANTA TENNESSEE-EASTMAN	51
1. ORIGEN DE LA PLANTA TENNESSEE EASTMAN	53
2. DESCRIPCION DE LA PLANTA	53
3. DATOS DE LA PLANTA	54



CAPITULO IV: APLICACIÓN DESARROLLADA	59
1. INTRODUCCIÓN DE LA APLICACIÓN PRÁCTICA.....	61
2. DETECCIÓN DE FALLOS MEDIANTE ANÁLISIS DE COMPONENTES PRINCIPALES.....	61
2.1 FUERA DE LÍNEA.....	61
2.2 EN LÍNEA.....	63
3. DETECCIÓN DE FALLOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO.....	67
3.1 REDES NEURONALES DE PROPAGACION HACIA DELANTE. AUTOENCODER.....	67
3.1.1 AUTOENCODER SIMPLE DECRECIENTE-CRECIENTE.....	68
3.1.2 RNN SIMPLE CRECIENTE-DECRECIENTE	72
3.1.3 AUTOENCODER GRU DECRECIENTE-CRECIENTE	75
3.1.3.1 ACTIVACIÓN SIGMOIDE - ACTIVACIÓN RECURRENTE SIGMOIDE	76
3.1.3.2 ACTIVACIÓN TANGENTE HIPERBÓLICA - ACTIVACIÓN RECURRENTE SIGMOIDE	80
3.1.4 RNN GRU CRECIENTE-DECRECIENTE.....	83
3.1.4.1 ACTIVACIÓN SIGMOIDE - ACTIVACIÓN RECURRENTE SIGMOIDE	84
3.1.4.2 ACTIVACIÓN TANGENTE HIPERBÓLICA - ACTIVACIÓN RECURRENTE SIGMOIDE	87
3.1.5 AUTOENCODER VARIACIONAL DECRECIENTE-CRECIENTE	90
3.1.5.1 ACTIVACION SIGMOIDE	91
3.1.5.2 ACTIVACION RELU	94
3.1.6 RNN VARIACIONAL CRECIENTE-DECRECIENTE.....	98
3.1.6.1 ACTIVACION SIGMOIDE	98
3.1.6.2 ACTIVACION RELU	101
3.1.7 AUTOENCODER VARIACIONAL RECURRENTE DECRECIENTE-CRECIENTE GRU.....	104
3.1.8 RNN VARIACIONAL RECURRENTE CRECIENTE-DECRECIENTE GRU.....	109



CAPITULO V: CONCLUSIONES Y TRABAJO FUTURO	115
1. CONCLUSIONES.....	117
1.1 TABLA COMPARATIVA DE RESULTADOS ESTADISTICO T ²	117
1.2 TABLA COMPARATIVA DE RESULTADOS ESTADISTICO Q.....	120
2. TRABAJO FUTURO	123
BIBLIOGRAFÍA	125



Índice de ilustraciones

Ilustración 1 Ciclo PDCA [4].....	22
Ilustración 2. Variabilidad del proceso. (a) Toma de muestras. (b) Distribuciones. (c) Variación de distribuciones. (d) Distribución con causas comunes de variación. (e) Distribución con causas especiales de variación [7].....	24
Ilustración 3. Tipos de salida del proceso. (a) Proceso fuera de control con causas especiales de variación. (b) Proceso bajo control con causas comunes de variación, pero fuera de los límites de especificación. (c) Proceso bajo control con causas comunes de variación y dentro de los límites de especificación o calidad establecidos. [7]	25
Ilustración 4. Relación de los límites de control con la distribución de muestreo [7]	27
Ilustración 5 Gráficas de control [7]	28
Ilustración 6 Grafica de control Shewhart [8].....	28
Ilustración 7 Interpretación geométrica PCA [11]	30
Ilustración 8 Tabla de distribución F con $\alpha = 0.01$ [12].....	34
Ilustración 9 Tabla de distribución F con $\alpha = 0.05$ [12].....	34
Ilustración 10 Red neuronal de 2 capas [16].....	38
Ilustración 11 Neurona artificial [16]	39
Ilustración 12 Diagrama de una neurona artificial [8].....	39
Ilustración 13 Perceptrón [17].....	40
Ilustración 14 Red neuronal feedforward multicapa [18]	40
Ilustración 15 Diagrama cíclico de una red neuronal recurrente [20]	42
Ilustración 16 Diagrama de red GRU [22].....	43
Ilustración 17 Estructura de un Autoencoder [25].....	45
Ilustración 18 Estructura de SAe con K capas [25]	45
Ilustración 19 Entrenamiento y unión de SAe [25]	46
Ilustración 20 Arquitectura Autoencoder Variacional [27]	48
Ilustración 21 Diferencias de funcionamiento entre Autoencoder tradicional y Autoencoder variacional [26]	48
Ilustración 22 Proceso Tennessee Eastman [8].....	54
Ilustración 23 Estadísticos con los datos de comportamiento normal de la planta	63
Ilustración 24 Detección del fallo 1 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q	63
Ilustración 25 Detección del fallo 7 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q	64
Ilustración 26 Detección del fallo 3 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q	65
Ilustración 27 Detección del fallo 21 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q	65
Ilustración 28 Detección del fallo 1 mediante Autoencoder decreciente-creciente. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q.....	69
Ilustración 29 Detección del fallo 7 mediante Autoencoder decreciente-creciente. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q.....	69
Ilustración 30 Detección del fallo 21 mediante Autoencoder decreciente-creciente. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q.....	70
Ilustración 31 Detección del fallo 1 mediante RNN creciente-decreciente. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q.....	73
Ilustración 32 Detección del fallo 7 mediante RNN creciente-decreciente. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q.....	73



Ilustración 33 Detección del fallo 21 mediante RNN creciente-decreciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 74

Ilustración 34 Detección del fallo 1 mediante Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 77

Ilustración 35 Detección del fallo 7 mediante Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 77

Ilustración 36 Detección del fallo 21 mediante Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 78

Ilustración 37 Detección del fallo 1 mediante Autoencoder decreciente-creciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 80

Ilustración 38 Detección del fallo 7 mediante Autoencoder decreciente-creciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 81

Ilustración 39 Detección del fallo 21 mediante Autoencoder decreciente-creciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 81

Ilustración 40 Detección del fallo 1 mediante RNN creciente-decreciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 84

Ilustración 41 Detección del fallo 7 mediante RNN creciente-decreciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 85

Ilustración 42 Detección del fallo 21 mediante RNN creciente-decreciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 85

Ilustración 43 Detección del fallo 1 mediante RNN creciente-decreciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 87

Ilustración 44 Detección del fallo 7 mediante RNN creciente-decreciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 88

Ilustración 45 Detección del fallo 21 mediante RNN creciente-decreciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 88

Ilustración 46 Detección del fallo 1 mediante Autoencoder Variacional decreciente-creciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 91

Ilustración 47 Detección del fallo 7 mediante Autoencoder Variacional decreciente-creciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 92

Ilustración 48 Detección del fallo 21 mediante Autoencoder Variacional decreciente-creciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 92

Ilustración 49 Detección del fallo 1 mediante Autoencoder Variacional decreciente-creciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 95



Ilustración 50 Detección del fallo 7 mediante Autoencoder Variacional decreciente-creciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 95

Ilustración 51 Detección del fallo 21 mediante Autoencoder Variacional decreciente-creciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 96

Ilustración 52 Detección del fallo 1 mediante RNN Variacional creciente-decreciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 99

Ilustración 53 Detección del fallo 7 mediante RNN Variacional creciente-decreciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 99

Ilustración 54 Detección del fallo 21 mediante RNN Variacional creciente-decreciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 100

Ilustración 55 Detección del fallo 1 mediante RNN Variacional creciente-decreciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 102

Ilustración 56 Detección del fallo 7 mediante RNN Variacional creciente-decreciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 102

Ilustración 57 Detección del fallo 21 mediante RNN Variacional creciente-decreciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 103

Ilustración 58 Detección del fallo 1 mediante Autoencoder Variacional decreciente-creciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 106

Ilustración 59 Detección del fallo 7 mediante Autoencoder Variacional decreciente-creciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 106

Ilustración 60 Detección del fallo 21 mediante Autoencoder Variacional decreciente-creciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 107

Ilustración 61 Detección del fallo 1 mediante RNN Variacional creciente-decreciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 111

Ilustración 62 Detección del fallo 7 mediante RNN Variacional creciente-decreciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 111

Ilustración 63 Detección del fallo 21 mediante RNN Variacional creciente-decreciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q..... 112



Índice de tablas

Tabla 1 Variables manipuladas del proceso [8] 55

Tabla 2 Variables medidas del proceso [8]..... 56

Tabla 3 Fallos posibles en el proceso [8]..... 57

Tabla 4 Resultados de la detección de fallos mediante PCA..... 66

Tabla 5 Resultados de la detección de fallos mediante Autoencoder decreciente-creciente 71

Tabla 6 Resultados de la detección de fallos mediante RNN creciente-decreciente 75

Tabla 7 Resultados de la detección de fallos mediante Autoencoder decreciente-creciente GRU sigmoid-sigmoid 79

Tabla 8 Resultados de la detección de fallos mediante Autoencoder decreciente-creciente GRU tanh-sigmoid..... 82

Tabla 9 Resultados de la detección de fallos mediante RNN creciente-decreciente GRU sigmoid-sigmoid 86

Tabla 10 Resultados de la detección de fallos mediante RNN creciente-decreciente GRU tanh-sigmoid 89

Tabla 11 Resultados de la detección de fallos mediante Autoencoder Variacional decreciente-creciente activación sigmoide..... 94

Tabla 12 Resultados de la detección de fallos mediante Autoencoder Variacional decreciente-creciente activación ReLU 97

Tabla 13 Resultados de la detección de fallos mediante RNN Variacional creciente-decreciente activación sigmoide 101

Tabla 14 Resultados de la detección de fallos mediante RNN Variacional creciente-decreciente activación ReLU..... 104

Tabla 15 Resultados de la detección de fallos mediante Autoencoder Variacional recurrente GRU decreciente-creciente..... 108

Tabla 16 Resultados de la detección de fallos mediante RNN Variacional recurrente GRU creciente-decreciente 113

Tabla 17 Tabla comparativa de resultados tiempo de detección mediante Estadístico T^2 ... 118

Tabla 18 Tabla comparativa de resultados tasa de detección mediante Estadístico T^2 120

Tabla 19 Tabla comparativa de resultados tiempo de detección mediante Estadístico Q... 122

Tabla 20 Tabla comparativa de resultados tasa de detección mediante Estadístico Q 123





RESUMEN

Este trabajo tiene como objetivo mejorar la calidad de un proceso de una planta industrial mediante distintas técnicas de detección y diagnóstico de fallos (FDD) basadas en datos. El uso de estas técnicas se basa en el avance de la industria y la tecnología que permiten la recogida de datos/información de los procesos a gran escala.

El primer método FDD que se utilizará será el Análisis de Componentes Principales (PCA), una técnica estadística multivariante, que reduce la dimensionalidad del proceso y permite detectar los fallos en la planta, es decir nos predice el comportamiento del proceso, indicando si trabaja en condiciones normales o existen fallos/anomalías en el proceso.

El creciente auge de la inteligencia artificial, y en particular el ‘Deep Learning’ o aprendizaje profundo, nos permite usar un segundo método FDD, en el que se utilizarán redes neuronales de propagación hacia adelante y en concreto Autoencoders. Se utilizarán varios tipos de Autoencoders: Simples, recurrentes GRU, variacionales y variacionales recurrentes para encontrar la mejor técnica para detectar los fallos de la planta.

Estas técnicas FDD se aplican sobre la planta química Tennessee Eastman de la que se obtienen datos de comportamiento normal y datos de fallo. Esta planta es un benchmark de la literatura científica, usado para probar métodos de control y de detección y diagnóstico de fallos. La planta contiene 52 variables del proceso y 21 fallos posibles que pueden ocurrir en el proceso.

Una vez desarrolladas todas las técnicas se obtiene una conclusión comparando todas ellas y se realiza un estudio de trabajo futuro para seguir mejorando en este trabajo.



ABSTRACT

This work aims to improve the quality of a process in an industrial plant by means of different fault detection and diagnosis (FDD) techniques based on data. The use of these techniques is based on the development of the industry and the technology which allows the collection of data and information from the large-scale processes.

The first FDD method that will be used is the Principal Component Analysis (PCA), a multivariate statistics technique which reduces the dimensionality of the process and allows the detection of faults in the facility. This is, it predicts the behaviour of the process, telling us whether it is working on normal conditions or if there are any anomalies in it.

The crescent rise in artificial intelligence, and more precisely what is known as “Deep Learning”, allows the use of a second FDD method, in which forward propagation neural networks called Autoencoders will be used. There will be different types of these that will be used: simple ones, GRU recurrents, variational and recurrent variational. All these in order to find the best detection technique for this facility.

These FDD techniques are applied on the chemical facility called Tennessee Eastman in which normal behaviour and fault data are found. This facility is a benchmark in scientific literature, used for the trial of several control and FDD methods. The facility has 52 variables and 21 possible faults that can take place in the process.

Once all these techniques are developed, there will be a conclusion comparing all of them. As well as a possible future works study in order to continue improving this work.



CAPITULO I: INTRODUCCIÓN





1. INTRODUCCIÓN AL PROYECTO

El control de calidad es fundamental en cualquier proceso industrial ya que permite hacer un seguimiento a las acciones productivas y así eliminar errores, fallas o defectos con el objetivo final de conseguir que el producto final cumpla unas especificaciones definidas [1].

Para un control de calidad robusto es esencial una rápida detección de fallos, así como, un diagnóstico eficaz de estos.

La detección de fallos identifica la ocurrencia de fallas que causan anomalías en el producto haciendo que el producto este fuera del límite de las especificaciones, además localizan el fallo en la planta y determinan las causas por las que se ha producido el fallo y las consecuencias que puede tener tanto en el producto como en el proceso.

Desde siempre los procesos productivos han experimentado cambios ocasionados por los avances de la tecnología. En estos momentos, los sistemas industriales están inmersos en el concepto de industria 4.0 que surgió en Alemania y hace referencia a la cuarta Revolución Industrial y consiste en interconectar todas las partes de una planta dando lugar a una automatización efectiva [2].

Los fabricantes están integrando nuevas tecnologías, incluida el Internet de las cosas (IoT), la computación y la analítica en la nube además de la Inteligencia Artificial junto con ‘machine learning’ o aprendizaje automático, tanto en sus instalaciones de producción como a lo largo de sus operaciones.

Estas fábricas inteligentes están equipadas con sensores avanzados, software y robótica incorporados que recopilan y analizan datos para obtener la mayor información posible de la planta en tiempo real.

Estas tecnologías digitales permiten una mayor automatización, un mantenimiento predictivo, una optimización automática de las mejoras de procesos y, sobre todo, un nuevo nivel de eficiencia y capacidad de respuesta a clientes que antes no era posible, lo que provoca que se incremente la productividad y se mejore la calidad del producto [3].

Estas tecnologías recogen una gran cantidad de datos heterogéneos y multivariantes, lo que es conocido como “Big Data”. Para tratar y analizar esta cantidad de datos se utilizan distintas técnicas como son el control estadístico de procesos (SPE) o bien el “machine learning” o “aprendizaje automático”, que son unas nuevas herramientas para el tratamiento de datos.



Como se vio anteriormente, la calidad y también la seguridad en la producción industrial son dos objetivos importantes para cualquier industria moderna, lo que se consigue con la implantación de métodos de detección y de diagnóstico de fallos que detecten cualquier anomalía que aparezca en la planta. Por lo que, uniendo estas dos ideas, en este proyecto se va a usar distintos métodos de inteligencia computacional para la detección y diagnóstico de fallos en una planta industrial.

Para la detección de fallos se va a utilizar una serie de técnicas multivariantes en la que se encuentra el control estadístico de procesos (SPE) que es la técnica más utilizada a escala industrial, donde se encuentra el Análisis de Componentes Principales (PCA) mediante la cual se reduce la dimensionalidad para obtener un número menor de variables de las que tenía la planta original, las cuales se llaman componentes principales, que van a ser una combinación lineal de las variables iniciales además de que en ellas se encuentran la mayor parte de la información de los datos. Dentro de PCA se utilizan dos estadísticos, para detectar anomalías, el primero (Estadístico de Hotelling T^2) para monitorizar la variabilidad en el subespacio definido por los componentes principales generado por PCA y el segundo (Estadístico Q o SPE) indica la diferencia entre la observación y la proyección en el espacio reducido de los componentes principales.

Otra de las técnicas que se va a utilizar para la detección de fallos es una técnica que se está empezando a implantar en la industria como son las técnicas de aprendizaje automático, que se dividen en aprendizaje supervisado y no supervisado.

El aprendizaje supervisado podría definirse como un tipo de aprendizaje en IA en el que un algoritmo es entrenado con variables que incluyen los valores que queremos predecir; a estos valores conocidos se les llama “etiquetas” y se usan también para la evaluación del modelo. El aprendizaje supervisado se puede subdividir en dos tipos: clasificación y regresión. El aprendizaje no supervisado se diferencia del supervisado en que basan su entrenamiento en datos sin etiquetas definidas.

Como técnicas de ‘machine learning’ en este proyecto se van a utilizar “Redes Neuronales” para procesar la gran cantidad de datos que se tiene (‘Deep Learning’), dentro de estas Redes Neuronales se van a utilizar un tipo de redes llamado ‘Autoencoder’.

Las redes neuronales artificiales (ANN) son un modelo inspirado en el funcionamiento del cerebro humano. Está formado por un conjunto de nodos conocidos como neuronas artificiales que están conectadas y transmiten señales entre sí. Estas señales se transmiten desde la entrada hasta generar una salida.



El 'Deep Learning' lleva a cabo el proceso de 'Machine Learning' usando una red neuronal artificial que se compone de un número de niveles jerárquicos. En el nivel inicial de la jerarquía la red aprende algo simple y luego envía esta información al siguiente nivel. El siguiente nivel toma esta información sencilla, la combina, compone una información algo un poco más compleja, y se lo pasa al tercer nivel, y así sucesivamente.

Un 'Autoencoder' es un tipo especial de red neuronal que está entrenada para copiar su entrada a su salida.

Dentro de estas técnicas de aprendizaje automático, en primer lugar, se realizará un Autoencoder sencillo para la detección de fallos en la planta, para a continuación utilizar una Red Neuronal Recurrente, una red GRU, que es una red que resuelve el problema de dependencia a largo plazo, además de que la velocidad de entrenamiento es rápida.

Por último, se aplicará otro tipo de Autoencoder como es el Autoencoder Variacional que es un modelo de aprendizaje que mezcla redes neuronales con distribuciones de probabilidad.

Con este tipo de Autoencoder se busca mejorar la detección de fallos tanto de PCA como de los Autoencoders utilizados anteriormente.

2. OBJETIVOS DEL PROYECTO

En este proyecto se va a plantear el desarrollo de técnicas de detección y diagnóstico de fallos (FDD) para mejorar la calidad en un proceso detectando y diagnosticando los fallos que se produzcan durante el proceso de la forma más rápida posible. Para ello se van a utilizar distintas técnicas aplicables a todo tipo de industrias, ya que en todas se pueden producir fallos que afectan a la fiabilidad del proceso, la seguridad de la planta, y a la calidad de los productos.

Para la detección del fallo o anomalía en la planta se utilizarán técnicas de control estadístico de procesos, y en concreto el análisis de componentes principales, así como técnicas basadas en datos, y en concreto, técnicas de inteligencia computacional como son las redes neuronales y en concreto Autoencoders. Además, se hará una comparativa de los resultados entre las distintas técnicas para sacar las conclusiones que correspondan.

Los datos de la planta de este proyecto provienen de la planta Tennessee-Eastman, que es una planta química usada como benchmark en la literatura científica, y en la que se realizan pruebas de monitorización de procesos multivariantes.



3. ORGANIZACIÓN DE LA MEMORIA

Para organizar la memoria del trabajo se separará en seis capítulos.

El primero será una pequeña introducción sobre el tema del que va a tratar el proyecto hablando sobre las distintas técnicas de la detección de fallos, así como los objetivos a conseguir en la realización de este proyecto.

En el segundo se profundiza de manera teórica en el control de calidad y en las técnicas de detección y diagnóstico de fallos tanto mediante el control estadístico de procesos como mediante inteligencia computacional.

En el tercer capítulo se habla sobre la planta Tennessee-Eastman explicando cómo surgió, así como las variables de la planta y los fallos que se van a producir en dicha planta y que se van a intentar detectar.

En el cuarto se describe la aplicación práctica del proyecto, explicando paso a paso el primer método a utilizar para la detección de fallos como es el Análisis de Componentes Principales, después se describirán las técnicas basadas en inteligencia computacional que se van a utilizar para la detección de fallos como son los distintos tipos de redes neuronales y Autoencoder que se van a utilizar.

En el quinto se explican las conclusiones a las que se ha llegado a partir de los resultados obtenidos en el capítulo anterior. También se comentan sugerencias para futuros estudios sobre este tema.

El último capítulo se utilizará para documentar la bibliografía utilizada en este trabajo.



CAPITULO II: ESTUDIO TEORICO





1. CONTROL DE CALIDAD

El interés de la sociedad por la calidad es tan antiguo como el origen de las sociedades humanas, por lo que tanto el concepto como las formas de gestionar la calidad han ido evolucionando progresivamente.

El desarrollo de la producción en masa, la especialización, el incremento en la complejidad de los procesos de producción y la introducción de la economía de mercado centrada en la competencia y en la necesidad de reducir los precios, hecho que implica reducir costes de materiales y de proceso, determinó la puesta en marcha de métodos para mejorar la eficiencia de las líneas de producción.

Así mismo, el aumento del uso de la tecnología obligó a que la calidad fuera controlada mediante el desarrollo de métodos de supervisión más específicos como el establecimiento de especificaciones escritas o el desarrollo de estándares.

Este desarrollo de métodos es lo que se conoce como control de calidad. El empleo de estas técnicas permitió un mayor control de la estandarización del producto fabricado.

El desarrollo de estos métodos fue impulsado por las necesidades de la industria de armamento, que, al precisar un gran número de componentes, potenciaron la introducción de la estandarización. Este es el inicio del establecimiento de estándares militares en los EEUU denominados Z-1, o los Estándares Británicos. Posteriormente se establecieron estándares en otras áreas de la ingeniería, construcción e industria química.

Se introdujeron elementos de medida (dispositivos de medición) y de aplicación de técnicas estadísticas en las actividades de inspección y control, con el fin de poder disminuir los costes de inspección restringiendo la inspección a un número significativo de muestras del producto.

Para el desarrollo de estas técnicas fue importante la aportación de Shewhart, quien aplicó los conceptos de la estadística a los problemas de la calidad, estableciendo el concepto de variabilidad y por tanto el de tolerancias. Así mismo, Shewhart introdujo los gráficos de control para conocer la variabilidad y causas asignables. Estas gráficas de control se aplicaban a cada fase del proceso, lo que permitía una respuesta rápida al cambio en la conducta del proceso (causas asignables).

Durante la segunda mitad del siglo XX, W. Edwards Deming popularizó el ciclo PDCA (Plan, Do, Check, Act), inicialmente desarrollado por Shewhart, que es utilizado extensamente en los ámbitos de la gestión de la calidad en la actualidad [4].

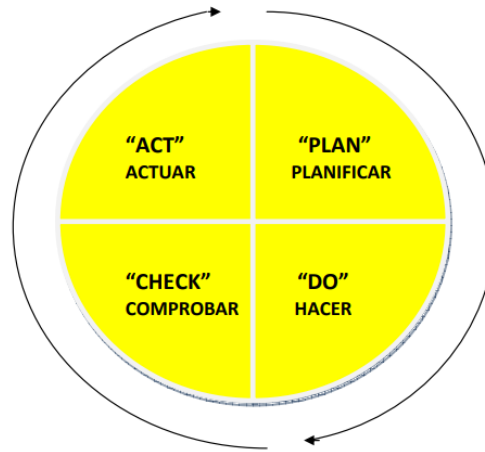


Ilustración 1 Ciclo PDCA [4]

En la actualidad los modelos de gestión de la mayoría de las empresas toman como puntos de referencia las normas ISO, que especifican requerimientos que las organizaciones implementan a fin de cumplir sus objetivos en términos de calidad, medio ambiente, prevención de riesgos laborales, seguridad y responsabilidad social.

El control de calidad tiene las siguientes fases: Monitorización del proceso, en el que se obtiene la información del proceso y se incluye la detección y análisis de los fallos, en el que se detectan y se diagnostican los fallos, y Actuación sobre el proceso en el que se retorna el proceso a la normalidad tras aplicar las medidas correspondientes.

Para un buen control de calidad son necesarios instrumentos de medida precisos, un robusto sistema de comunicación industrial para la toma y comunicación de los datos, una temprana detección de las anomalías y un correcto diagnóstico de los fallos.

Un esquema de detección y diagnóstico de fallos (FDD), que se incluye en la fase de monitorización del proceso del control de calidad, permite incrementar la fiabilidad de la planta.

El proceso de detección y diagnóstico de fallos (FDD) se divide en las siguientes etapas:

- *Detección del fallo*
- *Identificación del fallo:* Para buscar que variables de la planta son responsables del fallo



- *Diagnóstico del fallo:* Identificar tipo y causa del fallo
- *Recuperación del sistema:* Retornar la planta al funcionamiento normal [5]

Existen varias técnicas para aplicar a la detección y diagnóstico de fallos (FDD), el más utilizado es el Control Estadístico de Procesos (SPC), aunque en la actualidad se están desarrollando las técnicas de 'Machine Learning' para utilizarlas en las plantas industriales modernas y tratar la gran cantidad de datos que hay en ellas.

2. DETECCION DE FALLOS POR CONTROL ESTADISTICO DE PROCESOS

El Control Estadístico de Procesos (SPC) es la aplicación de técnicas estadísticas para comprobar la estabilidad de un proceso productivo y realizar una fácil identificación de posibles problemas en la planta.

El Control Estadístico de Procesos fue desarrollado por Dr. Walter A. Shewhart en los años 20 identificando dos causas posibles de variaciones de un proceso: causas asignables y causas aleatorias [6].

A partir de identificar estas variaciones de los procesos desarrolló las gráficas de control para distinguir las posibles causas de variaciones dentro del proceso productivo. A partir de estas gráficas desarrolladas por él comprobó que poner el proceso productivo bajo control estadístico de las causas aleatorias de fallos le permitía limitar la producción no conforme.

El Control Estadístico de Procesos también se utiliza para alertar al personal pertinente de anomalías en los procesos para aplicar las medidas y soluciones adecuadas para retornar el proceso al funcionamiento normal, además se usa para informar a la gerencia sobre los cambios introducidos en los procesos que hayan repercutido favorablemente en la producción resultante de dichos procesos.

Para entender la necesidad del Control Estadístico de Procesos hay que partir de las altas fuentes de variación de un proceso, ya que, aunque el proceso se desarrolle de la forma prevista no habrá dos productos finales exactamente iguales.

2.1 VARIABILIDAD EN EL PROCESO DE PRODUCCION

Todos los procesos tienen variaciones, pero mientras unos muestran solo variaciones naturales del proceso (causas comunes de variación), estando bajo control estadístico, otros presentan variaciones a mayores de las inherentes del proceso (causas especiales de variación), y se denominan fuera de control.

Las causas comunes de variación son fuentes normales de variación, actuando de forma consistente en el proceso y al acumularse dan lugar a una distribución estadística estable y replicable a lo largo del tiempo. Las causas especiales de variación pueden ser asignadas a una perturbación concreta del proceso, y por naturaleza son impredecibles e intermitentes.

Para determinar la variabilidad del proceso se toman una serie de pequeñas muestras y se las coloca en una escala proporcional, eje horizontal de la Ilustración 2 (a) y después se indica la frecuencia que ocurre cada muestra, eje vertical de la Ilustración 2 (a). Después de un número notable de muestras, si el patrón es estable, se le denomina distribución como se observa en la Ilustración 2 (b). Las distribuciones pueden diferir en función de lo que revelan las muestras como se observa en la Ilustración 2 (c).

Si solo se encuentran causas comunes de variación, la salida del proceso forma una distribución que es estable a través del tiempo y es predecible como se observa en la Ilustración 2 (d). Si aparecen causas especiales de variación, la salida del proceso no es estable a través del tiempo y no es predecible como se observa en la Ilustración 2 (e).

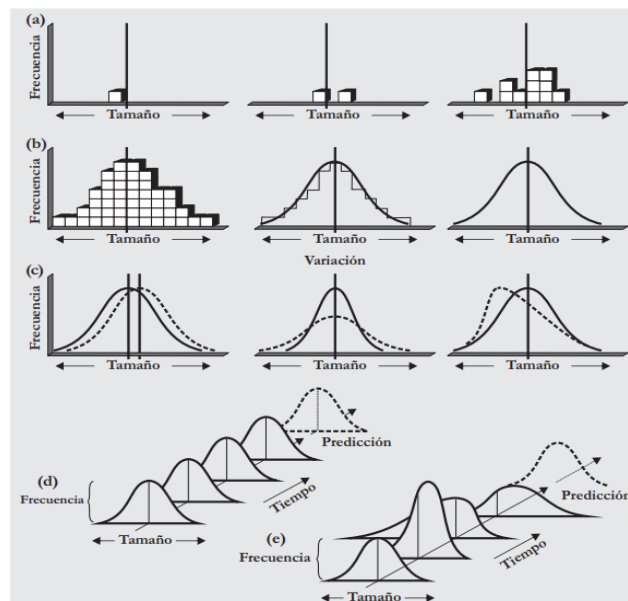


Ilustración 2. Variabilidad del proceso. (a) Toma de muestras. (b) Distribuciones. (c) Variación de distribuciones. (d) Distribución con causas comunes de variación. (e) Distribución con causas especiales de variación [7]

El objetivo por tanto es conseguir eliminar esas causas especiales de variación en el proceso, o si fuera posible, eliminarlas y mantener el proceso bajo control.

En la Ilustración 3 se muestran tres tipos de salidas del proceso. La Ilustración 3 (a) muestra un proceso fuera de control debido a causas especiales de variación. La Ilustración 3 (b) muestra un proceso bajo control ya que solo ocurren causas comunes de variación pero que se encuentra fuera de los límites de especificación. La Ilustración 3 (c) muestra un proceso bajo control con causas comunes de variación y que se encuentra dentro de los límites de calidad establecidos.

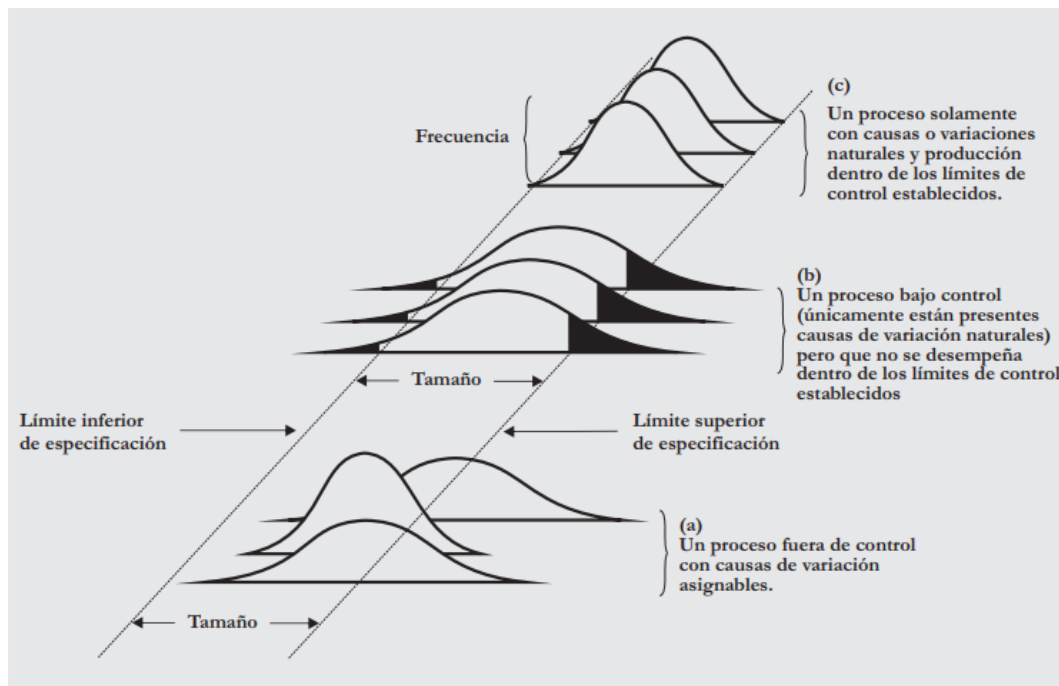


Ilustración 3. Tipos de salida del proceso. (a) Proceso fuera de control con causas especiales de variación. (b) Proceso bajo control con causas comunes de variación, pero fuera de los límites de especificación. (c) Proceso bajo control con causas comunes de variación y dentro de los límites de especificación o calidad establecidos. [7]

2.2 CAUSAS DE LA VARIABILIDAD

Existen dos tipos de causas de variación que afectan a un proceso, las causas comunes de variación y las causas especiales de variación.

- Causas comunes de variación

Son fuentes de variación puramente aleatorias, no identificables e imposibles de evitar mientras se utilice el procedimiento actual. Estas causas no provocan la pérdida de control del proceso.



Estas causas comunes forman una distribución que se caracteriza por su media, su varianza y su forma.

- La media es la suma de las observaciones dividida entre el número total de observaciones.
- La varianza es una medida de la dispersión de las observaciones en torno a la media.
- La forma de la distribución puede ser simétrica o asimétrica, la distribución simétrica presenta el mismo número de observaciones por encima que por debajo de la media, mientras que la asimétrica presenta distinto número de observaciones por debajo que por encima de la media.

Si la variabilidad del proceso proviene únicamente de causas comunes de variación; la suposición típica es que se trata de una distribución simétrica donde la mayoría de las observaciones se localiza cerca del centro.

- **Causas especiales de variación**

Las causas especiales de variación no son inherentes al proceso, provocan la pérdida de control del proceso, y por lo tanto deben ser identificadas y eliminadas. Este tipo de causas debe ser corregido para devolver al proceso al funcionamiento normal.

El Control Estadístico de Procesos se utiliza en procesos que están bajo control estadístico para detectar el momento en el que surgen causas especiales de variación para eliminarlas y devolver el proceso al control estadístico.

2.3 GRÁFICAS DE CONTROL

Para determinar si las variaciones observadas no son normales se puede trazar la gráfica de la característica de calidad tomada de la muestra en un diagrama ordenado por tiempo, lo que se conoce como gráfica de control.

La grafica de control tiene una línea central (LC), que es el objetivo que se desea alcanzar con el proceso, y dos límites de control basados en la distribución del muestreo, límite superior de control (UCL) y límite inferior de control (LCL).

Una distribución de muestras ubicada entre ambos limites indica que el proceso muestra causas comunes de variación, sin embargo, una distribución fuera de los límites de control indica que han ocurrido causas especiales de variación en el proceso.

En la Ilustración 4 se muestra la relación de los límites de control con la distribución de muestreo en tres muestras.

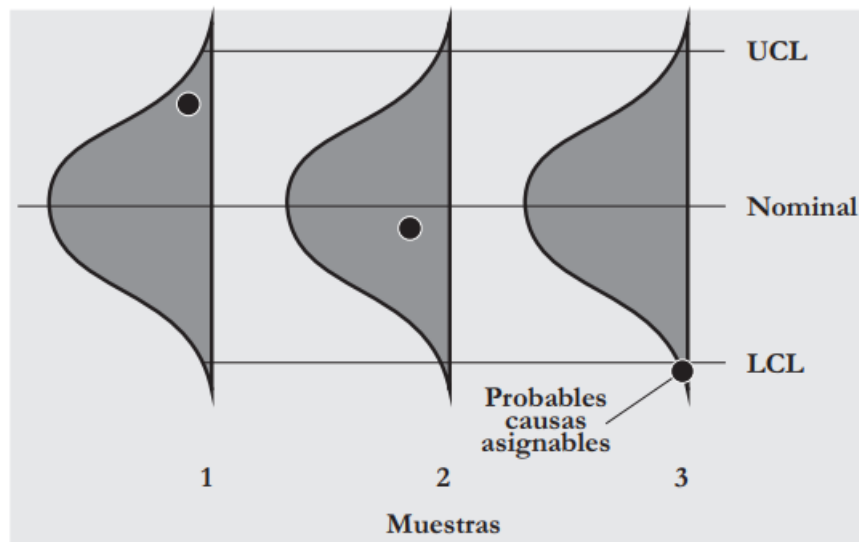


Ilustración 4. Relación de los límites de control con la distribución de muestreo [7]

Las gráficas de control se suelen utilizar mediante el siguiente procedimiento:

1. Tomar una muestra aleatoria del proceso, medir la característica de la calidad y calcular una medida variable.
2. Si la estadística se ubica fuera de los límites de control, buscar una causa especial de variación.
3. Eliminar la causa si degrada la calidad, o incorporar la causa si con ella mejora la calidad, si es así reconstruir la gráfica de control con los nuevos datos.
4. Repetir periódicamente todo el procedimiento.

Pueden ocurrir dos tipos de errores cuando se utiliza este tipo de gráficas.

El error 'tipo I' (rechazar un lote de buena calidad) se produce cuando se concluye que el proceso está fuera de control, basándose en un resultado de muestra ubicado fuera de los límites de control, cuando se trataba de un efecto puramente aleatorio.

El error 'tipo II' (aceptar un lote de mala calidad) se produce cuando se concluye que el proceso está bajo control y que únicamente presenta discrepancias aleatorias, cuando en realidad dicho proceso está fuera de control estadístico.

En la Ilustración 5 se muestran distintas gráficas de control con situaciones anormales en el comportamiento de las muestras respecto al comportamiento normal. [7]

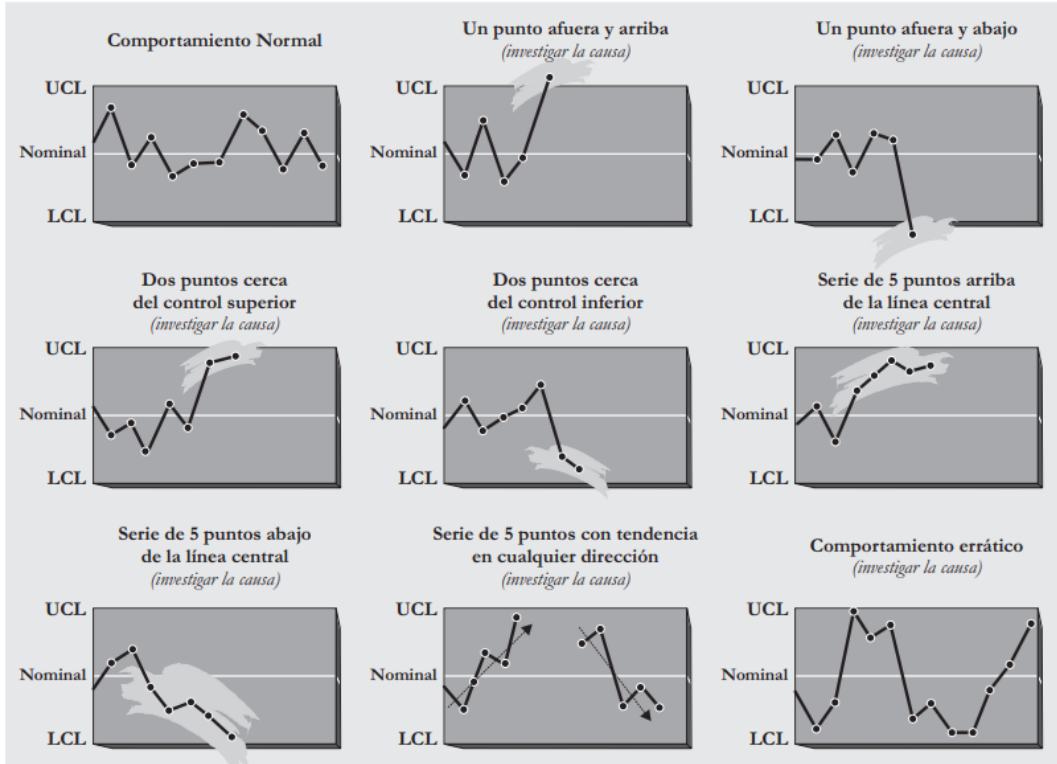


Ilustración 5 Gráficas de control [7]

Según Shewhart, la forma de una gráfica de control corresponde con la Ilustración 6 donde los límites de control y la línea central vienen definidas por las siguientes ecuaciones:

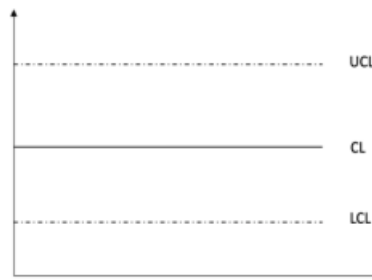


Ilustración 6 Gráfica de control Shewhart [8]

$$UCL = \mu_T + k\sigma_T \quad (1)$$

$$CL = \mu_T \quad (2)$$

$$LCL = \mu_T - k\sigma_T \quad (3)$$



En el que:

T es el estadístico muestral que mide la característica de calidad

μ_T es la media del estimador T

σ_T es la desviación típica del estimador T

k es una constante que indica la distancia entre el valor central y los límites de control expresada en unidades de desviación típica σ_T [8]

Después de estas graficas de control desarrolladas por Shewhart aparecieron los gráficos CUSUM que se basan en la representación de la acumulación de las desviaciones de cada observación respecto a un valor de referencia. [9]

Este tipo de gráficos detectan pequeñas desviaciones del estado de control más rápidamente que las gráficas de Shewhart.

Estas gráficas cuentan con los siguientes elementos:

- Observaciones normalizadas, Z_i , centradas alrededor de la media y baremadas con la desviación estándar

$$Z_i = \frac{\bar{X}_i - \bar{x}}{\sigma_i} \quad (4)$$

- Sumas acumuladas, S_H , que detecta una anomalía positiva, y S_L , que detecta una anomalía negativa

$$S_{H_{i+1}} = \max(0, S_{H_i} + Z_i - k) \quad (5)$$

$$S_{L_{i+1}} = \max(0, S_{L_i} - Z_i - k) \quad (6)$$

- Valor de referencia, k, definido de forma que detecte una variación de la mitad de la desviación estándar
- Umbral, h, para detectar fallo y dar la alarma, definido según S_H y S_L

Las desventajas de este tipo de graficas es la lentitud para detectar cambios grandes en el proceso, y que, al ser sumas acumuladas existe una correlación entre los datos y es difícil interpretar patrones. [10]

2.4 ANALISIS DE COMPONENTES PRINCIPALES (PCA)

El Análisis de Componentes Principales (PCA) pertenece al control estadístico de procesos, pero de naturaleza multivariante.

Universidad de Valladolid

El control estadístico de procesos univariante tiene varias limitaciones, las variables se procesan individualmente, con una gráfica de control por cada variable, por lo que solo es posible monitorizar un número pequeño de variables. Otra limitación es que no se tiene en cuenta la correlación entre variables, algo fundamental en un proceso de una planta industrial.

El control estadístico de procesos multivariante (MSPC) se implementa para monitorizar el conjunto completo de variables del proceso y tener en cuenta la correlación entre ellos.

La naturaleza de un proceso industrial es que tiene una alta dimensionalidad con una alta colinealidad, al estar la mayoría de las variables correlacionadas, se produce mucho ruido, además de que ocurren problemas como falta de datos si fallan las herramientas de captación de datos, además de datos espurios en la planta. Todo esto motiva a utilizar el control estadístico de procesos multivariante.

El Análisis de Componentes Principales es una de las técnicas MSPC más utilizadas. PCA es una técnica estadística para reducir la dimensionalidad original del proceso mediante la proyección de los datos a un espacio de menor dimensión que el original. Este conjunto reducido de variables es lo que se llama 'componentes principales', las cuales no están correlacionadas entre sí.

En un espacio muestral de m variables, PCA nos permite encontrar un número de factores subyacentes a , que son las que se denominan componentes principales, ($a < m$) que contienen casi toda la información que las m variables originales.

PCA tiene como objetivo proyectar los puntos sobre un espacio de dimensión menor, pero manteniendo lo máximo posible sus posiciones relativas, conservando la variabilidad de los puntos como se observa en la Ilustración 7. [10]

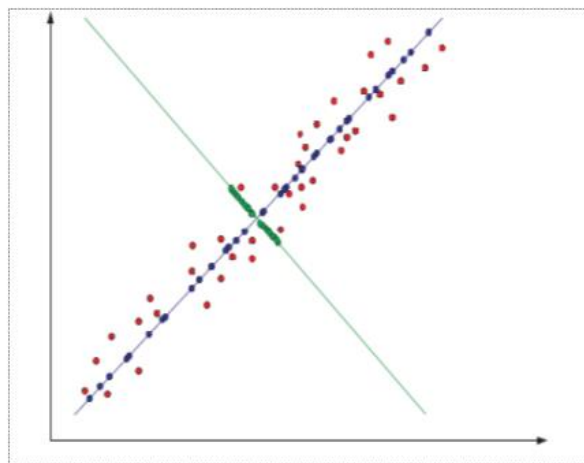


Ilustración 7 Interpretación geométrica PCA [11]



Como se observa en la Ilustración 7, una dirección nos mantiene la variabilidad de los puntos manteniendo la información, sin embargo, en la dirección perpendicular (puntos verdes) se tiene poca variabilidad y se pierde la información sobre las distancias.

Para comenzar PCA es necesario realizar un tratamiento de los datos representativos del comportamiento normal de la planta. Este tratamiento consiste en:

- Eliminar variables inapropiadas, como las que tienen errores de medida muy grande
- Normalizar los datos para que todos tengan el mismo peso, restando de cada variable su valor medio y dividiendo por su desviación estándar para capturar la variación de la media y tener varianza unidad
- Eliminar outliers

Una vez tratados los datos se comienza a realizar PCA a partir de los datos originales normalizados con media 0 y varianza 1 de la planta, en el que se tiene m variables y n observaciones de cada variable, se construye la matriz de datos X tal que $X \in R^{n \times m}$.

A partir de la matriz X se calcula la matriz de correlación de X, que la llamamos A:

$$A = \frac{1}{(n-1)} X^T X \tag{7}$$

Haciendo la descomposición en valores singulares de A, se obtienen los vectores propios, llamados vectores de carga, que se almacenan en una matriz llamada V, y sus correspondientes valores propios almacenados en la matriz Λ :

$$A = V \Lambda V^T \tag{8}$$

$V \in R^{n \times m}$ es una matriz ortogonal ($V^T V = I$) cuyas columnas son los vectores propios, que en PCA se llaman ‘loadings’ o vectores de carga. La matriz $\Lambda \in R^{n \times m}$ es una matriz que contiene los valores propios reales no negativos en orden decreciente de A de la manera $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$ y donde el valor singular i de A (o valor propio) es igual al cuadrado de i-esimo valor propio de X $\rightarrow \lambda_i = \sigma_i^2$

Con el objetivo de mantener la máxima información de los datos se retienen un numero a de vectores de carga correspondientes a los valores singulares más grandes ($a < m$) proyectando un vector de observaciones $X \in R^{n \times m}$ en este espacio de menor dimensión se obtienen los componentes principales $T \in R^{n \times a}$. Estos vectores retenidos se almacenan en una matriz $P \in R^{m \times a}$.

$$T = X P \tag{9}$$



Una vez definido este nuevo espacio T se pueden comprobar que se cumplen algunas propiedades. Definiendo t_i como la columna i-esima de T se cumple:

- $\text{Var}(t_1) \geq \text{Var}(t_2) \geq \dots \geq \text{Var}(t_a)$
- $\text{Media}(t_i) = 0$
- $t_i^T t_j = 0; i \neq j$
- No existe ninguna otra expansión ortogonal de a componentes que capture más variación de los datos

Acabada la proyección ortogonal para reducir el espacio se puede recalcular los datos originales en función de T.

$$\hat{X} = T P^T \quad (10)$$

Después se puede definir la matriz de residuos E calculada como la diferencia entre el espacio original y el espacio recalculado.

$$E = X - \hat{X} \quad (11)$$

El espacio de los residuos, E, captura la variación de los datos de observación contenidos en los vectores propios (carga) asociados con los m-a valores singulares más pequeños.

$$X = T P^T + E \quad (12)$$

Para reducir el orden del sistema de una manera correcta hay que hacer una buena elección de a, para ello hay varios criterios distintos de elección de a.

- **Test del porcentaje de la varianza:** Se selecciona a de forma que explique un porcentaje específico de la varianza total
- **Test del codo:** Se representan los valores de λ_i frente a i, y se busca un valor a partir del que λ sea igual y tenga un valor pequeño

Una vez obtenido el subespacio con los componentes principales se pasa a la fase de detección de fallos mediante PCA para ello se definen dos estadísticos que van a permitir definir un umbral para detectar anomalías en el proceso. El estadístico T^2 o de Hotelling monitoriza el espacio de los componentes principales mientras que el estadístico Q o SPE monitoriza el espacio de residuos.



2.4.1 ESTADÍSTICOS PARA LA DETECCIÓN DE FALLOS

Se van a utilizar dos estadísticos para la detección de fallos mediante PCA una vez se ha definido el espacio reducido de componentes principales.

- Estadístico T^2 o de Hotelling

El estadístico T^2 monitoriza el espacio de dimensión a , de las componentes principales. Para calcular este estadístico se van a utilizar las matrices obtenidas en el método de PCA.

$$T^2 = x P \Lambda_a^{-1} P^T x^T \quad (13)$$

Para una observación x , de m variables, T^2 se interpreta como la distancia de la observación al centro del modelo.

Una vez se calcule este estadístico para las n observaciones se compara con el umbral definido por las observaciones del comportamiento normal de la planta.

$$T_{\alpha}^2 = \frac{(n^2 - 1) * a}{n * (n - a)} F_{\alpha}(a, n - a) \quad (14)$$

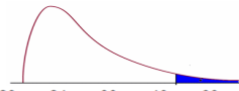
Donde n es el número de observaciones de comportamiento normal de la planta, a es el número de componentes principales y F_{α} es la distribución Fisher-Snedecor con grados de libertad a y $n - a$. [11]

La distribución Fisher-Snedecor es una distribución de probabilidad continua utilizada especialmente en el análisis de varianza. Para utilizarse se utilizan unas tablas definidas para la distribución F para las cuales hay que saber los grados de libertad y definir α , que es el grado de fiabilidad, que se va a utilizar cuyos valores más comunes son 0.01 y 0.05. Un ejemplo de dicha distribución se puede observar en las Ilustraciones 8 y 9.



Distribución F 0.01

En las columnas se encuentran los valores F que corresponden al área 0.05 a la derecha
En las columnas se encuentran los grados de libertad del numerador
En los renglones se encuentran los grados de libertad del denominador.

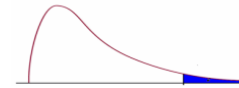


	1	2	3	4	5	6	7	8	9	10	11	12	15	20	24	30	40	60	120
1	4052	4999	5403	5625	5764	5859	5928	5981	6022	6056	6083	6106	6157	6209	6235	6261	6287	6313	6339
2	98.50	99.00	99.17	99.25	99.30	99.33	99.36	99.37	99.39	99.40	99.41	99.42	99.43	99.45	99.46	99.47	99.47	99.48	99.49
3	34.12	30.82	29.46	28.71	28.24	27.91	27.67	27.49	27.35	27.23	27.13	27.05	26.87	26.69	26.60	26.50	26.41	26.32	26.22
4	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.66	14.55	14.45	14.37	14.20	14.02	13.93	13.84	13.75	13.65	13.56
5	16.26	13.27	12.06	11.39	10.97	10.67	10.46	10.29	10.16	10.05	9.96	9.89	9.72	9.55	9.47	9.38	9.29	9.20	9.11
6	13.75	10.92	9.78	9.15	8.75	8.47	8.26	8.10	7.98	7.87	7.79	7.72	7.56	7.40	7.31	7.23	7.14	7.06	6.97
7	12.25	9.55	8.45	7.85	7.46	7.19	6.99	6.84	6.72	6.62	6.54	6.47	6.31	6.16	6.07	5.99	5.91	5.82	5.74
8	11.26	8.65	7.59	7.01	6.63	6.37	6.18	6.03	5.91	5.81	5.73	5.67	5.52	5.36	5.28	5.20	5.12	5.03	4.95
9	10.56	8.02	6.99	6.42	6.06	5.80	5.61	5.47	5.35	5.26	5.18	5.11	4.96	4.81	4.73	4.65	4.57	4.48	4.40
10	10.04	7.56	6.55	5.99	5.64	5.39	5.20	5.06	4.94	4.85	4.77	4.71	4.56	4.41	4.33	4.25	4.17	4.08	4.00
11	9.65	7.21	6.22	5.67	5.32	5.07	4.89	4.74	4.63	4.54	4.46	4.40	4.25	4.10	4.02	3.94	3.86	3.78	3.69
12	9.33	6.93	5.95	5.41	5.06	4.82	4.64	4.50	4.39	4.30	4.22	4.16	4.01	3.86	3.78	3.70	3.62	3.54	3.45
13	9.07	6.70	5.74	5.21	4.86	4.62	4.44	4.30	4.19	4.10	4.02	3.96	3.82	3.66	3.59	3.51	3.43	3.34	3.25
14	8.86	6.51	5.56	5.04	4.69	4.46	4.28	4.14	4.03	3.94	3.86	3.80	3.66	3.51	3.43	3.35	3.27	3.18	3.09
15	8.68	6.36	5.42	4.89	4.56	4.32	4.14	4.00	3.89	3.80	3.73	3.67	3.52	3.37	3.29	3.21	3.13	3.05	2.96
16	8.53	6.23	5.29	4.77	4.44	4.20	4.03	3.89	3.78	3.69	3.62	3.55	3.41	3.26	3.18	3.10	3.02	2.93	2.84
17	8.40	6.11	5.18	4.67	4.34	4.10	3.93	3.79	3.68	3.59	3.52	3.46	3.31	3.16	3.08	3.00	2.92	2.83	2.75
18	8.29	6.01	5.09	4.58	4.25	4.01	3.84	3.71	3.60	3.51	3.43	3.37	3.23	3.08	3.00	2.92	2.84	2.75	2.66
19	8.18	5.93	5.01	4.50	4.17	3.94	3.77	3.63	3.52	3.43	3.36	3.30	3.15	3.00	2.92	2.84	2.76	2.67	2.58
20	8.10	5.85	4.94	4.43	4.10	3.87	3.70	3.56	3.46	3.37	3.29	3.23	3.09	2.94	2.86	2.78	2.69	2.61	2.52
21	8.02	5.78	4.87	4.37	4.04	3.81	3.64	3.51	3.40	3.31	3.24	3.17	3.03	2.88	2.80	2.72	2.64	2.55	2.46
22	7.95	5.72	4.82	4.31	3.99	3.76	3.59	3.45	3.35	3.26	3.18	3.12	2.98	2.83	2.75	2.67	2.58	2.50	2.40
23	7.88	5.66	4.76	4.26	3.94	3.71	3.54	3.41	3.30	3.21	3.14	3.07	2.93	2.78	2.70	2.62	2.54	2.45	2.35
24	7.82	5.61	4.72	4.22	3.90	3.67	3.50	3.36	3.26	3.17	3.09	3.03	2.89	2.74	2.66	2.58	2.49	2.40	2.31
25	7.77	5.57	4.68	4.18	3.85	3.63	3.46	3.32	3.22	3.13	3.06	2.99	2.85	2.70	2.62	2.54	2.45	2.36	2.27
26	7.72	5.53	4.64	4.14	3.82	3.59	3.42	3.29	3.18	3.09	3.02	2.96	2.81	2.66	2.58	2.50	2.42	2.33	2.23
27	7.68	5.49	4.60	4.11	3.78	3.56	3.39	3.26	3.15	3.06	2.99	2.93	2.78	2.63	2.55	2.47	2.38	2.29	2.20
28	7.64	5.45	4.57	4.07	3.75	3.53	3.36	3.23	3.12	3.03	2.96	2.90	2.75	2.60	2.52	2.44	2.35	2.26	2.17
29	7.60	5.42	4.54	4.04	3.73	3.50	3.33	3.20	3.09	3.00	2.93	2.87	2.73	2.57	2.49	2.41	2.33	2.23	2.14
30	7.56	5.39	4.51	4.02	3.70	3.47	3.30	3.17	3.07	2.98	2.91	2.84	2.70	2.55	2.47	2.39	2.30	2.21	2.11
40	7.31	5.18	4.31	3.83	3.51	3.29	3.12	2.99	2.89	2.80	2.73	2.66	2.52	2.37	2.29	2.20	2.11	2.02	1.92
60	7.08	4.98	4.13	3.65	3.34	3.12	2.95	2.82	2.72	2.63	2.56	2.50	2.35	2.20	2.12	2.03	1.94	1.84	1.73
120	6.85	4.79	3.95	3.48	3.17	2.96	2.79	2.66	2.56	2.47	2.40	2.34	2.19	2.03	1.95	1.86	1.76	1.66	1.53

Ilustración 8 Tabla de distribución F con $\alpha = 0.01$ [12]

Distribución F 0.05

En las columnas se encuentran los valores F que corresponden al área 0.05 a la derecha
En las columnas se encuentran los grados de libertad del numerador
En los renglones se encuentran los grados de libertad del denominador.



	1	2	3	4	5	6	7	8	9	10	11	12	15	20	24	30	40	60	120
1	161.4	199.5	215.7	224.6	230.2	234.0	236.8	238.9	240.5	241.9	243.0	243.9	245.9	248.0	249.1	250.1	251.1	252.2	253.3
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.41	19.41	19.43	19.45	19.45	19.46	19.47	19.48	19.49
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79	8.76	8.74	8.70	8.66	8.64	8.62	8.59	8.57	8.55
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.94	5.91	5.86	5.80	5.77	5.75	5.72	5.69	5.66
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74	4.70	4.68	4.62	4.56	4.53	4.50	4.46	4.43	4.40
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	4.03	4.00	3.94	3.87	3.84	3.81	3.77	3.74	3.70
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64	3.60	3.57	3.51	3.44	3.41	3.38	3.34	3.30	3.27
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35	3.31	3.28	3.22	3.15	3.12	3.08	3.04	3.01	2.97
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14	3.10	3.07	3.01	2.94	2.90	2.86	2.83	2.79	2.75
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98	2.94	2.91	2.85	2.77	2.74	2.70	2.66	2.62	2.58
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85	2.82	2.79	2.72	2.65	2.61	2.57	2.53	2.49	2.45
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75	2.72	2.69	2.62	2.54	2.51	2.47	2.43	2.38	2.34
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67	2.63	2.60	2.53	2.46	2.42	2.38	2.34	2.30	2.25
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.57	2.53	2.46	2.39	2.35	2.31	2.27	2.22	2.18
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.51	2.48	2.40	2.33	2.29	2.25	2.20	2.16	2.11
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.46	2.42	2.35	2.28	2.24	2.19	2.15	2.11	2.06
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.41	2.38	2.31	2.23	2.19	2.15	2.10	2.06	2.01
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.37	2.34	2.27	2.19	2.15	2.11	2.06	2.02	1.97
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.34	2.31	2.23	2.16	2.11	2.07	2.03	1.98	1.93
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.31	2.28	2.20	2.12	2.08	2.04	1.99	1.95	1.90
21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.28	2.25	2.18	2.10	2.05	2.01	1.96	1.92	1.87
22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30	2.26	2.23	2.15	2.07	2.03	1.98	1.94	1.89	1.84
23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27	2.24	2.20	2.13	2.05	2.01	1.96	1.91	1.86	1.81
24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25	2.22	2.18	2.11	2.03	1.98	1.94	1.89	1.84	1.79
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24	2.20	2.16	2.09	2.01	1.96	1.92	1.87	1.82	1.77
26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.18	2.15	2.07	1.99	1.95	1.90	1.85	1.80	1.75
27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20	2.17	2.13	2.06	1.97	1.93	1.88	1.84	1.79	1.73
28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19	2.15	2.12	2.04	1.96					



Una vez definido el umbral, ya se puede realizar la detección de fallos. Si el valor de $T_i^2 > T_\alpha$ se produce un fallo.

- Estadístico Q o SPE

El estadístico Q monitoriza el espacio de dimensión $m - a$, enfocando en los valores singulares más pequeños.

Este estadístico se utiliza para mejorar la detección de fallos ya que se tiene en cuenta el espacio de residuos y utilizando el error de predicción para detectar los fallos.

Para calcular este estadístico también se van a utilizar las matrices obtenidas en el modelo de PCA.

$$Q = r^T r \quad (15)$$

Con r siendo el vector de residuos.

$$r = (I - PP^T) x^T \quad (16)$$

Cuando el sistema está bajo control Q es muy pequeña, ya que son variaciones debido al ruido. Para la detección de fallos se define un umbral a partir de las observaciones del comportamiento normal del proceso.

$$Q_\alpha = \frac{\sigma_Q^2}{2 * \mu_Q} Chi \left(p, 2 \frac{\mu_Q^2}{\sigma_Q^2} \right) \quad (17)$$

Como se observa en la ecuación 17, para calcular el umbral se tiene en cuenta la media μ_Q del estadístico Q y la desviación estándar σ_Q del estadístico Q. También el umbral se calcula utilizando la distribución Chi-cuadrado, en el que p es la probabilidad que se va a utilizar del 99% y $2 \frac{\mu_Q^2}{\sigma_Q^2}$ como grados de libertad de la distribución.

La detección de fallos se hace mucho más robusta utilizando ambos estadísticos, ya que el estadístico T^2 detecta cambios alejados del origen del sistema mientras que el estadístico Q captura el ruido. [11]

Se definirá fallo si las observaciones de los estadísticos sobrepasan los umbrales n veces consecutivas, en nuestro proyecto será de 10 veces consecutivas. Si se elige un numero demasiado pequeño de veces consecutivas que se supera el umbral se puede incurrir en posibles falsas alarmas y si se elige un numero demasiado grande se pueden pasar por alto los fallos.



3. DETECCIÓN DE FALLOS CON TÉCNICAS DE MACHINE LEARNING

El 'Machine Learning' o aprendizaje automático es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. La máquina que realmente aprende es un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros. Estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana. [13]

Hay dos tipos de aprendizaje automático, el aprendizaje supervisado y el no supervisado.

El aprendizaje supervisado son algoritmos que aprenden de los datos introducidos por lo que se necesita la intervención humana para etiquetar, clasificar e introducir los datos en el algoritmo. A partir de estos datos de entrada el algoritmo genera datos de salida esperados ya que los datos introducidos están etiquetados y clasificados. Hay dos tipos de aprendizaje supervisado:

- **Clasificación**

Clasifican un objeto dentro de diversas clases. Hay dos tipos de clasificación como son la clasificación binaria o la clasificación multi-clase.

- **Regresión**

Predicen una salida a partir de un rango continuo de valores posibles. Hay distintos tipos de regresión como son la lineal, la logística o la polinomial.

El aprendizaje no supervisado son algoritmos que aprenden de datos con elementos no etiquetados buscando patrones entre ellos ya que se introducen datos de entrada sin etiquetar por lo que no necesita la intervención humana. Hay dos tipos de aprendizaje no supervisado:

- **Clustering**

Clasifica en grupos los datos de salida. Hay distintos tipos de algoritmos e clustering como son K-Means, DBScan, Mean Shift, etc.

- **Asociación**

Descubre reglas del conjunto de datos.



Algunos ejemplos de utilización del ‘Machine Learning’ es la detección de fraudes en transacciones, predicción de tráfico, prediagnósticos médicos basados en síntomas del paciente y como en este proyecto, la detección de fallos en un proceso industrial. [14]

Dentro del ‘Machine Learning’ aparece el aprendizaje profundo o ‘Deep Learning’ que ha mejorado la capacidad de clasificar, reconocer, detectar y describir del aprendizaje automático. El ‘Deep Learning’ es el conjunto de algoritmos que permiten a las máquinas aprender por cuenta propia reconociendo patrones mediante el uso de muchas capas de procesamiento. El ‘Deep Learning’ es la nueva evolución de ‘Machine Learning’ aunque se centra sobre todo en redes neuronales, sin embargo, el ‘Machine Learning’ abarca muchas más técnicas. [8]

3.1 REDES NEURONALES ARTIFICIALES (ANN)

Las redes neuronales artificiales son uno de los exponentes de la inteligencia artificial (IA), bajo el paraguas del aprendizaje automático o ‘machine learning’, que hace posible que las máquinas puedan ejecutar tareas o funciones que no éramos capaces de automatizar con los sistemas convencionales.

Las redes neuronales artificiales son un modelo computacional que permite simular el comportamiento del cerebro humano, es decir, dotar a las máquinas de la capacidad de aprender de una manera similar a como lo hace nuestro cerebro. Una red neuronal artificial está formada por neuronas artificiales, que son unidades o nodos que reciben información del exterior o de otras neuronas, de manera similar a los impulsos nerviosos que reciben las neuronas del cerebro humano, las procesan y generan un valor de salida que alimenta a otras neuronas de la red o son la salida hacia el exterior de la red.

Una neurona artificial está formada por:

- Un conjunto de entradas que son las interconexiones por donde reciben información del exterior de la red o de otras neuronas. Cada una de las entradas está ponderada por un peso que determina la importancia de la información que recibe por esa interconexión y que es un valor que se va ajustando durante el entrenamiento de la red neuronal para minimizar el error de la red y que los resultados sean más fiables.
- Un conjunto de funciones como son la de propagación, la de activación y la de transferencia.

La función de propagación relaciona matemáticamente las entradas, sus pesos y un sesgo (bias) para calcular el valor de salida de la neurona.

La función de activación determina si la neurona esta activa o no, es decir, si produce un valor de salida o no.

La función de transferencia modula el valor calculado de salida de la neurona.

- La salida de la neurona que es la interconexión por donde entrega el resultado al exterior de la red neuronal o a otras neuronas. [15]

Una red neuronal consiste en una serie de capas de neuronas.

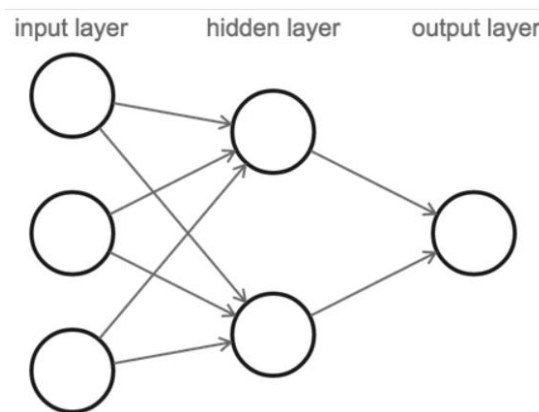


Ilustración 10 Red neuronal de 2 capas [16]

En la Ilustración 10 se ve una red neuronal con 3 neuronas de entrada, 2 neuronas en la capa oculta y 1 neurona de salida.

Para realizar los cálculos en la red neuronal hay que tener en cuenta que hay dos parámetros fundamentales en las redes neuronales como son el peso (w) y el bias (b). El peso representa la influencia relativa de la entrada por lo que un peso alto modifica de manera significativa la salida. El sesgo o bias representa la distancia que esta el resultado real del deseado por lo que un bias bajo significa que el valor de salida es similar al deseado. Ambos parámetros se modifican en la red durante el entrenamiento.

Las redes neuronales llevan asociadas una función de activación que se aplica para obtener la salida ya sea de la red o la salida que se transmite a la siguiente capa. La función de activación utilizada depende de la ecuación a partir de la entrada o entradas anteriores en la que se tiene en cuenta el peso y el bias.

$$z = F (b + \sum_i w_i x_i) \tag{18}$$

Hay muchas funciones de activación, pero las más utilizadas son la sigmoide y la ReLu.

En la Ilustración 11 se va a observar cómo trabaja una neurona artificial.

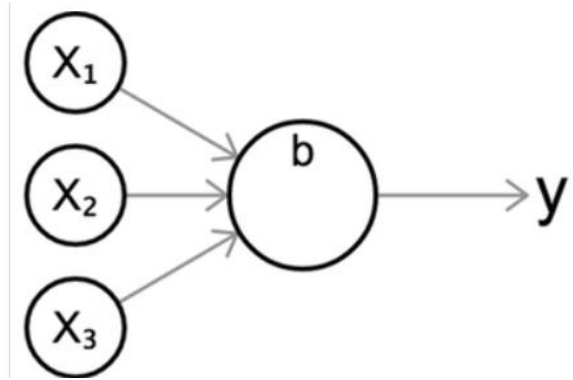


Ilustración 11 Neurona artificial [16]

En la Ilustración 11 se observa como una neurona con tres entradas, x_1 , x_2 y x_3 , genera un único valor y de salida. Primero se calcula la función a partir de las entradas y después aplicamos la función de activación para calcular la salida. [16]

$$z = b + w_1x_1 + w_2x_2 + w_3x_3 \quad (19)$$

$$y = \sigma(z) \quad (20)$$

En la Ilustración 12 se ve más en detalle el diagrama de una neurona artificial.

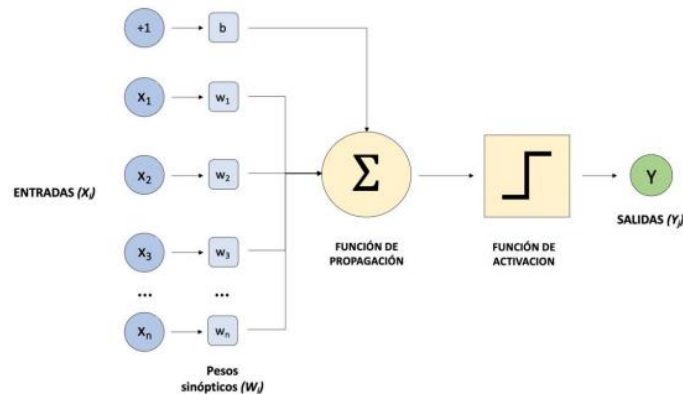


Ilustración 12 Diagrama de una neurona artificial [8]

Este tipo de redes neuronales en el que el proceso por el que la red neuronal envía su entrada a través de sus capas hacia la salida se les llama redes neuronales de propagación hacia adelante.

3.1.1 REDES NEURONALES DE PROPAGACION HACIA DELANTE

La propagación hacia adelante es el conjunto de procesos matemáticos desde que introducimos nuestros datos en la red neuronal hasta que la red neuronal

da un resultado. Las operaciones se realizan de izquierda a derecha, desde la capa de entrada hasta la capa de salida que proporciona el resultado.

El primer modelo de red neuronal artificial que se estudio fue el perceptrón alrededor de los años 60 y que está formado por varias neuronas para recibir las entradas a la red y una neurona de salida que decide cuando una entrada a la red pertenece a una de las dos clases que se puede reconocer.

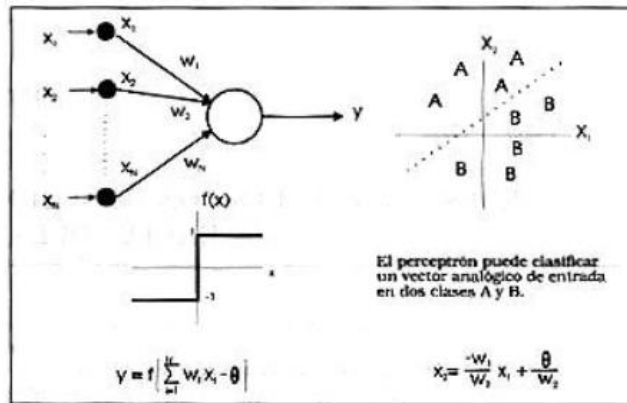


Ilustración 13 Perceptrón [17]

A pesar del gran potencial en su momento, pronto se vio que tenía una capacidad de representación bastante limitada ya que solo consta de una capa de entrada y otra de salida con una única neurona por lo que solo tenía capacidad de reconocer patrones muy sencillos. [17]

Después del perceptrón se desarrollan las redes neuronales de propagación hacia adelante multicapa que están compuestas por una capa de entrada, una capa de salida y una o más capas ocultas.

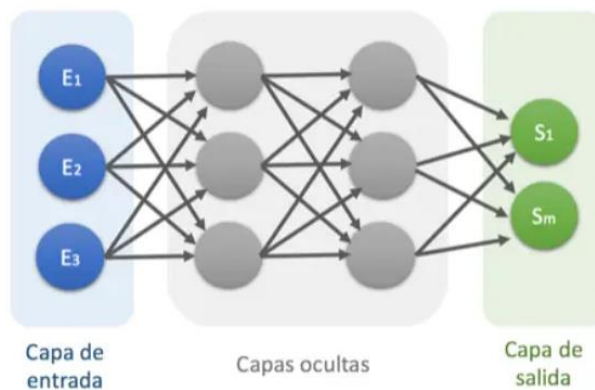


Ilustración 14 Red neuronal feedforward multicapa [18]

En la Ilustración 14 se observa el diagrama de una red neuronal multicapa con 2 capas ocultas en las que la capa de entrada conecta la red neuronal con el exterior en el que cada neurona corresponde con cada una de las variables de entrada de la red, las capas ocultas son una aglomeración de capas en las que



cada activación de una salida procede de la suma ponderada de las activaciones de la capa anterior más sus bias y la capa de salida conecta las capas ocultas con la salida de la red que proporciona el resultado. [18]

Entre las ventajas de las redes multicapa, destaca que pueden usarse para aproximar cualquier función por lo que pueden usarse para la creación de modelos de análisis de regresión. Como la clasificación es un tipo de regresión en la que las salidas son categóricas, las redes de propagación hacia delante multicapa son buenos algoritmos de clasificación.

Una de las principales características de las redes neuronales artificiales es su capacidad de aprendizaje. Las Redes Neuronales deben pasar por un procedimiento de aprendizaje o entrenamiento, en el cual se pretende modelar la red para que produzca una salida deseada a partir de unos datos de entrada.

Toda red multicapa se encuentra definida en términos de su arquitectura, sus funciones de activación, las bias y los pesos, siendo estas dos últimas las variables de ajuste en el momento de utilizar un algoritmo de entrenamiento para que la red aprenda.

En el entrenamiento, además de ajustar los pesos y las bias, es preciso optimizar el número de neuronas que la conforman porque de esto depende la velocidad que adquiere la red para aprender.

El entrenamiento de una red multicapa es de dos tipos: incremental si el aprendizaje se da al presentarle a la red cada entrada (variables explicativas) con su respectiva salida (variables explicadas), o batch, cuando el aprendizaje de la red ocurre sólo después de haberle presentado todo el conjunto de datos de entrada y salida. El objetivo del entrenamiento es minimizar una función de costo, la que se usa con frecuencia es el error cuadrático medio. [19]

3.1.2 REDES NEURONALES RECURRENTE

Las redes neuronales recurrentes (RNN) son una clase de aprendizaje profundo basada en los trabajos de David Rumelhart en 1986. Las RNN son conocidas por su capacidad para procesar y obtener información de datos secuenciales.

Las RNN generalmente aumentan la arquitectura de red multicapa convencional con la adición de ciclos que conectan nodos adyacentes o pasos de tiempo. Estos ciclos constituyen la memoria interna de la red que se utiliza para evaluar las propiedades del dato actual con respecto a los datos del pasado inmediato. [20]

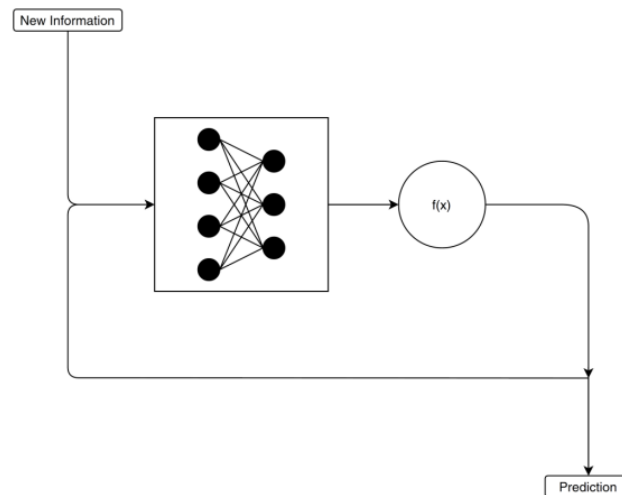


Ilustración 15 Diagrama cíclico de una red neuronal recurrente [20]

Ahora cada neurona recurrente tiene dos conjuntos de parámetros, uno que lo aplica a la entrada de datos que recibe de la capa anterior y otro conjunto que lo aplica a la entrada de datos correspondiente al vector salida del instante anterior y se puede expresar de la siguiente manera.

$$Y_t = f(Wx_t + Uy_{t-1} + b) \quad (21)$$

Donde $x = (x_1, \dots, x_T)$ representa la secuencia de entrada proveniente de la capa anterior, W los pesos de la matriz y b el bias vistos ya en las anteriores capas. Las RNN extienden esta función con una conexión recurrente en el tiempo donde U es la matriz de pesos que opera sobre el estado de la red en el instante de tiempo anterior (y_{t-1}) anterior. La parte de una red neuronal que preserva un estado a través del tiempo se suele llamar 'memory cell'.

Y precisamente esta 'memoria interna' es lo que hace de este tipo de redes muy adecuadas para problemas de aprendizaje automático que involucran datos secuenciales. Gracias a su memoria interna, las RNN pueden recordar información relevante sobre la entrada que recibieron, lo que les permite ser más precisas en la predicción de lo que vendrá después manteniendo información de contexto a diferencia de las redes neuronales de propagación hacia delante, que no pueden recordar acerca de lo que ha sucedido en el pasado, excepto lo reflejado en su entrenamiento a través de sus pesos. [21]

En los últimos años ha habido multitud de avances en la aplicación de RNN a la resolución de problemas, tales como reconocimiento de voz, tratamiento de imágenes, modelado de lenguaje, traducción de textos, etc.

3.1.3 REDES NEURONALES GRU

Las redes neuronales recurrentes más utilizadas en la actualidad son LSTM y GRU, pero en este proyecto se utilizará la red neuronal GRU. El modelo final de la red GRU es más simple que el modelo de la red LSTM, por lo que es mejor variante además de que resuelve el problema de dependencia a largo plazo de RNN.

El modelo GRU cuenta con dos puertas: actualizar puerta y restablecer puerta.

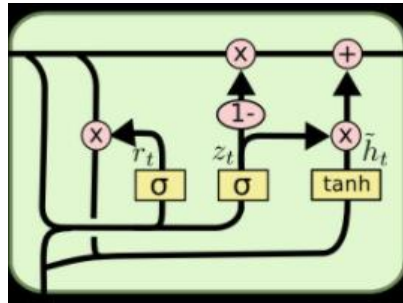


Ilustración 16 Diagrama de red GRU [22]

En la imagen se observa el diagrama de una red neuronal recurrente GRU en el que z_t representa la puerta de actualización y r_t representa la puerta de reinicio.

La puerta de actualización se usa para controlar el grado en que la información de estado en el momento anterior se lleva al estado actual. Un valor mayor de la puerta de actualización indica que se ingresa más información de estado en el momento anterior. La puerta de reinicio controla cuánta información en el estado anterior se escribe en el conjunto de candidatos actual \tilde{h}_t . Cuanto más pequeña es la puerta de reinicio, menos información se escribe en el estado anterior. Este tipo de red conserva características importantes a través de las funciones de puerta garantizando que no se perderán durante la propagación a largo plazo. [22]

La activación h_t^j de la red GRU en el tiempo t es una interpolación lineal entre la activación previa h_{t-1}^j y la activación candidata h_t^j .

$$h_t^j = (1 - z_t^j) h_{t-1}^j + z_t^j h_t^j \quad (22)$$

Donde la puerta de actualización z_t^j decide cuánto actualiza la unidad su activación. La puerta de actualización esta computada por:

$$z_t^j = \sigma (W_z x_t + U_z h_{t-1}^j) \quad (23)$$

Este procedimiento toma la suma lineal entre el estado existente y el nuevo estado calculado.

El candidato de activación h_t^j es calculado de forma similar que la unidad recurrente tradicional.

$$h_t^j = \tanh (Wx_t + U (r_t \odot h_{t-1}))^j \quad (24)$$

Donde r_t es un conjunto de puertas de reinicio y \odot es una multiplicación por elementos. Cuando la puerta de reinicio esta desactivada hace que la unidad actúe como si estuviera leyendo el primer símbolo de la secuencia de entrada, permitiendo olvidar el estado calculado previamente.

La puerta de reinicio r_t^j es calculada de forma similar que la puerta de actualización. [23]

$$r_t^j = \sigma (W_r x_t + U_r h_{t-1})^j \quad (25)$$

3.1.4 AUTOENCODERS

Un Autoencoder es un algoritmo de aprendizaje no supervisado siendo un tipo específico de red neuronal de propagación hacia adelante donde la entrada es la misma que la salida y consta de tres partes, el codificador, la capa oculta y el decodificador. La codificación es el proceso de proyectar las variables de entrada a la capa oculta, mientras que la decodificación es el proceso de proyectar la capa oculta a la salida.

El Autoencoder primero comprime la entrada en una representación latente a través de un codificador y luego reconstruye la salida utilizando esta representación. El modelo puede ser entrenado minimizando los errores de reconstrucción de las variables de entrada y salida. [24]

La aplicación más tradicional de los autoencoders es la reducción de dimensionalidad y el aprendizaje de características de conjuntos de datos para el reconocimiento de imágenes, pero en la última década se han estado utilizando para desarrollar potentes redes neuronales de múltiples capas ocultas capaces de procesar gran cantidad de datos. Esto se conoce como aprendizaje profundo o 'deep learning'. En este proyecto se va a utilizar una variante de los Autoencoder que es el denominado 'Stacked Autoencoder' (SAe).

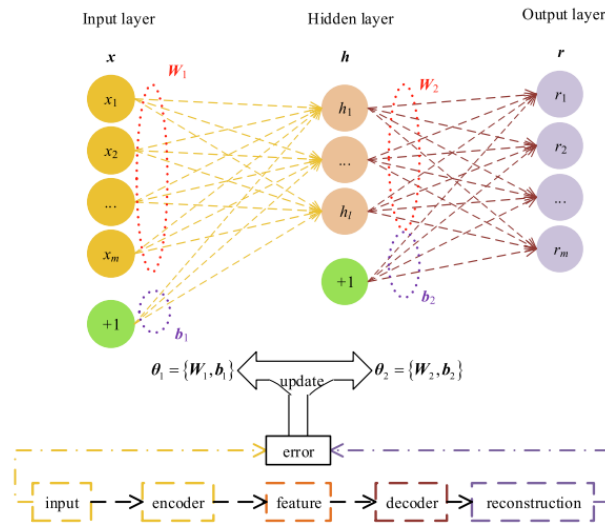


Fig. 1. Structure of AE.

Ilustración 17 Estructura de un Autoencoder [25]

El SAe es una cascada de múltiples Autoencoders utilizado para completar la extracción de características capa por capa. Las características resultantes son más representativas que las obtenidas en el Autoencoder. En el SAe las características extraídas por el Autoencoder anterior se consideran como la entrada del Autoencoder siguiente formando la cascada de Autoencoders.

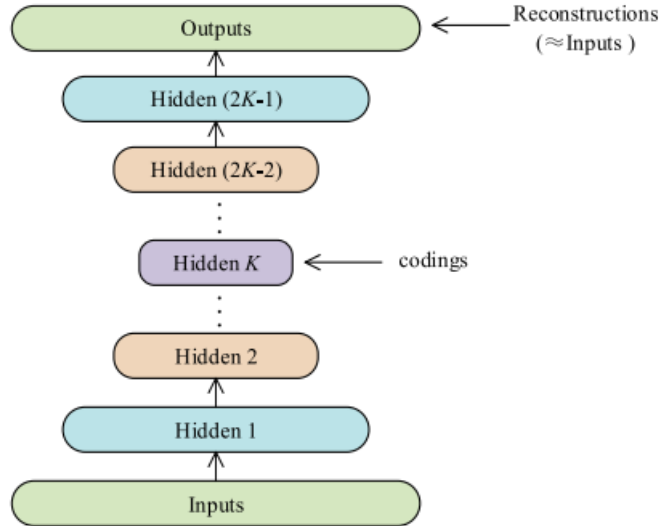


Ilustración 18 Estructura de SAe con K capas [25]

La Ilustración 18 muestra la estructura de SAe en el que la salida de la capa anterior es la entrada de la capa siguiente, y así sucesivamente. Este tipo de Autoencoders es simétrico respecto a la capa oculta.

Para entrenar el SAe se hace realizándolo capa a capa como se observa en la Ilustración 19.

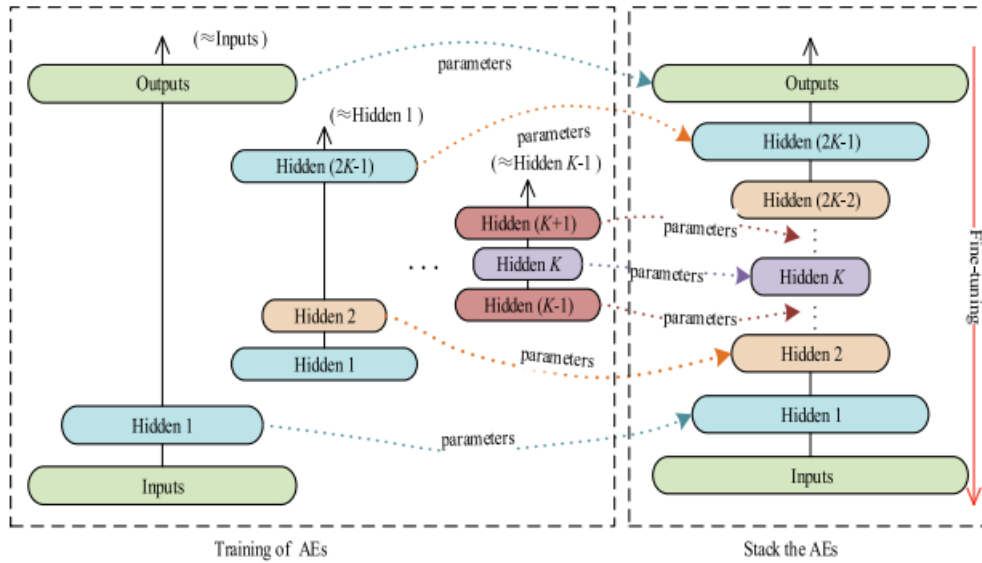


Ilustración 19 Entrenamiento y unión de SAE [25]

Mediante esta forma de entrenarlo, a partir de las entradas y como se busca que las salidas tengan la mayor similitud a las entradas se obtiene la primera capa, como esta primera capa tiene que ser igual a la capa justamente anterior a las salidas se obtiene la segunda capa, y así sucesivamente durante el proceso dependiendo de las capas ocultas que se vayan a utilizar.

Para la detección de fallos se definen los mismos dos estadísticos (T^2 y Q) que para la detección de fallos mediante PCA. Se definen los estadísticos mediante las siguientes expresiones.

$$T^2 = (h - \bar{h}_{nor})^T \Sigma_{h,nor}^{-1} (h - \bar{h}_{nor}) \quad (26)$$

En el que \bar{h}_{nor} y $\Sigma_{h,nor}^{-1}$ son la media y la matriz de covarianza de las características locales extraídas de la capa oculta, respectivamente.

$$SPE = (res - \bar{res}_{nor})^T \Sigma_{res,nor}^{-1} (res - \bar{res}_{nor}) \quad (27)$$

$$res = x - \hat{x} \quad (28)$$

En el que \bar{res}_{nor} y $\Sigma_{res,nor}^{-1}$ son la media y la matriz de covarianza de las características locales extraídas de la predicción (Diferencia entre los datos de la capa de salida y los datos de entrada), respectivamente. [25]

Los umbrales de cada estadístico se definen mediante el percentil 99, haciendo que el 99% de los datos de cada estadístico estén por debajo del umbral.



3.1.4.1 AUTOENCODERS VARIACIONALES

Los Autoencoders variacionales (VAE) son modelos de aprendizaje que mezclan las redes neuronales con distribuciones de probabilidad. Su principal uso es el de construir modelos generativos que son capaces de producir datos sintéticos que siguen los mismos patrones que los grandes conjuntos de datos de los que se alimentan.

VAE se ha desarrollado recientemente y crea un modelo generativo profundo, que es un potente método para aprender la representación a partir de los datos de forma no lineal. Aprovecha la información de la densidad de los datos para encontrar un espacio de características eficiente y de menor dimensión como la distribución normal multivariable.

Este tipo de Autoencoders se ha desarrollado para solucionar problemas que ocurren con los Autoencoders sobre todo en el espacio latente, que es la salida del encoder que sirve de entrada al decoder, es decir, la capa oculta.

El principal problema que se ha constatado es que, en la mayoría de los casos, las representaciones internas que se obtienen en el espacio latente forman un espacio que no es continuo, sino formado por diversas bolsas aisladas que agrupan en su interior representaciones de datos de entrada similares. [26] Mediante el VAE podemos resolver este problema.

Debido a que los métodos existentes de supervisión basados en variables latentes utilizan T^2 en el espacio reducido, se basan en la suposición de que las observaciones incontroladas siguen una distribución normal. La VAE utiliza la inferencia variacional para reducir los datos de alta dimensión a datos de baja dimensión que siguen una distribución normal multivariante. Por lo tanto, el uso de VAE para la supervisión de procesos puede abordar la limitación que supone la suposición distributiva que subyace a la T^2 de muchos de los métodos de supervisión basados en variables latentes.

En la Ilustración 20 se observa como el espacio latente es continuo, permitiendo generar objetos por medio de la interpolación de representaciones latentes de datos de entrada.

Este espacio latente está formado por dos vectores de mismo tamaño, un vector de medias $\vec{\mu}$ y un vector de desviaciones estándar $\vec{\sigma}$ que conjuntamente forman un vector de variables aleatorias descritas por medio de distribuciones normales. [27]

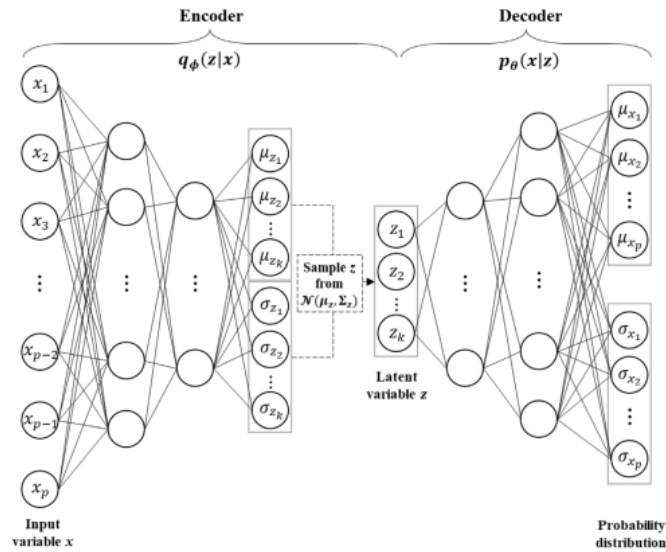


Ilustración 20 Arquitectura Autoencoder Variacional [27]

A partir de este vector de distribuciones normales podemos obtener un vector (que realmente es una muestra al azar siguiendo esas funciones de distribución) que podemos pasar al decoder para que genere una salida.

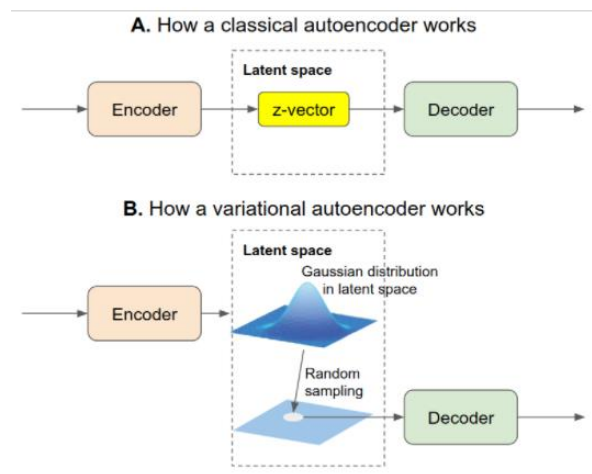


Ilustración 21 Diferencias de funcionamiento entre Autoencoder tradicional y Autoencoder variacional [26]

La diferencia del modelo VAE con otros autoencoders es que podemos interpretar el espacio latente no como un espacio de codificación, sino como una distribución de probabilidad (porque no proporciona una codificación única, sino un conjunto de codificaciones que, con mayor o menor probabilidad, podrían ser el resultado del encoder), y esta interpretación también se traslada al espacio de salida por medio del decoder. En consecuencia, las funciones de pérdida habituales (las pérdidas de representación), que suelen venir dadas por distancias entre las entradas y las salidas, no son adecuadas para medir el error que comete la red en su generación.



Para solucionar este problema se introduce un factor nuevo en la función de pérdida llamada divergencia KL que en vez de medir la distancia entre puntos mide la diferencia entre dos distribuciones de probabilidad.

Para calcular esta divergencia KL se tiene en cuenta la distribución ideal centrada en el origen y de desviación 1 para compararla con la distribución normal del espacio latente de nuestra red.

$$KL = \sum \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1 \quad (29)$$

Pero si solo consideramos este término y no imponemos condiciones adicionales, la red aprenderá a situar todas las representaciones alrededor del origen (aproximándose a la distribución objetivo) y será incapaz de diferenciar las diversas clases que podemos encontrar dentro de nuestro conjunto de entrenamiento (todas las representaciones latentes estarían demasiado cerca entre sí como para poder agruparlas en bolsas diferenciadas). Así pues, hemos de considerar también durante el entrenamiento una información adicional que el decodificador debería aprender: debe dar como salida el mismo dato que recibe como entrada, y para eso podemos usar la función de pérdida habitual.

$$\text{Loss}'(x) = \text{Loss}(x, x') + KL \quad (30)$$

Loss es la función de pérdida normal como puede ser el error cuadrático medio, x es el dato de entrada y x' es la salida producida por el decoder, lo que suele ser la función de pérdida de un Autoencoder normal, para el VAE se le suma a esta función de pérdida normal la divergencia KL expresada en la ecuación 29. [26]

Una vez definida esta nueva función de pérdida estamos en disposición de calcular tanto el estadístico T^2 y Q , que se calculan de la misma manera que en los autoencoders normales explicados en la sección anterior. Los umbrales también se calculan de la misma manera mediante el percentil 99, haciendo que el 99% de los datos de cada uno de los estadísticos estén por debajo del umbral.

Este tipo de Autoencoders se van a utilizar para mejorar sobre todo el estadístico T^2 ya que se consigue una capa oculta más eficaz.

Otra arquitectura a usar en este trabajo es emplear un Autoencoder pero donde las capas que definen esta red neuronal en vez de ser como las capas de propagación hacia adelante serán capas recurrentes tipo GRU. Es decir, cada una de las capas ocultas mostradas en la ilustración 18, serán capas formadas por neuronas recurrentes tipo GRU. A esta arquitectura la llamaremos Autoencoder GRU.



Universidad de Valladolid

Finalmente, también se utilizará un Autoencoder variacional con neuronas GRU, para comprobar la mejoría respecto a un Autoencoder de propagación hacia adelante, un Autoencoder GRU y un Autoencoder variacional.



CAPITULO III: PLANTA TENNESSEE-EASTMAN



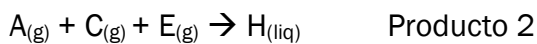


1. ORIGEN DE LA PLANTA TENNESSEE EASTMAN

La planta Tennessee Eastman (TEP) surgió en los años 90 por la compañía química Eastman y la universidad de Tennessee para desarrollar un modelo de planta que sirviera de referencia para el estudio del control de procesos multivariados. Está basada en un proceso real de la compañía Eastman, obteniendo así un modelo no lineal considerablemente complejo de un problema multivariable.

2. DESCRIPCION DE LA PLANTA

La planta TEP presenta 4 reactivos (A, C, D, E), un inerte (B) y un subproducto (F) a partir de los cuales se obtienen dos productos (G y H). Las reacciones que tienen lugar en el proceso son:



Estas reacciones son irreversibles y exotérmicas, y su equilibrio depende de la temperatura. La reacción que da lugar al producto G tiene una energía de activación mayor por lo que su sensibilidad frente a la temperatura será mayor. [8]

La planta consta principalmente de cinco operaciones unitarias: una reacción, una condensación, una separación vapor-líquido, una compresión, y un stripping. Además, hay elementos de control repartidos por todo el proceso: válvulas, indicadores, analizadores, etc. En la Ilustración 22 se observan las cinco partes del proceso donde se realizan las operaciones unitarias.

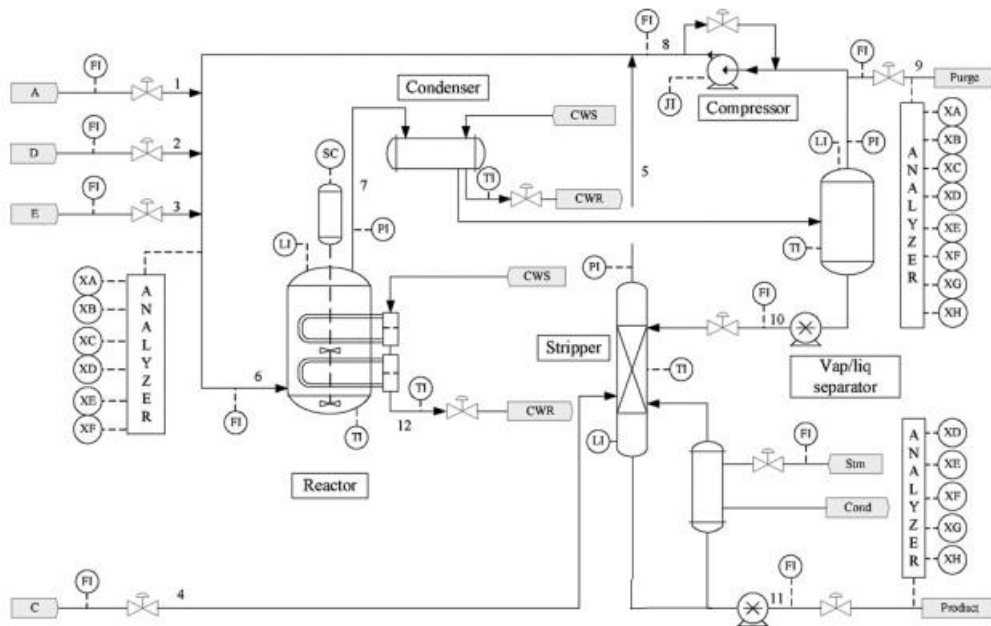


Ilustración 22 Proceso Tennessee Eastman [8]

El proceso comienza introduciendo los reactivos A, D y E al reactor del que salen los productos como vapor junto con los productos no reaccionados y se llevan al condensador y después al separador liquido-vapor.

Los productos no condensables se recirculan al compresor centrifugo para volverlos a juntar con los reactivos A, D y E que vuelven a alimentar el reactor.

Los condensables, una vez han pasado el separador liquido-vapor se dirigen a una columna de destilación donde se eliminan los reactivos remanentes mediante stripping donde se ha añadido el reactivo C.

La salida de base de la columna de destilación son los productos G y H, mientras que el inerte B y el subproducto F son purgados en el separador liquido-vapor. [10]

3. DATOS DE LA PLANTA

El modelo del proceso cuenta con 12 variables manipuladas (Tabla 1) y 41 variables medidas (Tabla 2), de las cuales las 22 primeras son variables del proceso y las 19 restantes son variables procedentes del muestreo de la planta cada cierto intervalo de tiempo.



Las 12 variables manipuladas del proceso se controlan mediante las 9 válvulas de flujo, 2 válvulas de control de temperatura y la velocidad de agitación que se pueden manipular en la planta.

En la planta pueden ocurrir 21 fallos distintos (Tabla 3), que en este proyecto se tiene como objetivo detectarlos. [8]

Variable	Variable	Unidades
XMV (1)	Flujo de alimentación D	Kgh^{-1}
XMV (2)	Flujo de alimentación E	Kgh^{-1}
XMV (3)	Flujo de alimentación A	kscmh
XMV (4)	Flujo de alimentación A y C	kscmh
XMV (5)	Válvula de recirculación del compresor	%
XMV (6)	Válvula de carga	%
XMV (7)	Flujo de líquido del separador LV	$\text{m}^3 \text{h}^{-1}$
XMV (8)	Flujo de líquido de la columna de stripping	$\text{m}^3 \text{h}^{-1}$
XMV (9)	válvula de vapor de la columna de stripping	%
XMV (10)	Flujo de agua de refrigeración del reactor	$\text{m}^3 \text{h}^{-1}$
XMV (11)	Flujo de agua en el condensador	$\text{m}^3 \text{h}^{-1}$
XMV (12)	Velocidad del agitador del reactor	rpm

Tabla 1 Variables manipuladas del proceso [8]



Variable	Variable	Unidades
XMEAS (1)	Flujo de alimentación A	kscmh
XMEAS (2)	Flujo de alimentación D	kscmh
XMEAS (3)	Flujo de alimentación E	kscmh
XMEAS (4)	Flujo de alimentación A y C	kscmh
XMEAS (5)	Flujo de recirculación	kscmh
XMEAS (6)	Flujo de alimentación reactor	kscmh
XMEAS (7)	Presión del reactor	kPa
XMEAS (8)	Nivel del reactor	%
XMEAS (9)	Temperatura del reactor	°C
XMEAS (10)	Flujo de purga	kscmh
XMEAS (11)	Temperatura del separador	°C
XMEAS (12)	Nivel del separador	%
XMEAS (13)	Presión del separador	kPa
XMEAS (14)	Corriente del separador	m ³ h ⁻¹
XMEAS (15)	Nivel de destilador (stripper)	%
XMEAS (16)	Presión del destilador (stripper)	kPa
XMEAS (17)	Corriente del destilador (stripper)	m ³ h ⁻¹
XMEAS (18)	Temperatura del destilador (stripper)	°C
XMEAS (19)	Flujo de vapor del destilador (stripper)	Kgh ⁻¹
XMEAS (20)	Potencia de compresor	kW
XMEAS (21)	Temperatura de la salida del agua de refrigeración del reactor	°C
XMEAS (22)	Temperatura de la salida del agua de refrigeración del separador	°C
XMEAS (23-28)	Concentración de la alimentación del reactor (A-F)	% mol
XMEAS (29-36)	Concentración de la purga (A-H)	% mol
XMEAS (37-41)	Concentración aguas abajo del destilador (A-H)	% mol

Tabla 2 Variables medidas del proceso [8]



Fallo	VARIABLES DE FALLO	Tipo de Fallo
IDV (1)	Relación de flujo de alimentaciones A/C, composición de B constante	Escalón
IDV (2)	Composición de B con relación A/C constante	Escalón
IDV (3)	Temperatura de alimentación D	Escalón
IDV (4)	Temperatura de entrada del agua del refrigerante del reactor	Escalón
IDV (5)	Temperatura de entrada del agua refrigerante del condensador	Escalón
IDV (6)	Pérdida de alimentación de A	Escalón
IDV (7)	Pérdida de presión en la corriente C	Escalón
IDV (8)	Composición de las alimentaciones A, B y C	Variación aleatoria
IDV (9)	Temperatura de alimentación D	Variación aleatoria
IDV (10)	Temperatura de alimentación C	Variación aleatoria
IDV (11)	Temperatura de entrada del agua refrigerante al reactor	Variación aleatoria
IDV (12)	Temperatura de entrada del agua de refrigerante del condensador	Variación aleatoria
IDV (13)	Cinética de las reacciones	Variación lenta
IDV (14)	Válvula del agua de refrigerante del reactor	Bloqueo
IDV (15)	Válvula del agua de refrigerante del condensador	Bloqueo
IDV (16)	Desconocido	No especificado
IDV (17)	Desconocido	No especificado
IDV (18)	Desconocido	No especificado
IDV (19)	Desconocido	No especificado
IDV (20)	Desconocido	No especificado
IDV (21)	Desconocido	Constante

Tabla 3 Fallos posibles en el proceso [8]

Se ha probado en otros estudios que hay fallos más difíciles de detectar que otros, como son los fallos 3, 9 y 15 ya que son variables que no sufren cambios significativos respecto al comportamiento normal del proceso por lo que se hace complicada la detección de dichos fallos. [8]

Para este proyecto se ha trabajado con dos conjuntos de datos de comportamiento normal de la planta, un conjunto de datos de 960 observaciones de cada variable para el entrenamiento y otro conjunto de datos de 500 observaciones de cada variable para la validación del entrenamiento.

El conjunto de datos de la ocurrencia de fallos son 960 observaciones de cada variable en el que el fallo ocurre a partir de la observación 160, por lo que el objetivo es intentar detectar el fallo lo más cercano posible a la observación 160 para definir una técnica rápida de detección del fallo.





CAPITULO IV: APLICACIÓN DESARROLLADA





1. INTRODUCCIÓN DE LA APLICACIÓN PRÁCTICA

Para desarrollar este proyecto se trabaja a partir de 23 simulaciones: dos simulaciones con la que se obtienen los datos de comportamiento normal de la planta, una de 960 observaciones de cada una de las 52 variables y otra de 500 observaciones de cada una de las 52 variables, y después habrá 21 simulaciones, una por cada fallo que puede ocurrir en la planta explicados en el capítulo anterior, en el que son 960 observaciones de cada una de las 52 variables en la que el fallo ocurre a partir de la observación 160.

Se trabajará con el lenguaje de programación Python, en el que se hará uso de la librería Keras de TensorFlow para la implementación de los distintos Autoencoders.

Primero se realizará el Análisis de Componentes Principales para la detección de fallos y se compararán los resultados obtenidos con los que se obtengan mediante Autoencoders, Autoencoders recurrentes GRU, Autoencoders variacionales y Autoencoders variacionales recurrentes GRU que se desarrollarán después.

2. DETECCIÓN DE FALLOS MEDIANTE ANÁLISIS DE COMPONENTES PRINCIPALES

Para realizar la aplicación de PCA, primero se realizan los cálculos fuera de línea tanto de las componentes principales con la reducción de la dimensionalidad del espacio, el cálculo de los estadísticos T^2 y Q explicados en el Capítulo II así como los umbrales para la detección de fallos mediante los datos de comportamiento normal de la planta.

Posteriormente, mediante los datos de fallos de la planta se realiza la detección de fallos en línea.

2.1 FUERA DE LÍNEA

Fuera de línea se define la matriz de datos de comportamiento normal X de dimensiones 500×52 . Este espacio $R^{500 \times 52}$ es el que se reducirá de dimensión mediante PCA.



A continuación, se normalizan los datos por columnas con media 0 y varianza 1, para que no haya diferencia de relevancia entre los datos. Se obtiene la matriz de correlación de X, llamada A, (ecuación 7) la cual se descompone en valores singulares y se extraen las matrices Λ , donde se almacenan los valores propios, y la matriz V donde se almacenan los vectores propios (ecuación 8).

A partir de la matriz Λ podemos obtener el número de componentes principales, a, a partir de realizar el test de porcentaje de varianza, en el que se va a buscar una variabilidad máxima del 90 %. Tras realizar este test se obtienen 31 componentes principales con una varianza del 89.01 %.

Una vez obtenidos el número de componentes principales se define la matriz de carga, llamada P, en el que se retienen los 31 vectores de carga correspondientes a los 31 valores singulares más grandes. A partir de la matriz P se proyectan los datos de la matriz X normalizados en un espacio reducido llamado T (ecuación 9) de dimensión $R^{500 \times 31}$.

Como ya se ha obtenido el espacio reducido, a partir del espacio de partida y el espacio reducido se puede recalculan los datos originales a partir del espacio T (ecuación 10) para después obtener una matriz de residuos siendo la diferencia entre el espacio recalculado a partir del espacio reducido y el espacio original (ecuación 11).

Una vez acabado el procedimiento anterior se calculan los estadísticos T^2 y Q para monitorizar el proceso. Para calcular T^2 se hace a partir de cada vector de observación, es decir, se tienen 500 vectores de observación (dimensión 1×52) al tener 500 observaciones de cada variable (ecuación 13). Para calcular el umbral del estadístico T^2 (ecuación 14) mediante el que se van a detectar los fallos se necesita el valor F_α que se obtiene con la tabla representada en la Ilustración 8 ya que se va a utilizar un grado de fiabilidad $\alpha = 0.01$, con grados de libertad $(31, 500-31)$ ya que los grados de libertad son $(a, n-a)$. En este caso se obtuvo un umbral con el que no se tiene ninguna muestra de T^2 por encima del umbral por lo que se define un umbral manualmente para tener un 0.2 % de las muestras por encima del umbral con el objetivo de que al realizar la detección de fallos no se produzcan falsas alarmas.

Para calcular Q es necesario calcular el vector de residuos (ecuación 15) para lo cual es necesario cada vector de observación (ecuación 16) como en el caso del cálculo de T^2 . Para calcular el umbral de Q es necesaria la distribución chi-cuadrado (ecuación 17) con el que se obtiene un 0.2 % de muestras por encima del umbral.

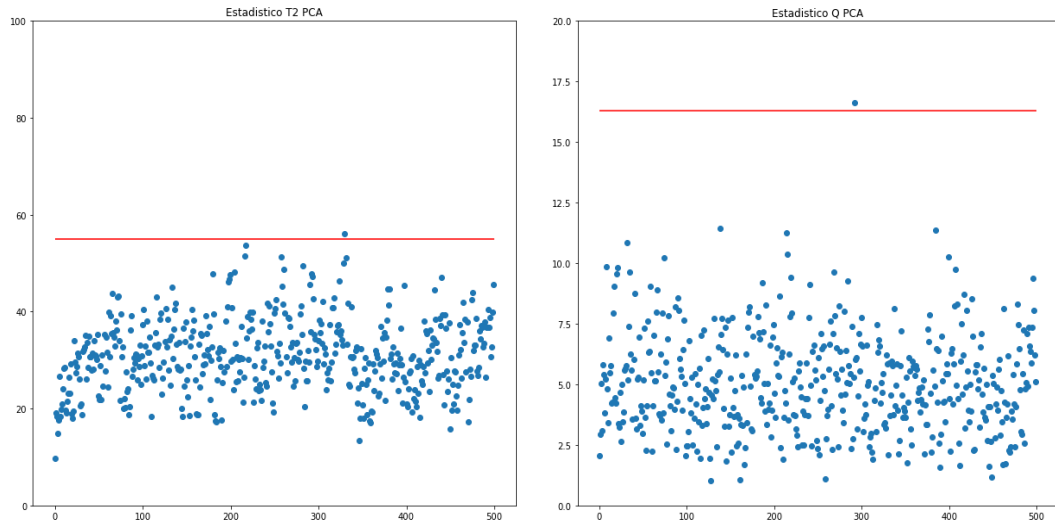


Ilustración 23 Estadísticos con los datos de comportamiento normal de la planta

2.2 EN LÍNEA

Una vez obtenidos los datos fuera de línea del PCA, se realiza la detección de fallos. Para realizarlo se obtienen los estadísticos T^2 y Q a partir de los datos reales de cada fallo y se comparan con el umbral obtenidos fuera de línea.

Para evitar la detección de falsas alarmas se busca que el umbral sea sobrepasado 10 veces consecutivos para determinar que ha ocurrido un fallo en la planta.

A continuación, se van a representar los resultados y gráficos más significativos de PCA.

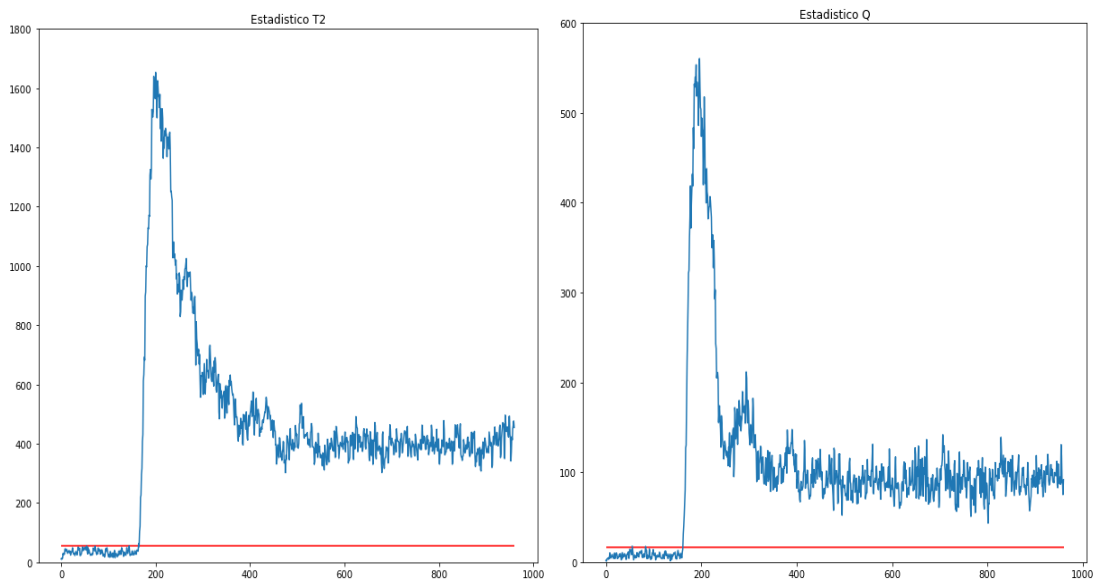


Ilustración 24 Detección del fallo 1 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

Universidad de Valladolid

En la Ilustración 24 se observa la detección del fallo 1 (relación de flujo de alimentaciones A/C) mediante ambos estadísticos. Se observa como en ambos estadísticos aproximadamente a partir de la observación 160 se supera el umbral, en concreto mediante el estadístico T^2 se detecta en la observación 166, es decir, 6 observaciones después de que se produzca y mediante el estadístico Q se detecta en la observación 162, es decir, 2 observaciones después de que se produzca el fallo.

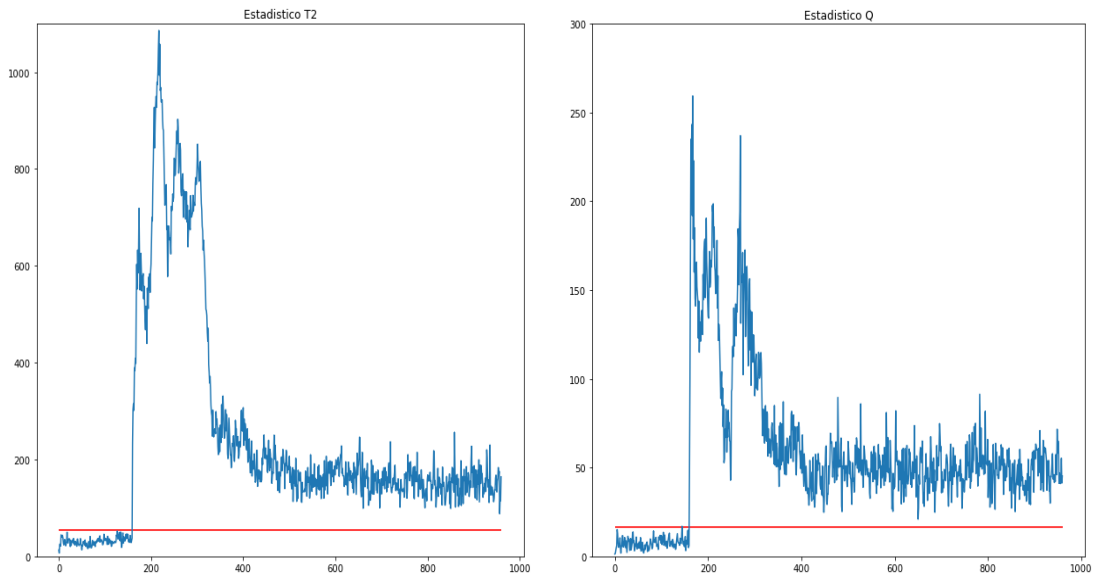


Ilustración 25 Detección del fallo 7 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

En la Ilustración 25 se observa la detección del fallo 7 (Pérdida de presión en la corriente C) que es el fallo que mejor se detecta ya que tanto con el estadístico T^2 como mediante el estadístico Q el fallo se detecta en la observación 160, es decir en cuanto ocurre el fallo.

En la Ilustración 26 se observa la detección del fallo 3 (Temperatura de alimentación D), en el que es un fallo que no se detecta mediante ninguno de los dos estadísticos.

En la Ilustración 27 se observa la detección del fallo 21 (Fallo desconocido) que es el fallo que mas se tarda en detectar en el proceso como se observa en las gráficas.

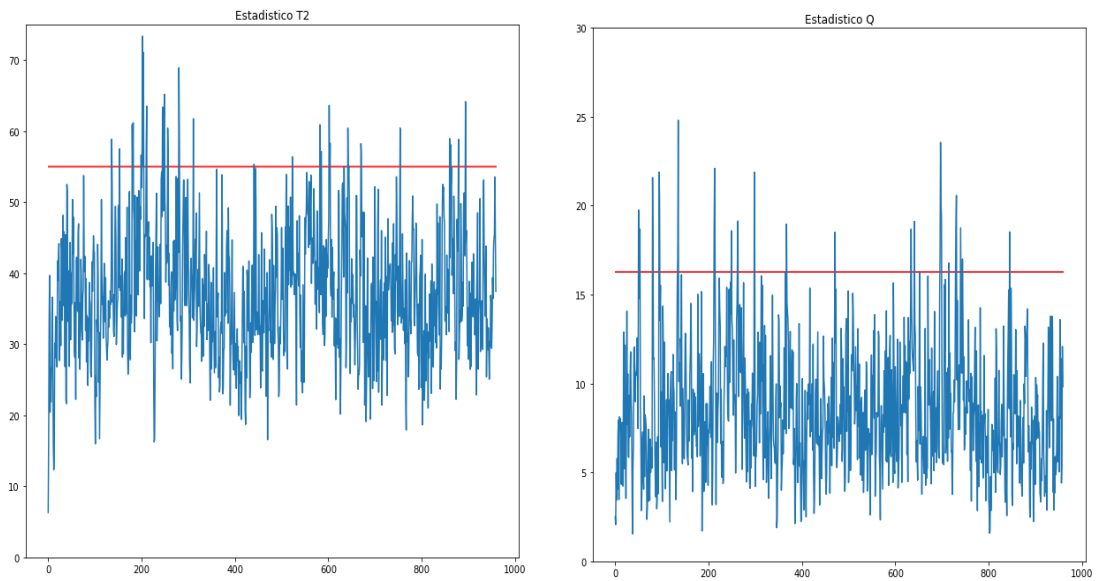


Ilustración 26 Detección del fallo 3 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

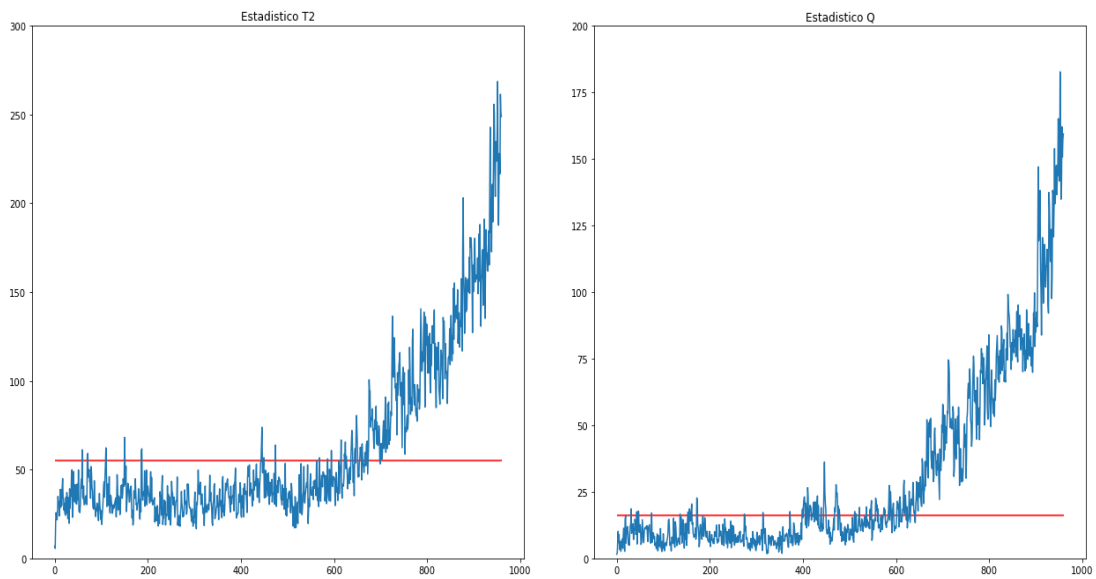


Ilustración 27 Detección del fallo 21 mediante PCA. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

A continuación, se va a presentar la tabla de resultados de la detección de fallos mediante PCA, en la que se va a observar el tiempo de detección de cada fallo para cada estadístico, es decir el número de observaciones que tarda en detectar el fallo desde que se produce, así como el porcentaje de fallos detectados, es decir, el número de observaciones de fallo que son detectados para cada fallo. También se va a representar en la tabla el porcentaje de falsas



alarmas en las 160 observaciones antes de que se produzca el fallo, es decir, datos de comportamiento normal que se detecte como fallo.

En los datos de comportamiento normal tanto para el estadístico T^2 como Q hay un 3.125% de falsas alarmas.

Fallo	Tiempo de detección mediante T^2	FDR (% fallos detectados) con T^2	% Falsas alarmas previas al fallo con T^2	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	6	99.375	0.625	2	99.75	1.25
2	14	98.375	1.25	10	98.875	4.375
3	No detección del fallo	4.25	1.25	No detección del fallo	3	5.62
4	63	59.75	2.5	0	99.625	2.5
5	10	28.75	2.5	2	26.5	2.5
6	6	99.25	0	0	99.99	1.25
7	0	99.99	0	0	99.99	0.625
8	22	97.5	1.25	17	95.5	0.625
9	No detección del fallo	5.375	7.5	No detección del fallo	3.75	3.75
10	96	47.5	0.625	98	48	3.75
11	187	57.25	2.5	55	54.875	2.5
12	6	99	1.25	22	96.5	2.5
13	44	95.375	0	41	95.125	1.25
14	0	99.99	1.875	1	96.625	2.5
15	No detección del fallo	7.875	1.25	No detección del fallo	6.25	1.25
16	196	28.75	13.125	192	44.5	2.5
17	26	82.5	0.625	21	95.75	2.5
18	87	89.875	1.875	83	90.25	1.25
19	No detección del fallo	13.5	1.875	No detección del fallo	16.25	0
20	86	43.75	1.25	80	55.25	0.625
21	514	39.875	3.125	482	52	5.625

Tabla 4 Resultados de la detección de fallos mediante PCA

Como se observa en la Tabla 4 hay fallos que no se detectan como son el fallo 3 (Temperatura de alimentación D), el fallo 9 (Temperatura de alimentación D), el fallo 15 (Valvula del agua de refrigerante del condensador) y el fallo 19 (Fallo desconocido).



El resto de fallos se consiguen detectar mediante PCA, algunos rapidamente como el fallo 7 (Pérdida de presión en la corriente C) que se detecta en cuanto ocurre el fallo, o el fallo 14 (Válvula de agua de refrigerante del reactor) que mediante el estadístico T^2 se detecta en cuanto ocurre y mediante el estadístico Q se detecta una observación después de que ocurra. También hay fallos que tardan mucho tiempo en detectarse como son el fallo 16 (Fallo desconocido) o el fallo 21 (Fallo desconocido) que es el que más tarda en detectarse, ya que mediante el estadístico T^2 tarda en detectarse 514 observaciones y mediante el estadístico Q tarda en detectarse 482 observaciones.

3. DETECCIÓN DE FALLOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Para la detección de fallos mediante aprendizaje automático se van a utilizar las técnicas explicadas en el capítulo II. En primer lugar, se utilizará una red de propagación hacia delante Autoencoder con 5 capas ocultas (sin contar la capa de entrada y la capa de salida) de dos maneras: Un Autoencoder con el encoder con capas decrecientes y el decoder con capas crecientes y otra red neuronal profunda con varias capas similar al autoencoder pero con el encoder con capas crecientes y el decoder con capas decrecientes, siempre siendo simétricas las capas alrededor de la capa oculta central. Después se utilizará un Autoencoder recurrente de las mismas dos maneras que en el caso anterior respecto a las capas y además se utilizarán distintas funciones de activación ya que hay que utilizar una función de activación de la propagación hacia delante y otra función de activación para la realimentación. La tercera técnica será un Autoencoder variacional que también se realizará de las dos maneras que las explicadas en el primer caso y también se utilizarán distintas funciones de activación. Por último, se utilizará un Autoencoder variacional recurrente.

Una vez realizadas estas 4 técnicas, además de buscar mejorar los resultados con PCA, se va a buscar cuál es la mejor técnica de aprendizaje automático para la detección de fallos de nuestro proceso.

3.1 REDES NEURONALES DE PROPAGACION HACIA DELANTE. AUTOENCODER

Se van a realizar distintos Autoencoder (explicados teóricamente en el capítulo II) con 5 capas ocultas sin contar la capa de entrada y la capa de salida que tienen la dimensión original de 52 variables.

3.1.1 AUTOENCODER SIMPLE DECRECIENTE-CRECIENTE

Lo primero de todo y al igual que en PCA es normalizar los datos de entrada por columnas entre 0 y 1.

A continuación, se desarrollará un Autoencoder con las siguientes dimensiones: la primera capa (y la quinta) tendrá 42 neuronas, la segunda capa (y la cuarta) tendrá 35 neuronas y la capa central tendrá 24 neuronas.

Antes de tener el Autoencoder completo se irán entrenando distintos Autoencoders capa a capa antes de juntarlo en el Autoencoder completo de la manera:

- Autoencoder 1: Dimensiones 52-42-52
- Autoencoder 2: Dimensiones 42-35-42
- Autoencoder 3: Dimensiones 35-24-35

Una vez entrenados estos Autoencoders por separado se juntan en el Autoencoder completo con la estructura final 52-42-35-24-35-42-52.

Se va a utilizar una función de activación sigmoide para todos los Autoencoders que sigue la siguiente expresión: [28]

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (31)$$

Otros parámetros que se utilizan para la compilación y entrenamiento del Autoencoder son:

- N.º de iteraciones (épocas) = 500
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM
- Función de pérdida: Error cuadrático medio

Una vez definidos estos parámetros se está en disposición de entrenar la red, conseguir el modelo del Autoencoder decreciente-creciente y realizar la detección de fallos.

En la Ilustración 28 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante el Autoencoder decreciente-creciente en el que se detecta el fallo mediante el estadístico T^2 en la observación 175 (15 observaciones después de que se produzca) y mediante el estadístico Q en la observación 163 (3 observaciones después de que se produzca).

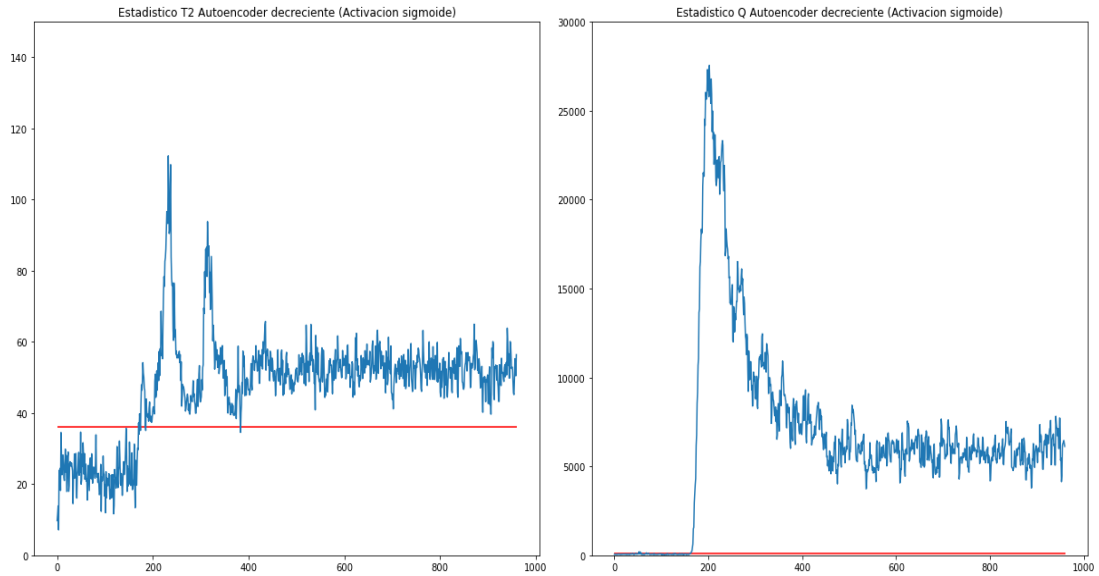


Ilustración 28 Detección del fallo 1 mediante Autoencoder decreciente-creciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

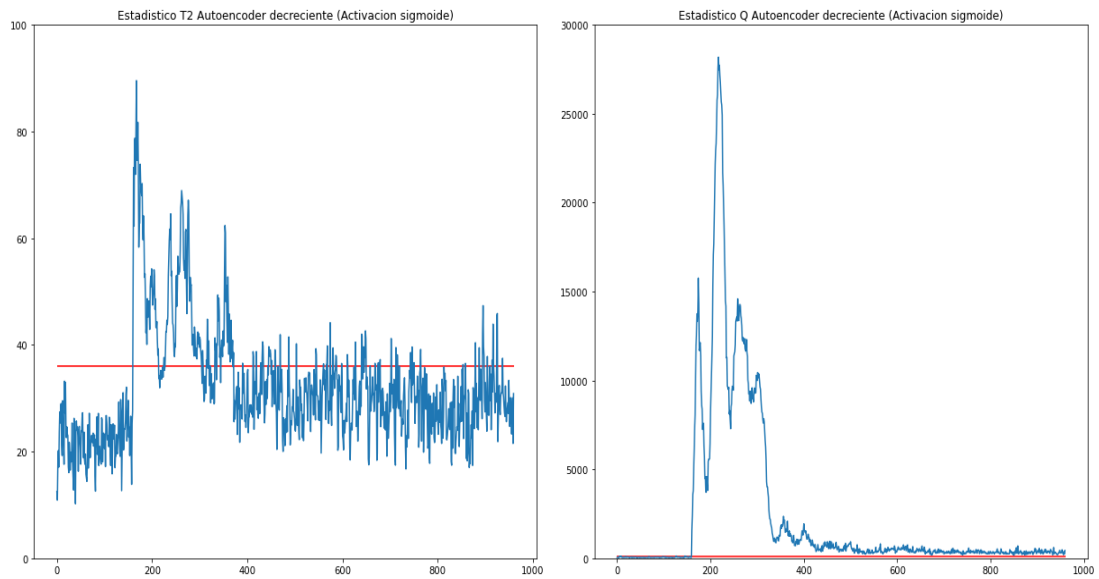


Ilustración 29 Detección del fallo 7 mediante Autoencoder decreciente-creciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

Como se observa en la Ilustración 29, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante PCA ya que se detecta en cuanto ocurre.

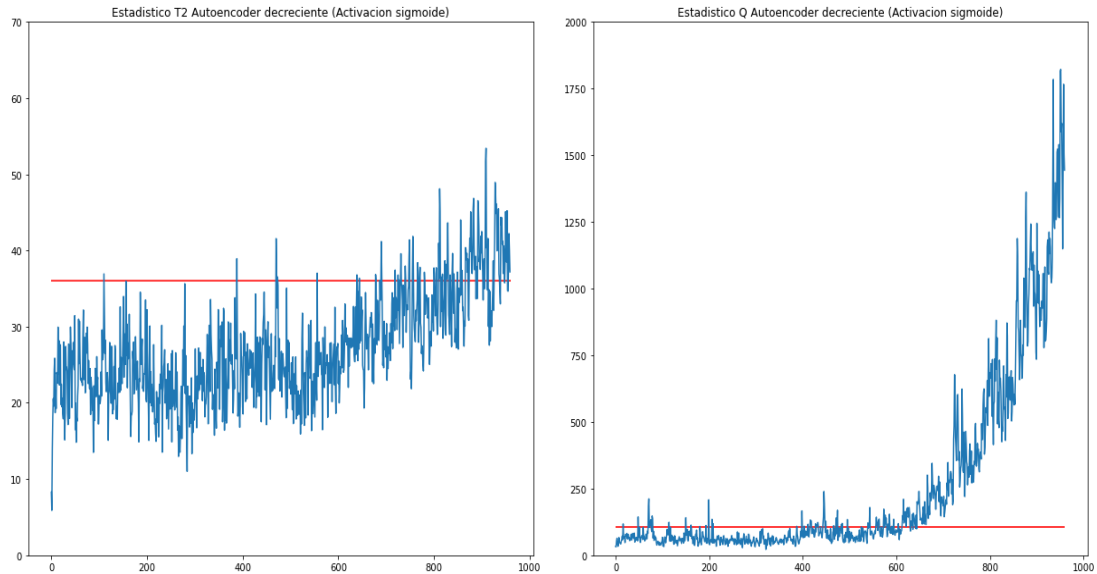


Ilustración 30 Detección del fallo 21 mediante Autoencoder decreciente-creciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 30 se observa la detección del fallo 21 (Fallo desconocido) y como es muy tardía la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 714 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 452 observaciones desde que ocurre el fallo en detectarlo.

A continuación, se representará la tabla (Tabla 5) de resultados de detección de fallos mediante el Autoencoder decreciente-creciente desarrollados.

En los datos de comportamiento normal para el estadístico T^2 hay 0% de falsas alarmas y para el estadístico Q hay un 0.625% de falsas alarmas.

Como se observa en la Tabla 5 mediante el Autoencoder decreciente-creciente mejoramos los resultados mediante el estadístico Q respecto a PCA ya que se consiguen detectar el fallo 15 (Válvula de agua de refrigerante del condensador) y el fallo 19 (desconocido) que con PCA no se detectaban, además se reduce el tiempo de detección de la mayoría de los otros fallos, sobre todo los que tardaban mucho en detectarse se reduce el tiempo notablemente. En contra partida el número de falsas alarmas en los fallos aumentan en comparación con los datos de comportamiento normal.

Por otro lado, se empeoran mucho los resultados mediante el estadístico T^2 ya que se utiliza la capa oculta, se dejan de detectar los fallos 4 (Temperatura de entrada del agua de refrigerante del reactor) y fallo 16 (Desconocido) que si se detectaban mediante PCA. Además, todos los fallos tardan más en detectarse, con la excepción del fallo 7 (Pérdida de presión en la corriente C) que se tarda lo mismo en detectar.

Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	15	98.25	0	3	99.625	4.375
2	29	97.25	1.25	10	98.75	3.75
3	No detección del fallo	0.125	0	No detección del fallo	9.125	10
4	No detección del fallo	9.375	0	2	99	4.375
5	17	13.875	0	0	99.99	4.375
6	24	9.125	0	0	99.99	1.875
7	0	30.5	0	0	99.99	3.125
8	56	83.625	0	19	97.99	1.875
9	No detección del fallo	0.5	1.25	No detección del fallo	9.25	13.125
10	545	4.25	0	24	84.375	1.25
11	374	22.75	0.625	6	73.375	0.625
12	69	75.5	0	2	99.75	10
13	46	78.875	1.25	36	95.5	1.875
14	32	82.125	0	0	99.99	1.875
15	No detección del fallo	0.875	0	671	13.75	3.125
16	No detección del fallo	2.125	0.625	9	89	18.75
17	111	50	0	23	95.125	3.125
18	97	87	0	83	90.75	5
19	No detección del fallo	3.25	0	10	80.125	1.875
20	90	17.375	1.25	69	84.75	1.875
21	714	15.125	0.625	452	49.625	6.875

Tabla 5 Resultados de la detección de fallos mediante Autoencoder decreciente-creciente

Como conclusión de esta primera técnica se llega a que hay que conseguir mejorar el estadístico T², es decir la capa oculta, para conseguir mejores resultados que PCA ya que mediante el estadístico Q ya se ha obtenido una mejoría.



3.1.2 RNN SIMPLE CRECIENTE-DECRECIENTE

A continuación, se desarrollará una red neuronal de varias capas con las siguientes dimensiones: la primera capa (y la quinta) tendrá 80 neuronas, la segunda capa (y la cuarta) tendrá 110 neuronas y la capa central tendrá 150 neuronas por lo que tenemos una red neuronal de dimensiones de las capas: 52-80-110-150-110-80-52.

Antes de tener la red completa se irán entrenando distintas redes capa a capa antes de juntarlo en una red completa de la manera:

- Red 1: Dimensiones 52-80-52
- Red 2: Dimensiones 80-110-80
- Red 3: Dimensiones 110-150-110

Una vez entrenadas estas redes por separado se juntan en una red neuronal completa con la estructura final 52-80-110-150-110-80-52.

Se va a utilizar una función de activación sigmoide para todas las redes.

Otros parámetros que se utilizan para la compilación y entrenamiento de esta red son:

- N.º de iteraciones (épocas) = 150
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM
- Función de pérdida: Error cuadrático medio

Una vez definidos estos parámetros estamos en disposición de entrenar la red, conseguir el modelo de la red creciente-decreciente y realizar la detección de fallos.

En la Ilustración 31 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante la RNN creciente-decreciente en el que se detecta el fallo mediante el estadístico T^2 en la observación 166 (6 observaciones después de que se produzca) y mediante el estadístico Q en la observación 164 (4 observaciones después de que se produzca).

Como se observa en la Ilustración 32, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

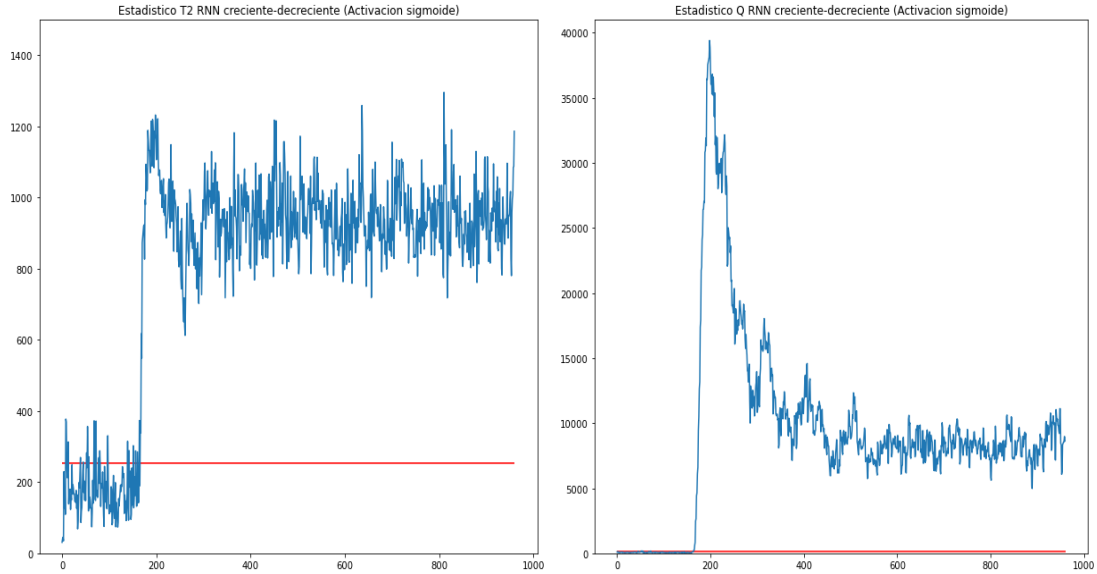


Ilustración 31 Detección del fallo 1 mediante RNN creciente-decreciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

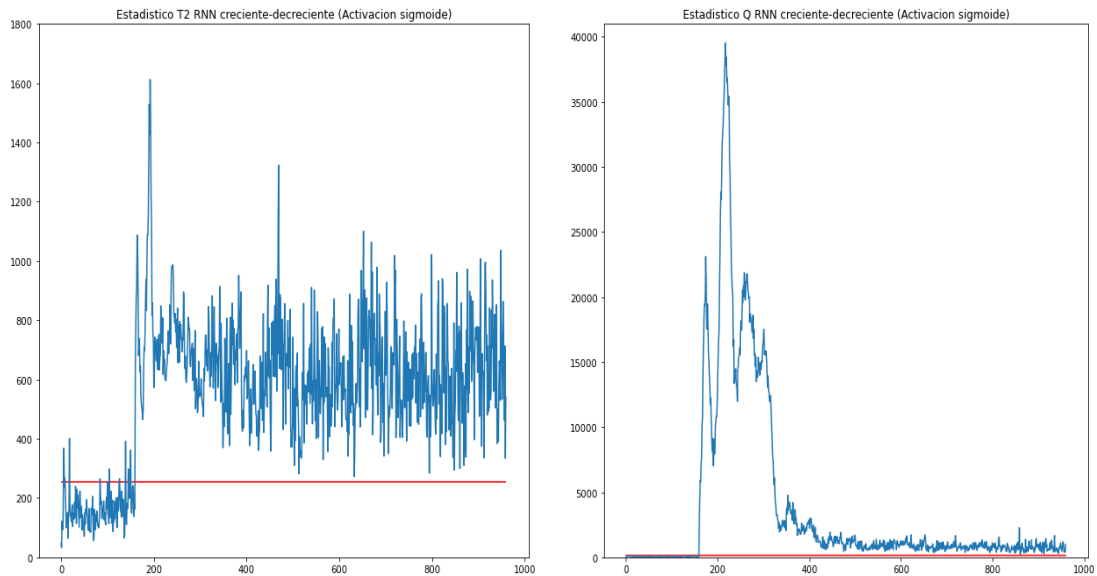


Ilustración 32 Detección del fallo 7 mediante RNN creciente-decreciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 33 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 510 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 504 observaciones desde que ocurre el fallo en detectarlo.

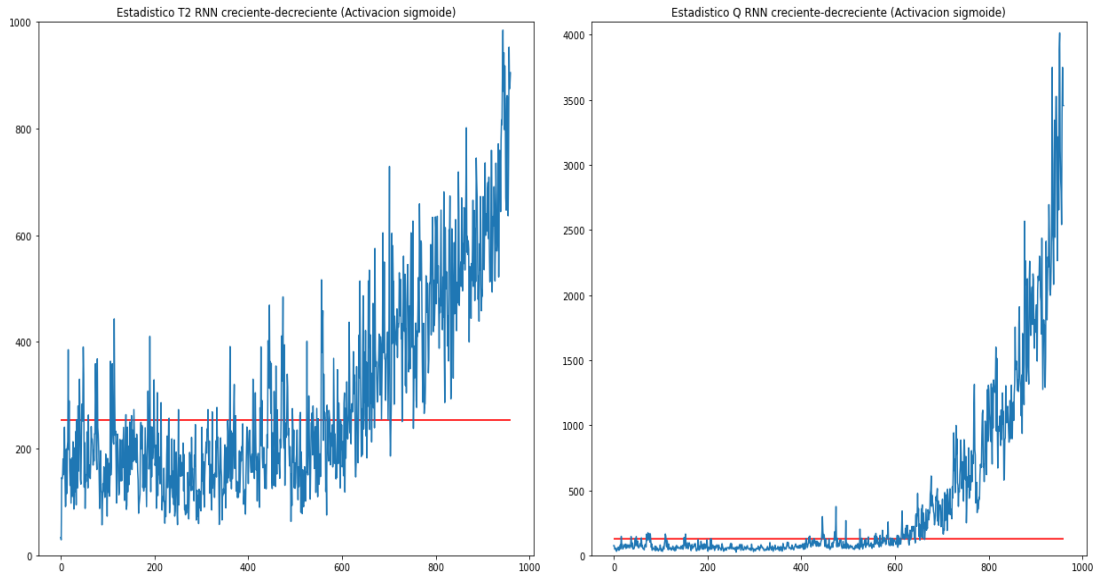


Ilustración 33 Detección del fallo 21 mediante RNN creciente-decreciente. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En los datos de comportamiento normal para el estadístico T^2 hay 6.25% de falsas alarmas y para el estadístico Q hay un 0% de falsas alarmas.

En la tabla 6 se observan los resultados de la detección de todos los fallos. En comparación con el Autoencoder decreciente-creciente se observa como mediante el estadístico T^2 se consiguen detectar el fallo 4 (Temperatura de entrada del agua del refrigerante del reactor) y el fallo 16 (Fallo desconocido), esto es debido a que la capa oculta es más eficiente al tener más neuronas. Como inconveniente a la mejora en el estadístico T^2 es que se aumenta notablemente el número de falsas alarmas respecto al comportamiento normal.

Respecto al estadístico Q entre el Autoencoder decreciente-creciente y la RNN creciente-decreciente no hay una gran diferencia de resultados ya que el fallo 3 (Temperatura de alimentación D) y el fallo 9 (Temperatura de alimentación D) siguen sin detectarse y el resto de fallos se detectan prácticamente en el mismo tiempo. El número de falsas alarmas es más alto que para el comportamiento normal sin ser notablemente alto a excepción de los fallos 3, 9, 16 y 21.

Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	6	99.5	12.5	4	99.5	3.125
2	11	98.875	12.5	10	98.75	3.75
3	No detección del fallo	17.875	16.25	No detección del fallo	7.625	8.75
4	354	46	19.375	2	92.125	3.125
5	6	39	19.375	0	99.99	3.125
6	9	99.5	8.75	0	99.99	0.625
7	0	99.99	7.5	0	99.99	1.25
8	22	98.125	11.875	19	97.875	1.25
9	No detección del fallo	18.25	16.875	No detección del fallo	7.125	13.125
10	145	44.375	8.125	21	80.875	1.875
11	188	62.5	11.25	10	69.99	0.625
12	21	98.625	10.625	2	99.5	6.875
13	43	95.625	11.875	36	95.625	1.25
14	0	99.75	10	0	99.99	1.875
15	No detección del fallo	19.25	13.75	676	12	1.25
16	310	31.875	20.625	9	81.99	19.375
17	8	84.875	12.5	23	93.375	2.5
18	87	91.625	11.875	83	90.625	1.875
19	No detección del fallo	35.875	11.875	10	74.875	1.25
20	84	48.5	13.125	71	74.5	1.875
21	510	49.5	13.125	504	46.5	9.375

Tabla 6 Resultados de la detección de fallos mediante RNN creciente-decreciente

3.1.3 AUTOENCODER GRU DECRECIENTE-CRECIENTE

A continuación, se van a desarrollar dos autoencoders recurrentes GRU decrecientes-crecientes: Uno con activación, tanto recurrente como la de propagación hacia adelante, sigmoide y otro con activación tangente hiperbólica y activación recurrente sigmoide.



Las dimensiones de estos Autoencoders decreciente-creciente será la misma que para el Autoencoder simple 52-42-35-24-35-42-52 para así poder realizar la comparación entre ellos.

Antes de tener el Autoencoder completo se irán entrenando distintos Autoencoders capa a capa antes de juntarlo en el Autoencoder completo de la manera:

- Autoencoder 1: Dimensiones 52-42-52
- Autoencoder 2: Dimensiones 42-35-42
- Autoencoder 3: Dimensiones 35-24-35

Una vez entrenados estos Autoencoders por separado se juntan en el Autoencoder completo con la estructura final 52-42-35-24-35-42-52.

Otros parámetros que se utilizan para la compilación y entrenamiento del Autoencoder recurrente son:

- N.º de iteraciones (épocas) = 500
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM

3.1.3.1 ACTIVACIÓN SIGMOIDE – ACTIVACIÓN RECURRENTE SIGMOIDE

Una vez definidos estos parámetros estamos en disposición de entrenar la red, conseguir el modelo del Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide y realizar la detección de fallos.

En la Ilustración 34 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante el Autoencoder decreciente-creciente GRU con activación sigmoide y activación recurrente sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 167 (7 observaciones después de que se produzca) y mediante el estadístico Q en la observación 163 (3 observaciones después de que se produzca).

Como se observa en la Ilustración 35, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

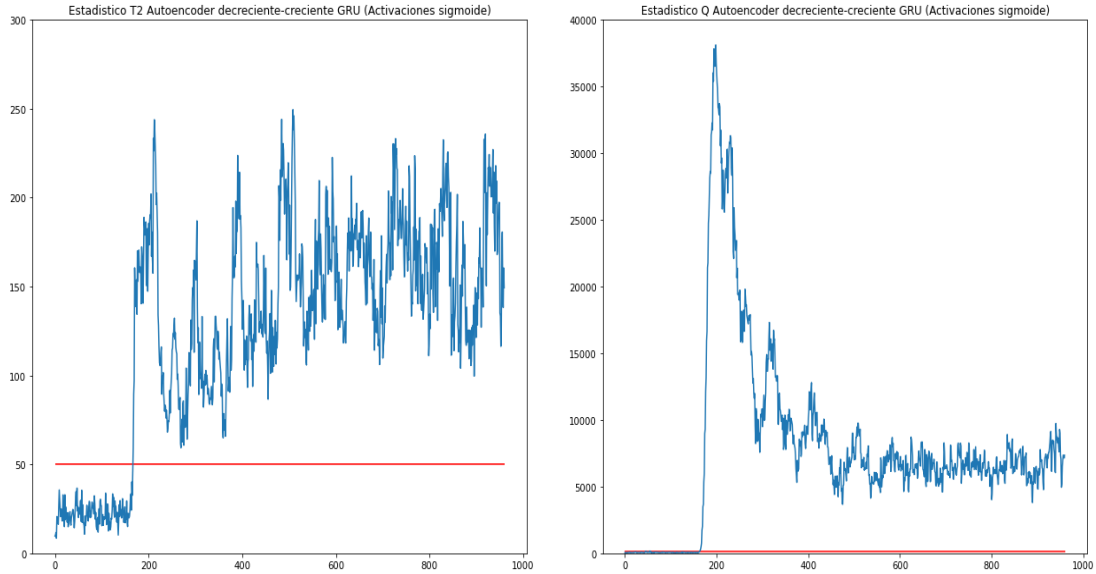


Ilustración 34 Detección del fallo 1 mediante Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

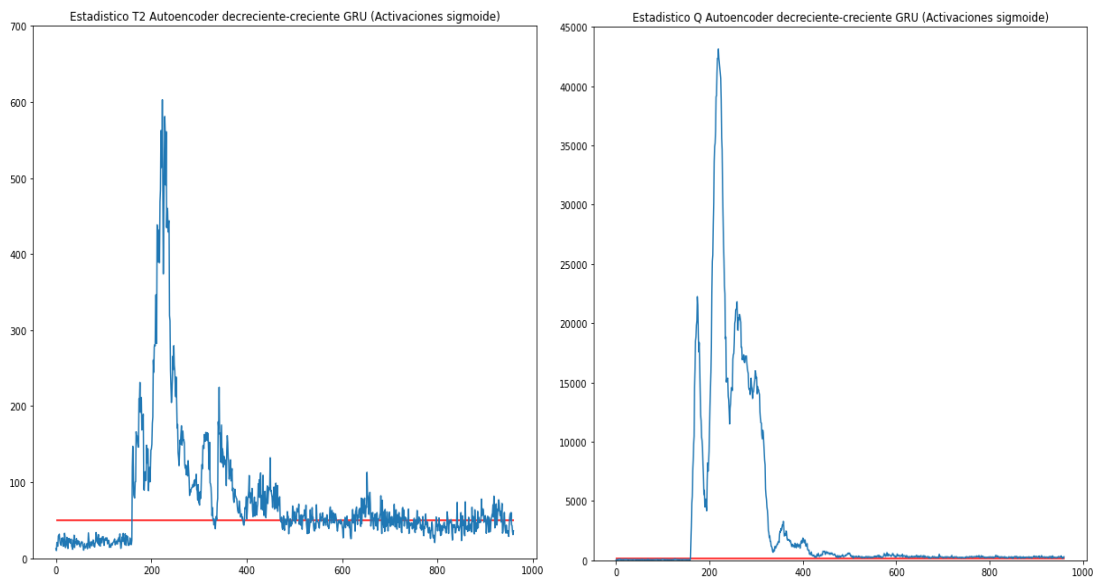


Ilustración 35 Detección del fallo 7 mediante Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 36 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 560 observaciones desde que ocurre el fallo en

detectarlo y mediante el estadístico Q se tarda 484 observaciones desde que ocurre el fallo en detectarlo.

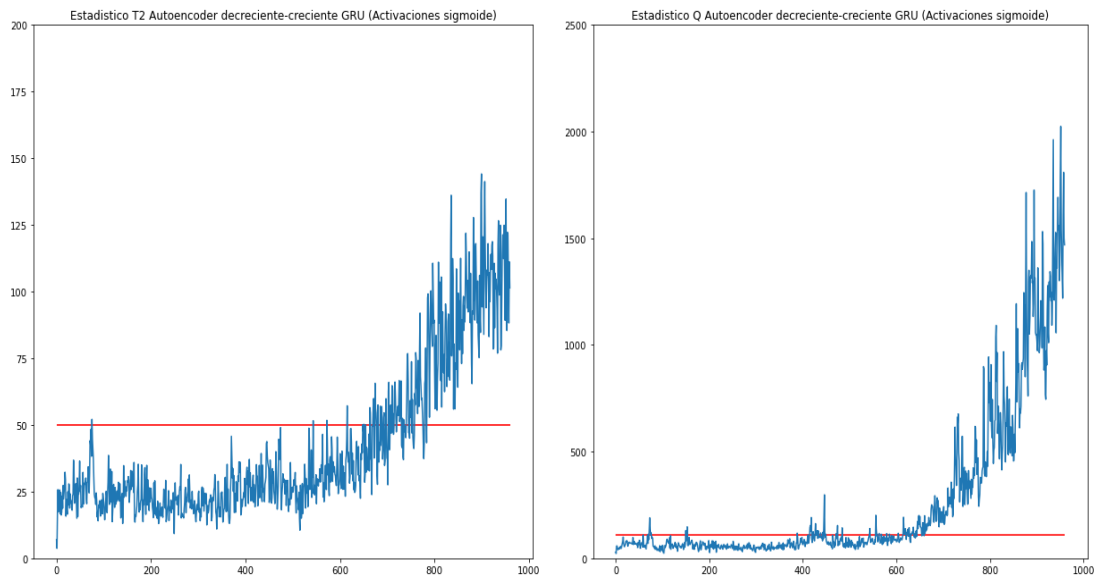


Ilustración 36 Detección del fallo 21 mediante Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

A continuación, se representará la tabla de resultados (Tabla 7) de detección de fallos mediante el Autoencoder decreciente-creciente GRU con ambas activaciones sigmoide.

En los datos de comportamiento normal para el estadístico T^2 hay 0% de falsas alarmas y para el estadístico Q hay un 0.625% de falsas alarmas.

En comparación con el Autoencoder simple decreciente-creciente se observa como la detección de fallos mediante el estadístico T^2 mejora notablemente en los resultados. En primer lugar, el fallo 16 (Fallo desconocido) que no se detectaba mediante el Autoencoder simple ahora si se consigue detectar mediante el Autoencoder recurrente. Los fallos 3,9, 15 y 19 siguen sin detectarse mediante el estadístico T^2 . Se consigue un mejor tiempo de detección en todos los fallos menos en el fallo 13 que se retrasa en 2 observaciones la detección. Notable es la mejoría en el fallo 10 que pasa a detectarse a las 545 observaciones desde que ocurre el fallo mediante el Autoencoder simple a detectarse en 96 observaciones desde que ocurre el fallo.

Mediante el estadístico Q los resultados son muy similares al Autoencoder simple, aunque se aumentan el número de falsas alarmas en comparación con el comportamiento normal.



Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	7	99.125	0	3	99.625	2.5
2	25	97.125	0	11	98.625	1.875
3	No detección del fallo	1.25	1.25	No detección del fallo	9.375	8.125
4	No detección del fallo	1.375	0	2	99.125	1.875
5	12	19.75	0	0	99.99	1.875
6	10	98.75	0	0	99.99	0.625
7	0	63.99	0	0	99.99	0.625
8	25	94.625	0.625	19	97.75	0.625
9	No detección del fallo	1.125	0	No detección del fallo	7.25	14.375
10	96	24.875	0	24	83.5	3.125
11	304	16.25	0	10	69.99	1.875
12	22	91.875	1.875	2	99.75	8.75
13	48	93.99	0	42	95.25	0
14	7	88.5	0	0	99.99	0.625
15	No detección del fallo	2.625	0	626	12.625	1.25
16	624	11.5	5	9	86.75	16.25
17	30	72.25	0	23	94.875	1.875
18	94	88.5	0	83	90.375	1.875
19	No detección del fallo	1.125	0	10	76.99	0
20	86	25.125	0	69	78.375	0
21	560	30.99	0.625	484	46.125	5.625

Tabla 7 Resultados de la detección de fallos mediante Autoencoder decreciente-creciente GRU sigmoid-sigmoid

3.1.3.2 ACTIVACIÓN TANGENTE HIPERBÓLICA - ACTIVACIÓN RECURRENTE SIGMOIDE

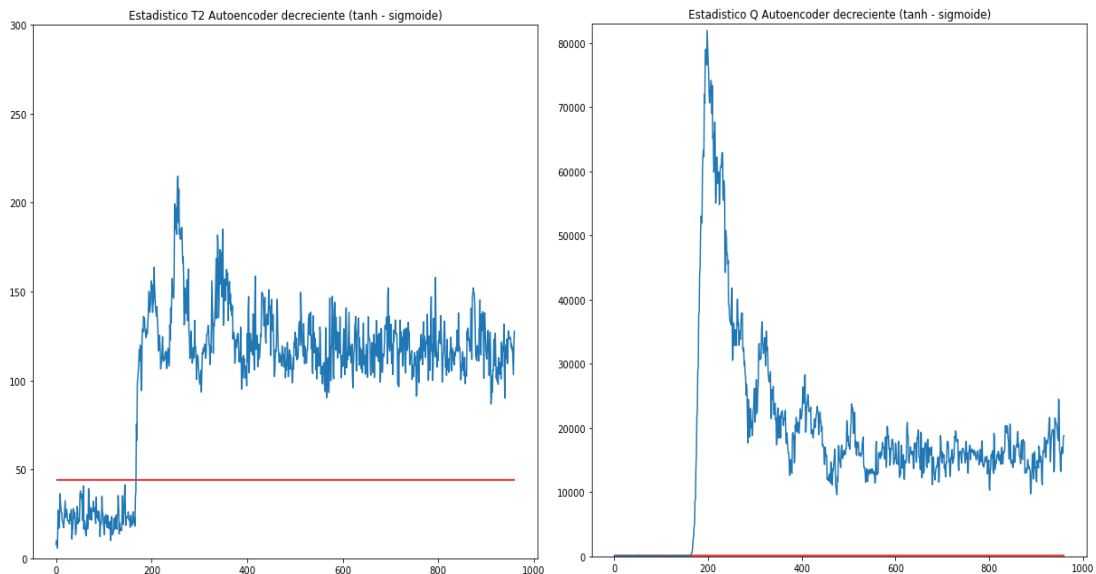


Ilustración 37 Detección del fallo 1 mediante Autoencoder decreciente-creciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 37 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante el Autoencoder decreciente-creciente GRU con activación tangente hiperbólica y activación recurrente sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 168 (8 observaciones después de que se produzca) y mediante el estadístico Q en la observación 160 (nada más ocurre el fallo).

Como se observa en la Ilustración 38, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

En la Ilustración 39 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 621 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 453 observaciones desde que ocurre el fallo en detectarlo.

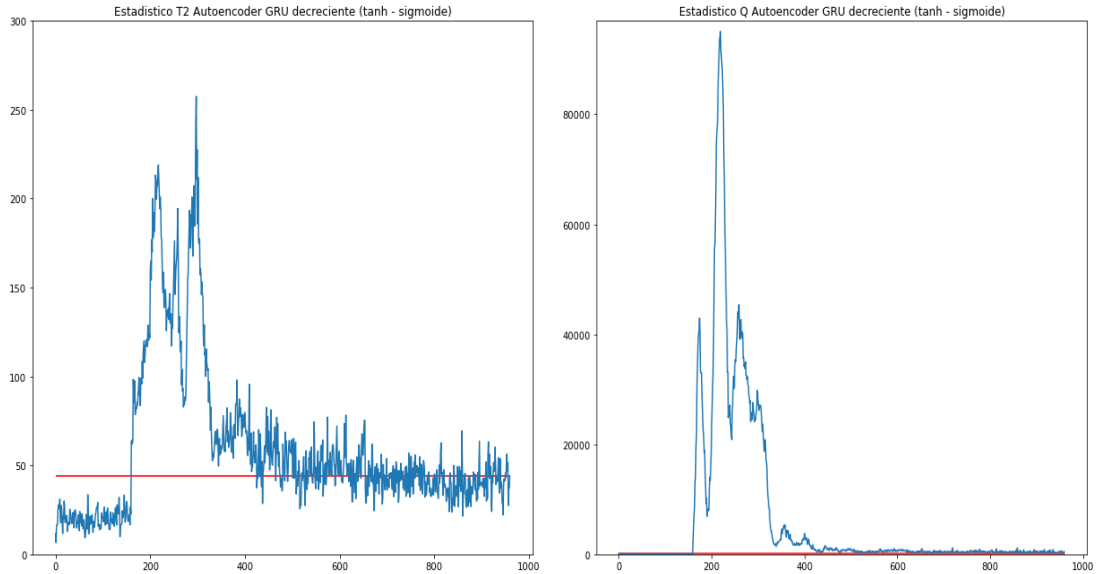


Ilustración 38 Detección del fallo 7 mediante Autoencoder decreciente-creciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

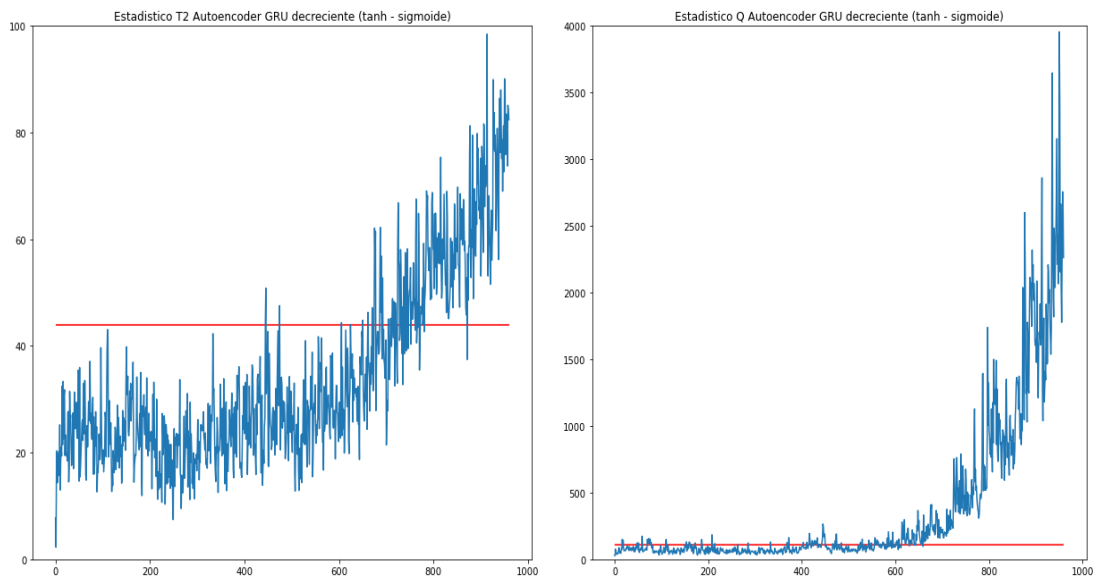


Ilustración 39 Detección del fallo 21 mediante Autoencoder decreciente-creciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

A continuación, se representará la tabla de resultados (Tabla 8) de detección de fallos mediante el Autoencoder decreciente-creciente GRU con activaciones tangente hiperbolica y activación recurrente sigmoide.

En los datos de comportamiento normal para el estadístico T² hay 0% de falsas alarmas y para el estadístico Q hay un 1.25% de falsas alarmas.

Fallo	Tiempo de detección mediante T^2	FDR (% fallos detectados) con T^2	% Falsas alarmas previas al fallo con T^2	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	8	98.99	0	0	99.99	6.25
2	12	98.5	0.625	10	98.875	6.25
3	No detección del fallo	0.625	0	82	15.625	13.125
4	No detección del fallo	11.75	0	0	99.875	8.125
5	14	20.375	0	0	99.99	8.125
6	9	98.99	0	0	99.99	4.375
7	0	67.125	0	0	99.99	2.5
8	23	96.75	0	19	98.375	5
9	No detección del fallo	0.25	2.5	2	14.5	26.875
10	500	13.25	0	13	85.375	3.125
11	462	25.375	0	10	76.5	6.25
12	22	95.75	0	2	99.625	13.125
13	48	91.375	0.625	36	95.625	4.375
14	14	88.375	0	0	99.99	6.25
15	No detección del fallo	0.875	0	630	22.25	3.75
16	664	6	0.625	9	89.25	30
17	30	68.75	0	23	95.75	7.5
18	100	88	0	78	91.625	4.375
19	No detección del fallo	4.25	0	10	83.375	3.125
20	No detección del fallo	7.75	0	69	80.99	6.25
21	621	29.99	0	453	52.125	9.375

Tabla 8 Resultados de la detección de fallos mediante Autoencoder decreciente-creciente GRU tanh-sigmoid

En comparación con el Autoencoder simple decreciente-creciente se observa como la detección de fallos mediante el estadístico T^2 mejora notablemente en los resultados. En primer lugar, el fallo 16 (Fallo desconocido) que no se detectaba mediante el Autoencoder simple ahora si se consigue detectar mediante el Autoencoder recurrente, pero en contraposición el fallo 20 que si se detectaba mediante el Autoencoder decreciente-creciente simple ahora no



se detecta mediante este Autoencoder recurrente con estas activaciones. Los fallos 3, 9, 15 y 19 siguen sin detectarse mediante el estadístico T^2 . Se consigue un mejor tiempo de detección en la mayoría de los fallos sin que el número de falsas alarmas se aumente.

Mediante el estadístico Q los resultados son muy similares al Autoencoder simple, con la diferencia de que el fallo 3 (Temperatura de alimentación D) y el fallo 9 (Temperatura de alimentación D) se consiguen detectar mediante este tipo de Autoencoder algo que es muy valioso en la detección de fallos ya que mediante este estadístico se detectan todos los fallos. En contra posición, se aumentan notablemente las falsas alarmas en la detección de fallos en concreto en esos dos fallos y en el fallo 16 que tiene unas falsas alarmas del 30% de los datos de comportamiento normal antes de que se produzca el fallo.

Este Autoencoder decreciente-creciente con activación tangente hiperbólica y activación recurrente sigmoide mejora en los resultados sobre todo mediante el estadístico Q que se consiguen detectar todos los fallos, mediante el estadístico T^2 se consigue reducir el tiempo de detección en la mayoría de los fallos, aunque hace que no se detecte el fallo 20 (Fallo desconocido) que si se detectaba mediante el Autoencoder decreciente-creciente simple.

En comparación con el otro Autoencoder recurrente GRU desarrollado con ambas activaciones sigmoide no mejora en cuanto al estadístico T^2 ya que el fallo 20 deja de detectarse mediante este tipo de Autoencoder. Los tiempos de detección son muy similares. En cuanto al estadístico Q se mejora notablemente ya que se consiguen detectar todos los fallos, ya que mediante el otro Autoencoder no se conseguían detectar los fallos 3 y 9 pero se aumentan notablemente las falsas alarmas mediante este estadístico.

3.1.4 RNN GRU CRECIENTE-DECRECIENTE

A continuación, se van a desarrollar dos redes neuronales recurrentes GRU crecientes-decrecientes: Uno con activación, tanto recurrente como la de propagación hacia adelante, sigmoide y otro con activación tangente hiperbólica y activación recurrente sigmoide.

Las dimensiones de estas redes recurrentes creciente-decreciente será la misma que para la red simple 52-80-110-150-110-80-52 para así poder realizar la comparación entre ellos.

Antes de tener la red completa se irán entrenando distintas redes capa a capa antes de juntarlo en la red completa de la manera:

- Red 1: Dimensiones 52-80-52
- Red 2: Dimensiones 80-110-80
- Red 3: Dimensiones 110-150-110

Una vez entrenadas estas redes por separado se juntan la red neuronal completa con la estructura final 52-80-110-150-110-80-52.

Otros parámetros que se utilizan para la compilación y entrenamiento de esta red recurrente son:

- N.º de iteraciones (épocas) = 100
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM

3.1.4.1 ACTIVACIÓN SIGMOIDE – ACTIVACIÓN RECURRENTE SIGMOIDE

Una vez definidos estos parámetros estamos en disposición de entrenar la red, conseguir el modelo de la red recurrente creciente-decreciente GRU con ambas activaciones sigmoide y realizar la detección de fallos.

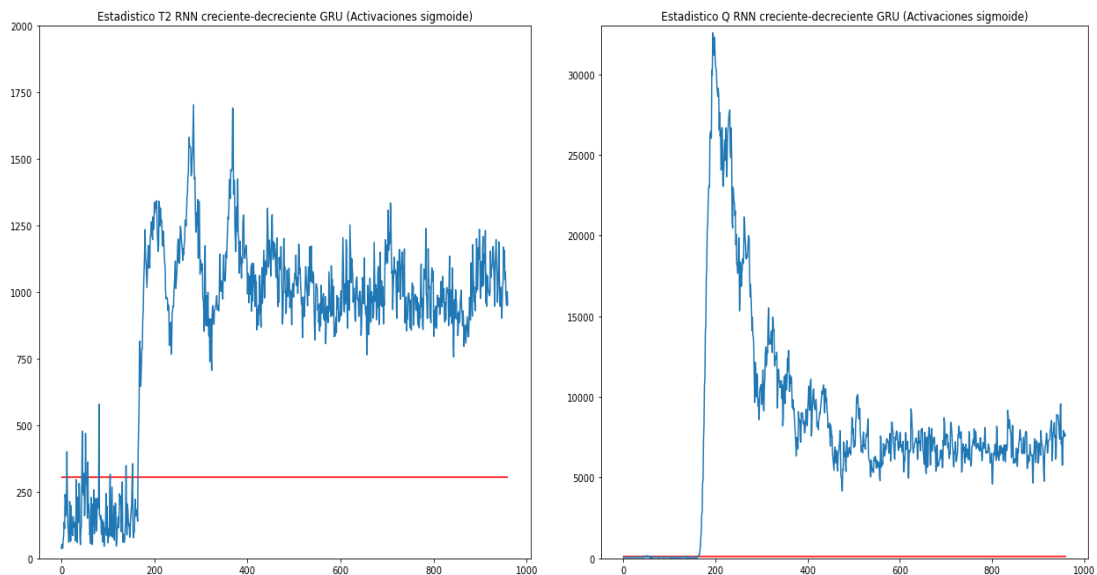


Ilustración 40 Detección del fallo 1 mediante RNN creciente-decreciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 40 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante la RNN creciente-decreciente GRU con activación sigmoide y activación recurrente sigmoide en el que se detecta el fallo mediante el estadístico T² en la observación 165 (5 observaciones después de que se produzca) y mediante el

estadístico Q en la observación 163 (3 observaciones después de que se produzca).

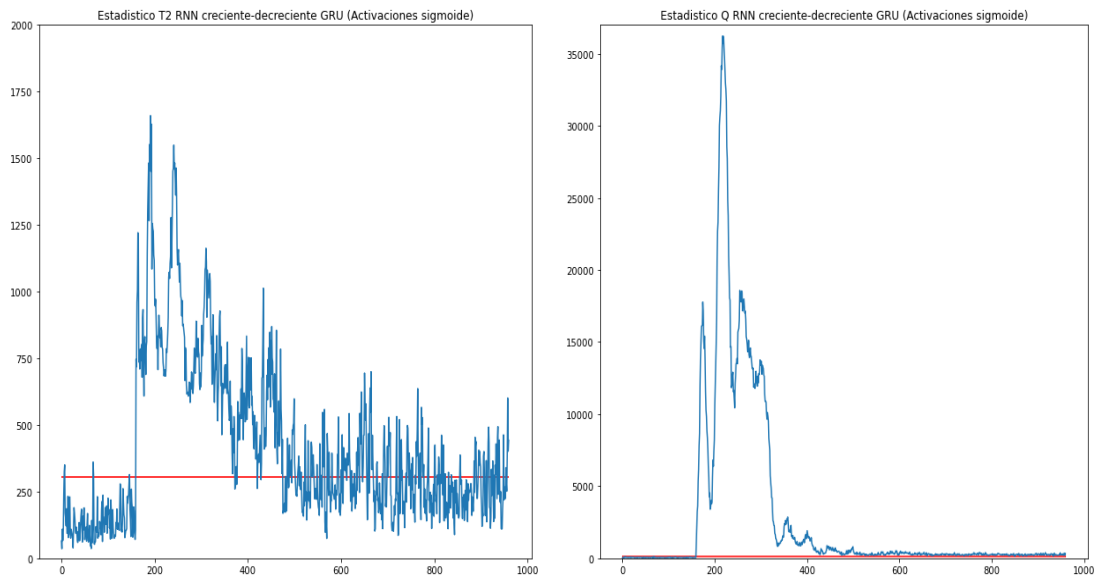


Ilustración 41 Detección del fallo 7 mediante RNN creciente-decreciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

Como se observa en la Ilustración 41, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

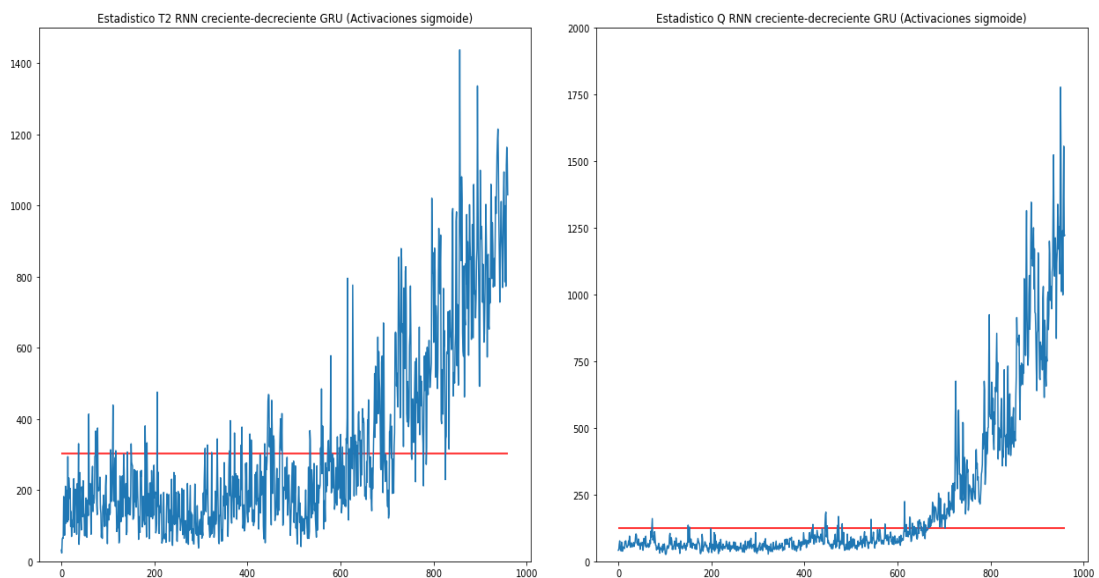


Ilustración 42 Detección del fallo 21 mediante RNN creciente-decreciente GRU con ambas activaciones sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 42 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 511 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 506 observaciones desde que ocurre el fallo en detectarlo.

Fallo	Tiempo de detección mediante T^2	FDR (% fallos detectados) con T^2	% Falsas alarmas previas al fallo con T^2	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	5	99.375	6.875	3	99.625	1.25
2	15	98.5	5	11	98.625	1.875
3	No detección del fallo	10	11.25	No detección del fallo	4.125	4.375
4	No detección del fallo	16.125	8.125	2	92.5	1.25
5	8	36.125	8.125	0	99.99	1.25
6	10	98.875	3.125	0	99.99	0.625
7	0	62.5	3.125	0	99.99	0
8	22	97.625	9.375	19	97.875	0.625
9	No detección del fallo	9.625	11.875	No detección del fallo	3.25	8.125
10	82	41.5	1.875	24	79.75	0
11	378	38.75	7.5	24	61.125	0
12	21	98	7.5	2	99.5	1.25
13	46	95.25	3.75	43	95.375	1.25
14	3	92.875	4.375	1	99.875	0
15	No detección del fallo	11.375	9.375	672	5.875	1.25
16	No detección del fallo	27.375	23.75	9	82.125	12.5
17	27	79.99	3.75	23	92.75	0
18	93	90.125	6.875	83	89.75	1.25
19	No detección del fallo	21.875	5.625	10	68.25	0
20	83	44.625	4.375	70	72.99	0
21	511	42.125	5.625	506	40.125	1.875

Tabla 9 Resultados de la detección de fallos mediante RNN creciente-decreciente GRU sigmoid-sigmoid

En la Tabla 9 se representan los resultados de la detección de fallos mediante la RNN creciente-decreciente GRU con ambas activaciones sigmoide.

En los datos de comportamiento normal para el estadístico T^2 hay 3.75% de falsas alarmas y para el estadístico Q hay un 0% de falsas alarmas.

En comparación con la RNN simple creciente-decreciente se observa como la detección de fallos es muy similar mediante ambos estadísticos en el tiempo de detección, pero empeoran los resultados ya que hay fallos que no se detectan.

Mediante el estadístico T^2 , se dejan de detectar los fallos 4 (Temperatura de entrada del agua del refrigerante del reactor) y los fallos 16 (Fallo desconocido) que si se detectaban mediante la RNN simple. Los fallos 3, 9, 15 y 19 siguen sin detectarse en comparación con la RNN simple. Además, aumentan las falsas alarmas considerablemente.

Mediante el estadístico Q, los fallos 3 y 9 siguen sin detectarse y el tiempo de detección de todos los fallos es muy similar, y las falsas alarmas están bastante controladas.

3.1.4.2 ACTIVACIÓN TANGENTE HIPERBÓLICA - ACTIVACIÓN RECURRENTE SIGMOIDE

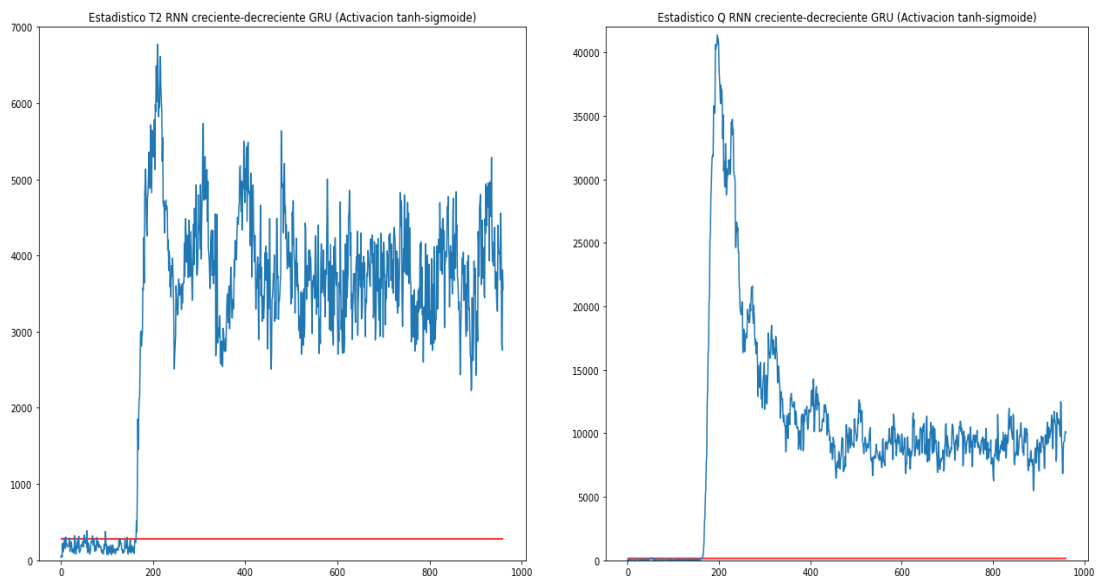


Ilustración 43 Detección del fallo 1 mediante RNN creciente-decreciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 43 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante la RNN creciente-decreciente GRU con activación tangente hiperbólica y activación

recurrente sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 164 (4 observaciones después de que se produzca) y mediante el estadístico Q en la observación 163 (3 observaciones después de que se produzca el fallo).

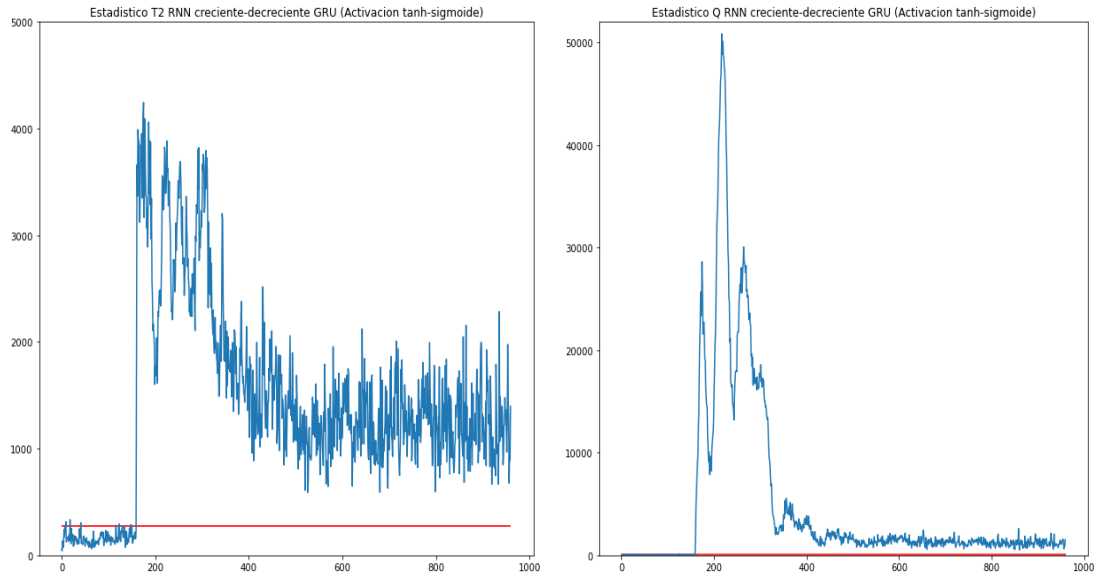


Ilustración 44 Detección del fallo 7 mediante RNN creciente-decreciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

Como se observa en la Ilustración 44, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

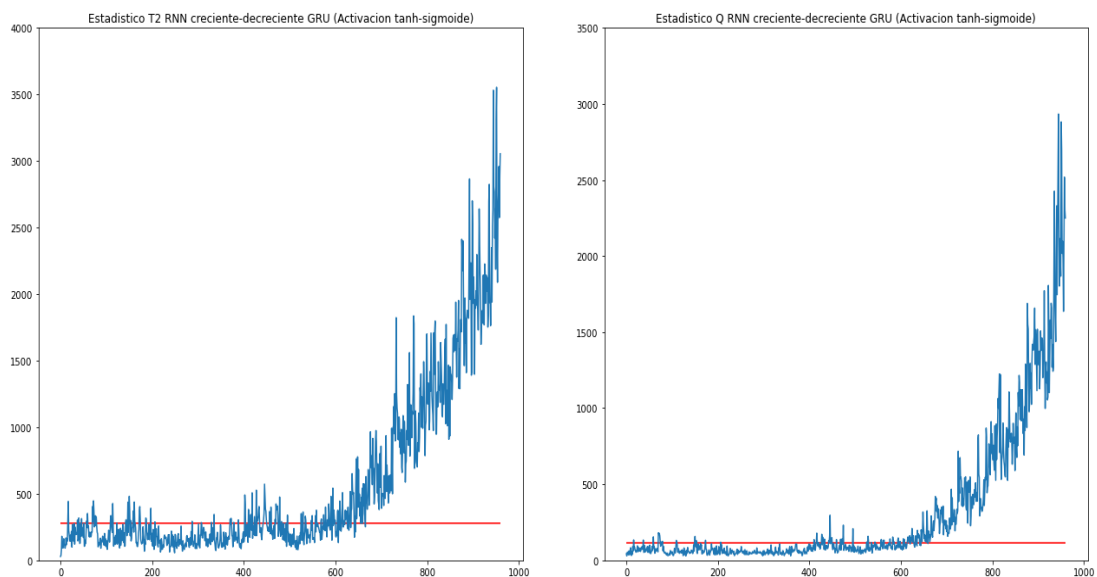


Ilustración 45 Detección del fallo 21 mediante RNN creciente-decreciente GRU con activaciones tanh-sigmoide. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

En la Ilustración 45 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante ambos estadísticos se tarda 484 observaciones desde que ocurre el fallo en detectarlo.

A continuación, se representará la tabla de resultados (Tabla 10) de detección de fallos mediante la RNN creciente-decreciente GRU con activación tangente hiperbólica y activación recurrente sigmoide.

En los datos de comportamiento normal para el estadístico T^2 hay 6.875% de falsas alarmas y para el estadístico Q hay un 0.625% de falsas alarmas.

Fallo	Tiempo de detección mediante T^2	FDR (% fallos detectados) con T^2	% Falsas alarmas previas al fallo con T^2	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	4	99.5	9.375	3	99.625	4.375
2	11	98.625	10.625	10	98.875	3.75
3	No detección del fallo	19.125	13.125	354	14.75	11.875
4	55	82.125	11.875	2	99.125	4.375
5	0	40.375	11.875	0	70.625	4.375
6	4	99.625	5.625	0	99.99	1.875
7	0	99.99	3.75	0	99.99	1.25
8	17	98.5	13.125	13	98.375	2.5
9	No detección del fallo	18.5	19.375	No detección del fallo	12.75	23.125
10	14	58.99	7.5	15	81.375	2.5
11	46	73.25	11.25	9	76.875	2.5
12	6	99.25	10.625	2	99.625	5.625
13	40	95.5	5.625	36	95.625	3.125
14	0	99.99	8.125	0	99.99	5.625
15	577	19.5	7.5	630	19	1.875
16	195	46.99	25	10	83.875	23.75
17	24	93.75	9.375	23	94.625	5
18	87	91.75	10	83	91.5	3.75
19	74	32.875	6.875	59	68.99	3.125
20	80	58.625	5.625	71	76.875	3.75
21	484	52.375	16.875	484	47.75	10

Tabla 10 Resultados de la detección de fallos mediante RNN creciente-decreciente GRU tanh-sigmoid

En comparación con la RNN simple creciente-decreciente se observa como la detección de fallos mediante el estadístico T^2 mejora notablemente en los



resultados. En primer lugar, se consiguen detectar los fallos 15 (Válvula del agua de refrigerante del condensador) y 19 (Fallo desconocido) que mediante la RNN simple creciente-decreciente no se conseguían detectar. En segundo lugar, se mejora el tiempo de detección en todos los fallos, notable es la mejoría en la detección del fallo 4 (Temperatura de entrada del agua del refrigerante del reactor) que se consigue detectar 299 observaciones más rápido que mediante la RNN simple o el fallo 11 (Temperatura de entrada del agua refrigerante al reactor) que se consigue detectar 142 observaciones más rápido que mediante la RNN simple. En contra partida, las falsas alarmas son notablemente altas.

Mediante el estadístico Q los resultados son muy similares a la RNN simple, con la diferencia de que el fallo 3 (Temperatura de alimentación D) se consigue detectar mediante este tipo de red neuronal recurrente algo que es muy valioso en la detección de fallos, el fallo 9 (Temperatura de alimentación D) sigue sin detectarse mediante este tipo de red neuronal recurrente.

Mediante ambos estadísticos, se siguen detectando muchas falsas alarmas, muy similar a las que se detectaban mediante la RNN simple.

En comparación con la otra RNN desarrollada con ambas activaciones sigmoide se mejoran mucho los resultados. En primer lugar, mediante el estadístico T^2 se consiguen detectar los fallos 4, 15, 16 y 19 que no se consiguen detectar mediante la RNN creciente-decreciente GRU con ambas activaciones sigmoide. Además, se mejora el tiempo de detección en todos los fallos. En segundo lugar, mediante el estadístico Q se consigue detectar el fallo 3 que no se conseguía detectar mediante la otra red neuronal recurrente. El tiempo de detección del resto de fallos se mejora, pero las falsas alarmas son notablemente más altas.

3.1.5 AUTOENCODER VARIACIONAL DECRECIENTE-CRECIENTE

A continuación, se van a desarrollar dos Autoencoders variacionales decrecientes-crecientes: Uno con activación sigmoide y otro con activación ReLu.

Las dimensiones de estos Autoencoders decreciente-creciente será la misma que para el Autoencoder simple 52-42-35-24-35-42-52 para así poder realizar la comparación entre ellos.

Antes de tener el Autoencoder completo se irán entrenando distintos Autoencoders capa a capa antes de juntarlo en el Autoencoder completo de la manera:

- Autoencoder 1: Dimensiones 52-42-52
- Autoencoder 2: Dimensiones 42-35-42

A diferencia que el Autoencoder simple y el Autoencoder recurrente no se entrena un tercer Autoencoder por separado para entrenar previamente la capa oculta, sino que la capa oculta se consigue directamente al unir los Autoencoders entrenados por separado en el Autoencoder variacional final.

Una vez entrenados estos Autoencoders por separado se juntan en el Autoencoder completo con la estructura final 52-42-35-24-35-42-52.

Otros parámetros que se utilizan para la compilación y entrenamiento del Autoencoder recurrente son:

- N.º de iteraciones (épocas) = 500
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM

3.1.5.1 ACTIVACION SIGMOIDE

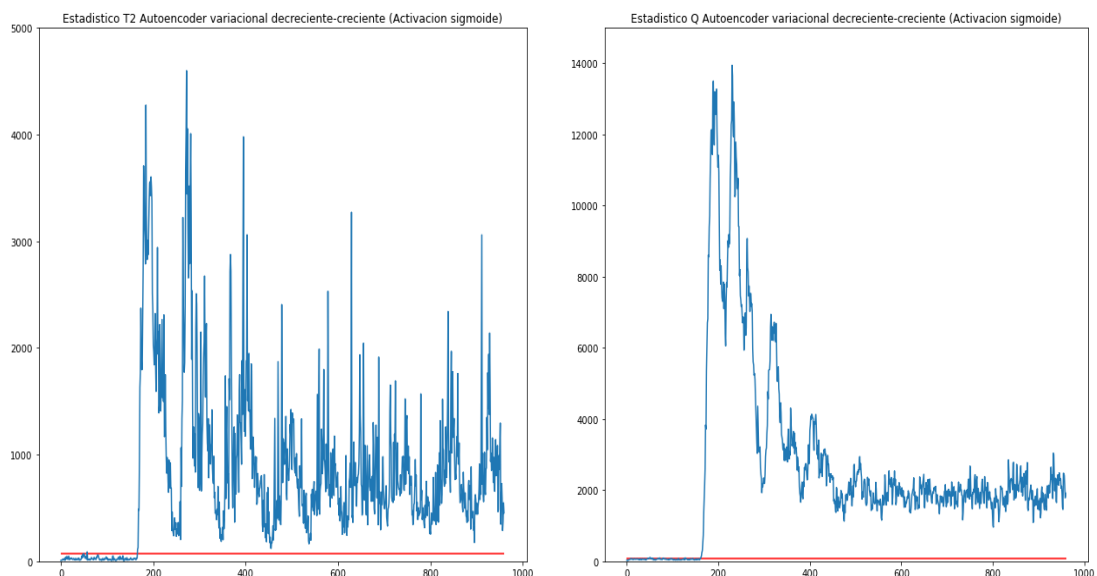


Ilustración 46 Detección del fallo 1 mediante Autoencoder Variacional decreciente-creciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 46 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante el Autoencoder Variacional decreciente-creciente con activación sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 166 (6 observaciones después de que se produzca) y mediante el estadístico Q en la observación 162 (2 observaciones después de que se produzca el fallo).

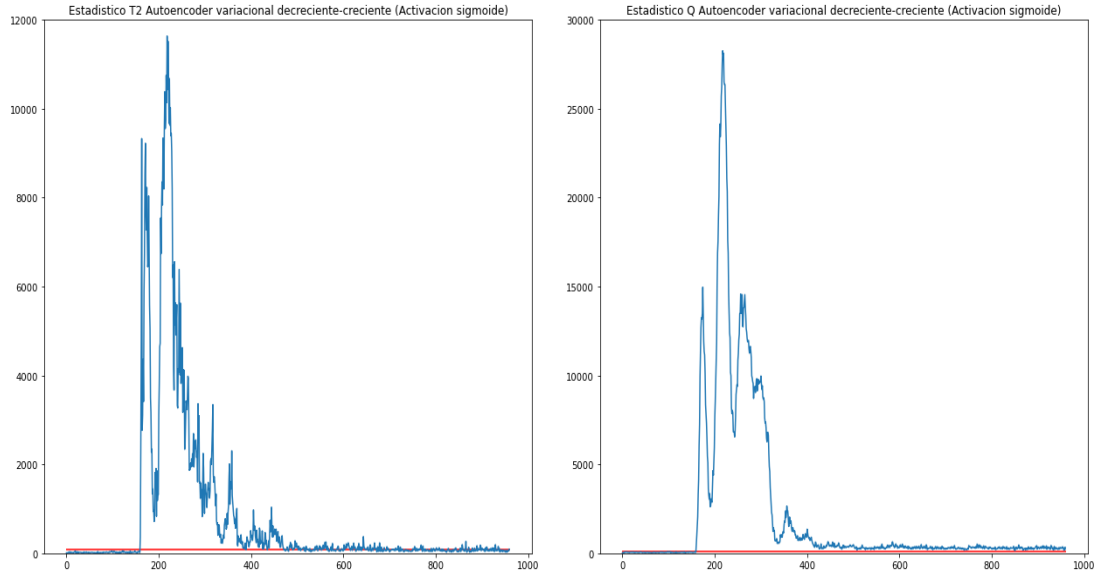


Ilustración 47 Detección del fallo 7 mediante Autoencoder Variacional decreciente-creciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

Como se observa en la Ilustración 47, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

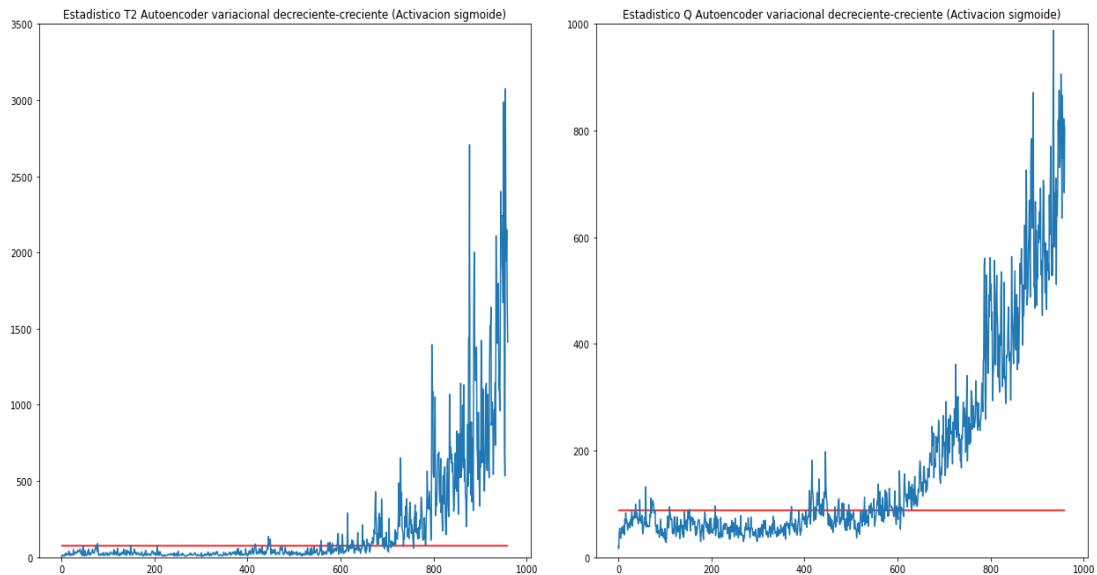


Ilustración 48 Detección del fallo 21 mediante Autoencoder Variacional decreciente-creciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 48 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 505 observaciones desde que ocurre el fallo en



detectarlo y mediante el estadístico Q se tarda 454 observaciones desde que ocurre el fallo en detectarlo.

A continuación, se representará la tabla de resultados (Tabla 11) de detección de fallos mediante el Autoencoder variacional decreciente-creciente con activación sigmoide.

En los datos de comportamiento normal para el estadístico T^2 hay 1.25% de falsas alarmas y para el estadístico Q hay un 1.875% de falsas alarmas.

En comparación al Autoencoder simple decreciente-creciente desarrollado anteriormente que también se utiliza la activación sigmoide se observa como mediante el estadístico T^2 en primer lugar, se mejoran los resultados ya que se consigue detectar el fallo 16 (Fallo desconocido) que mediante el Autoencoder simple decreciente-creciente no se conseguían detectar, aunque los fallos 3, 4, 9, 15 y 19 siguen sin conseguir detectarse. En segundo lugar, se mejora notablemente el tiempo de detección de los fallos que es el objetivo de la utilización del Autoencoder Variacional ya que se modifica la forma de obtener la capa oculta, y, por tanto, el estadístico T^2 . Notable es la mejoría en la detección del fallo 10 (Temperatura de alimentación C) que se consigue detectarlo 448 observaciones antes que mediante el Autoencoder simple, o el fallo 21 (Fallo desconocido) que se consigue detectarlo 209 observaciones antes que mediante el Autoencoder simple. Además, el número de falsas alarmas, exceptuando los fallos 3, 9 y 16, son bastante normales.

Mediante el estadístico Q se consiguen detectar los fallos 3 y 9 que no se conseguían detectar mediante el Autoencoder simple, mientras que el tiempo de detección de fallos se mejora en casi todos los fallos, aunque en pequeña medida. El número de falsas alarmas, con excepciones son bastante normales.

Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	6	99.25	0.625	2	99.75	4.375
2	30	96.99	0	11	98.625	1.875
3	No detección del fallo	3.5	3.125	394	10.75	8.75
4	No detección del fallo	11.375	1.25	0	99.99	1.25
5	11	23.625	1.25	0	99.99	1.25
6	3	99.625	0	0	99.99	0.625
7	0	71.5	0	0	99.99	0.625
8	22	94.99	1.875	15	98.375	6.25
9	No detección del fallo	3.875	4.375	227	11.25	21.25
10	97	34.625	0	20	85.375	0.625
11	302	29.99	0.625	5	78.25	1.875
12	21	91.99	1.25	2	99.75	10
13	43	94.875	0	35	95.625	1.25
14	1	99.625	0.625	0	99.99	0.625
15	No detección del fallo	3.5	0.625	626	16.5	1.25
16	618	16.875	13.75	9	92.25	23.125
17	23	84.5	1.25	21	96.5	2.5
18	91	89.125	0.625	78	91.125	3.75
19	No detección del fallo	3.75	0.625	9	88.5	0.625
20	84	38.875	0.625	69	87.125	1.875
21	505	40.125	2.5	454	53.375	8.125

Tabla 11 Resultados de la detección de fallos mediante Autoencoder Variacional decreciente-creciente activación sigmoide

3.1.5.2 ACTIVACION RELU

A continuación, se va a desarrollar otro Autoencoder variacional realizado de la misma manera que el Autoencoder anterior, pero utilizando como función de activación la función ReLu para intentar conseguir mejores resultados.

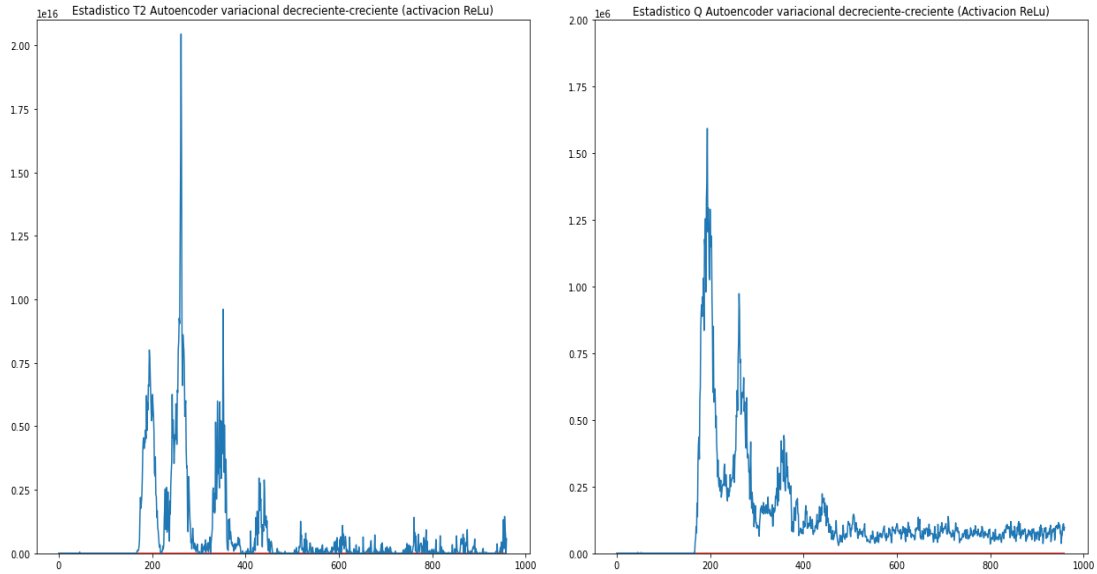


Ilustración 49 Detección del fallo 1 mediante Autoencoder Variacional decreciente-creciente activación ReLu. Izquierda, Detección mediante estadístico T². Derecha, Detección mediante estadístico Q

En la Ilustración 49 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante el Autoencoder Variacional decreciente-creciente con activación ReLu en el que se detecta el fallo mediante el estadístico T² en la observación 166 (6 observaciones después de que se produzca) y mediante el estadístico Q en la observación 165 (5 observaciones después de que se produzca el fallo).

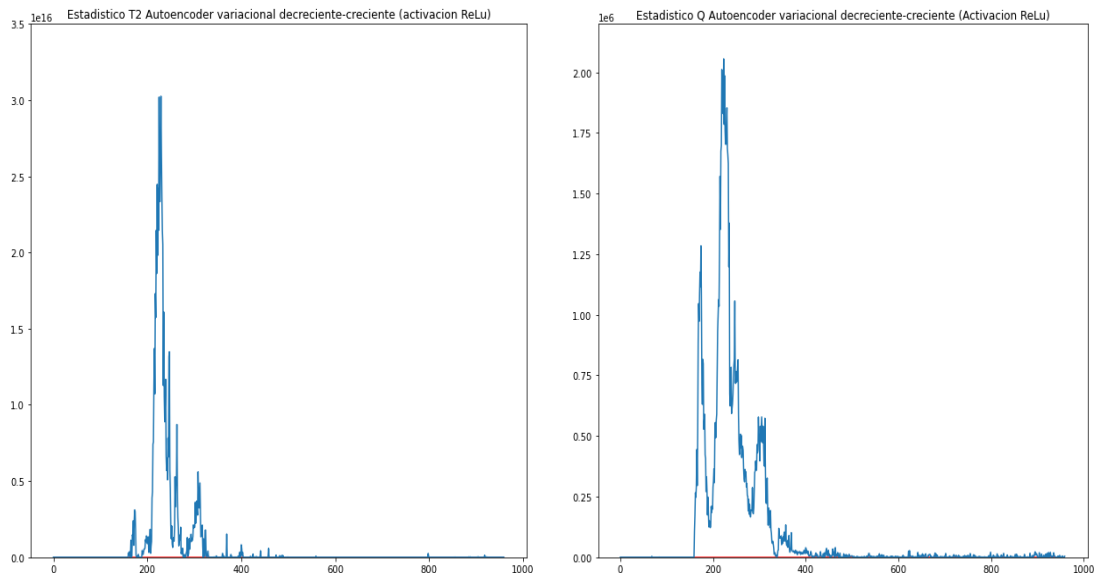


Ilustración 50 Detección del fallo 7 mediante Autoencoder Variacional decreciente-creciente activación ReLu. Izquierda, Detección mediante estadístico T². Derecha, Detección mediante estadístico Q

Como se observa en la Ilustración 50, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

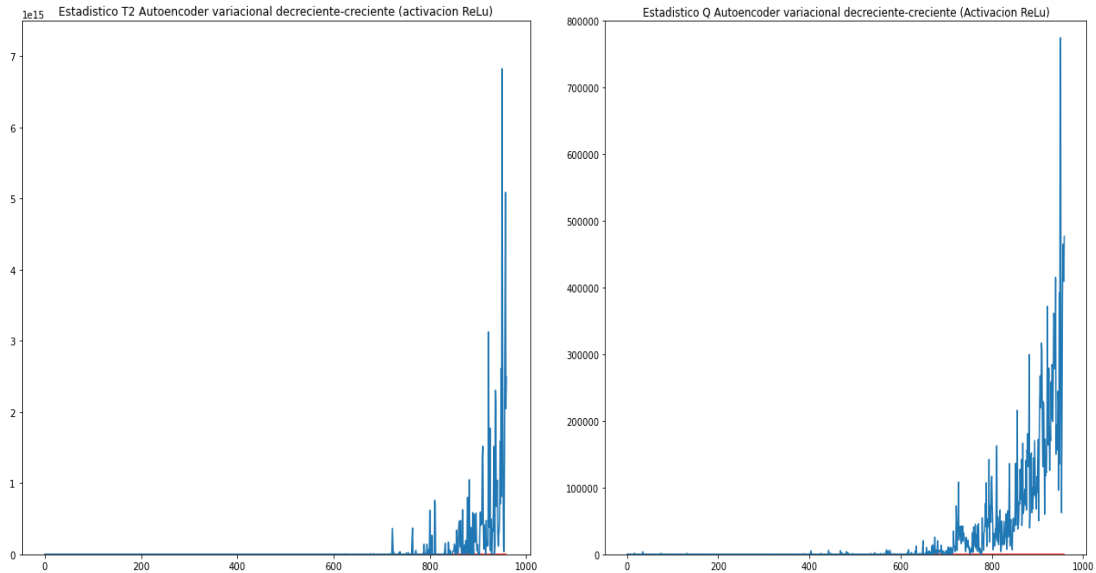


Ilustración 51 Detección del fallo 21 mediante Autoencoder Variacional decreciente-creciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 51 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 520 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 514 observaciones desde que ocurre el fallo en detectarlo.

A continuación, se representará la tabla de resultados (Tabla 12) de detección de fallos mediante el Autoencoder variacional decreciente-creciente con activación sigmoide.

En los datos de comportamiento normal tanto para el estadístico T^2 como para el estadístico Q hay un 1.25% de falsas alarmas.



Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	6	99.25	1.25	5	99.375	2.5
2	14	98.375	1.25	13	98.625	2.5
3	No detección del fallo	3.75	0.625	No detección del fallo	8.75	10.625
4	No detección del fallo	5.375	1.25	354	16.75	2.5
5	14	25.375	1.25	1	99.875	2.5
6	8	99.25	1.25	0	99.99	0
7	0	75.625	1.25	0	92.5	1.875
8	25	96.375	1.25	22	96.875	1.25
9	No detección del fallo	3.25	3.75	No detección del fallo	7.25	14.375
10	101	28.75	2.5	26	71.25	2.5
11	No detección del fallo	25	4.375	143	46.625	2.5
12	31	96.625	1.25	2	99.5	5.625
13	44	94.375	2.5	44	94.625	2.5
14	3	95.625	1.25	1	99.875	2.5
15	No detección del fallo	3.375	0	No detección del fallo	8.5	2.5
16	No detección del fallo	15.375	16.25	11	71.75	24.375
17	28	75.25	1.25	25	83.75	1.875
18	91	88.875	2.5	86	89.375	2.5
19	No detección del fallo	8.125	1.25	No detección del fallo	37.25	1.875
20	86	28.99	1.25	74	65.125	0.625
21	520	40.25	4.375	514	44.875	5.625

Tabla 12 Resultados de la detección de fallos mediante Autoencoder Variacional decreciente-creciente activación ReLu

Los resultados del Autoencoder variacional decreciente-creciente con activación ReLu son mucho peores que mediante la activación sigmoide. Mediante el estadístico T² se dejan de detectar los fallos 11 y 16 que, si se detectaban mediante el Autoencoder variacional con activación sigmoide,



además de que en la mayoría de fallos se tardan más en detectar. Mediante el estadístico Q se dejan de detectar los fallos 3, 9, 15 y 19 además de que el tiempo de detección en el resto de fallos es mucho más lento. La activación ReLu da muchos peores resultados que la activación sigmoide.

3.1.6 RNN VARIACIONAL CRECIENTE-DECRECIENTE

A continuación, se van a desarrollar dos redes neuronales variacionales crecientes-decrecientes: Uno con activación sigmoide y otro con activación ReLu.

Las dimensiones de estas redes neuronales variacionales creciente-decreciente será la misma que para la RNN simple 52-80-110-150-110-80-52 para así poder realizar la comparación entre ellos.

Antes de tener la red completa se irán entrenando distintas redes capa a capa antes de juntarlo una red completa de la manera:

- Red 1: Dimensiones 52-80-52
- Red 2: Dimensiones 80-110-80

A diferencia que la RNN simple y la RNN recurrente no se entrena una tercera red por separado para entrenar previamente la capa oculta, sino que la capa oculta se consigue directamente al unir las redes entrenadas por separado en la RNN variacional final.

Una vez entrenados estas redes por separado se juntan en la red neuronal completa con la estructura final 52-80-110-150-110-80-52.

Otros parámetros que se utilizan para la compilación y entrenamiento de esta red recurrente son:

- N.º de iteraciones (épocas) = 150
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM

3.1.6.1 ACTIVACION SIGMOIDE

En la Ilustración 52 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante la RNN Variacional creciente-decreciente con activación sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 165 (5 observaciones

después de que se produzca) y mediante el estadístico Q en la observación 162 (2 observaciones después de que se produzca el fallo).

Como se observa en la Ilustración 53, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

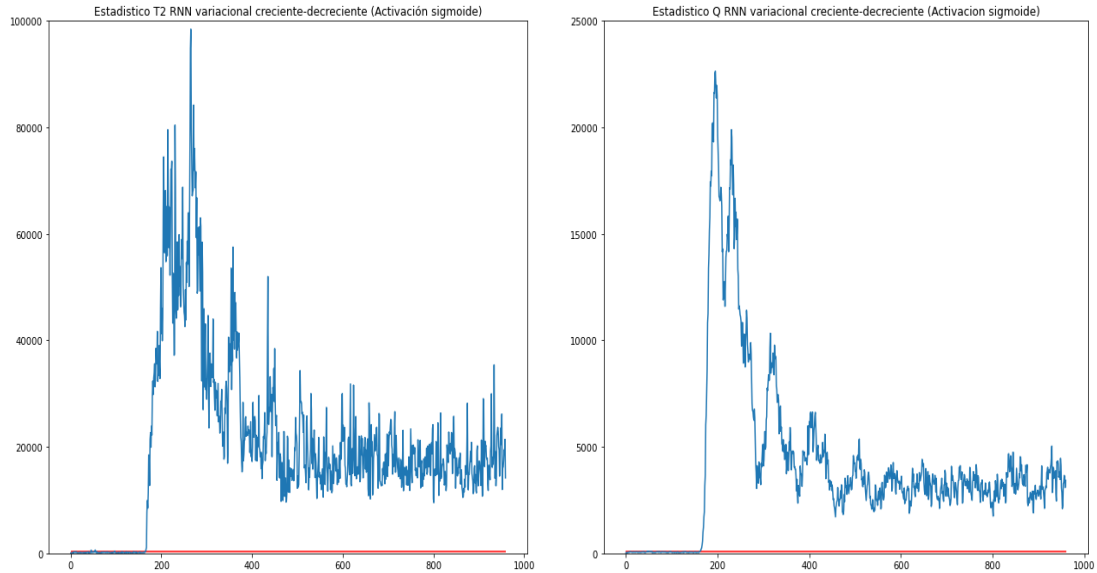


Ilustración 52 Detección del fallo 1 mediante RNN Variacional creciente-decreciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

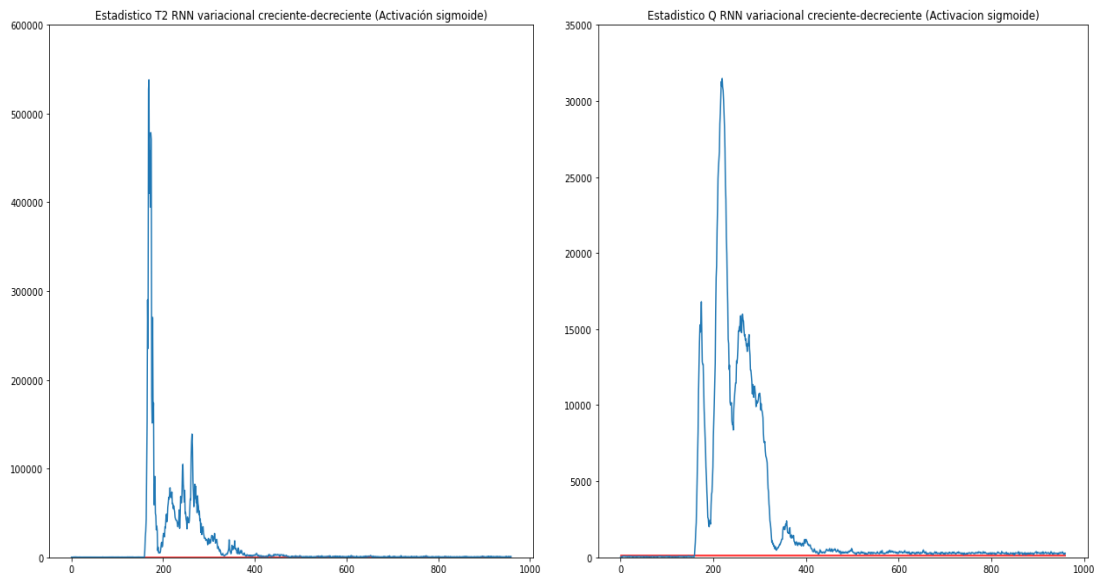


Ilustración 53 Detección del fallo 7 mediante RNN Variacional creciente-decreciente activación sigmoide. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 54 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 506 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 471 observaciones desde que ocurre el fallo en detectarlo.

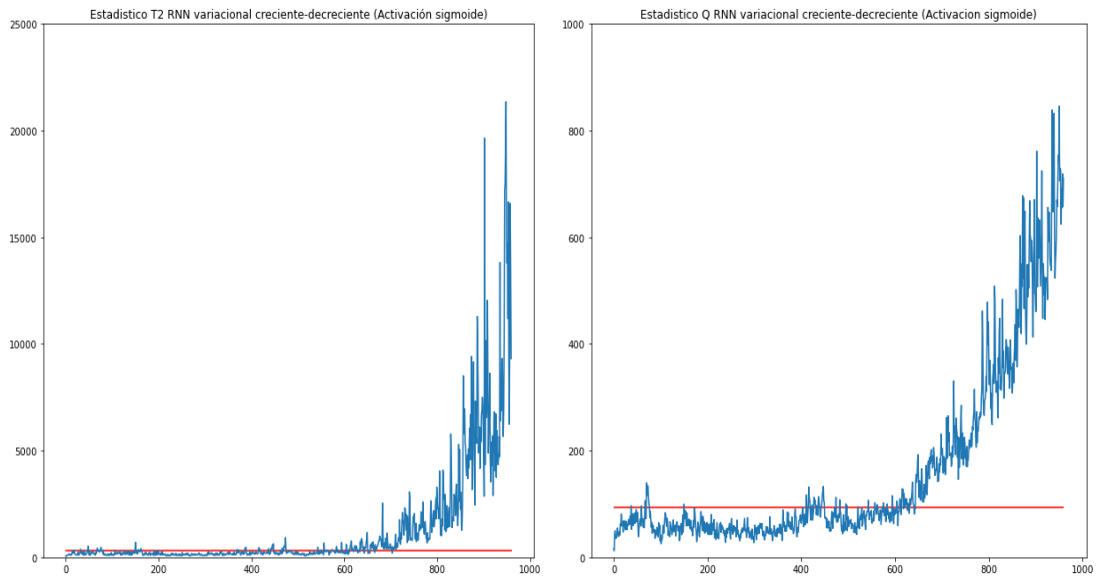


Ilustración 54 Detección del fallo 21 mediante RNN Variacional creciente-decreciente activación sigmoide. Izquierda, Detección mediante estadístico T^2 . Derecha, Detección mediante estadístico Q

A continuación, se representará la tabla de resultados (Tabla 13) de detección de fallos mediante la RNN variacional creciente-decreciente con activación sigmoide.

En los datos de comportamiento normal para el estadístico T^2 hay 1.875% de falsas alarmas y para el estadístico Q hay un 0.625% de falsas alarmas.

En comparación con la RNN simple creciente-decreciente, mediante el estadístico T^2 se consiguen mejorar los resultados con la RNN Variacional ya que se consigue detectar el fallo 9 y el fallo 19 que no se conseguían detectar mediante la RNN simple pero el tiempo de detección en el resto de fallos es muy similar. Mediante el estadístico Q siguen sin detectarse los fallos 3 y 9 y la mejoría es notable en la detección de algunos fallos como el fallo 15 que se consigue detectar 50 observaciones antes. Además, el número de falsas alarmas está dentro de valores normales excepto en algunos fallos, como son los fallos 3, 9, 12, 16 y 21.

Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	5	99.375	3.125	2	99	0.625
2	14	98.375	4.375	11	98.625	0.625
3	No detección del fallo	11	5.625	No detección del fallo	7.75	8.75
4	310	30.125	4.375	0	99.99	0.625
5	10	35.75	4.375	0	99.99	0.625
6	5	99.375	0.625	0	99.99	0
7	0	92.5	0	0	99.99	0
8	22	97.625	5	19	97.99	1.25
9	730	11.375	13.125	No detección del fallo	6.375	16.25
10	76	47.125	0.625	21	85.375	0
11	291	52.99	4.375	5	74.125	1.25
12	21	98.375	6.25	2	99.75	8.125
13	45	95.125	3.125	40	95.375	1.875
14	0	99.99	3.125	0	99.99	1.875
15	No detección del fallo	12.625	1.875	626	11.875	0.625
16	313	27.25	21.25	8	90.75	15.625
17	24	89.875	2.5	23	95.875	0.625
18	91	90.99	3.125	78	90.875	1.25
19	75	17.875	5.625	10	82.125	0.625
20	80	50.25	1.875	69	80.625	1.25
21	506	46.125	7.5	471	48.375	6.875

Tabla 13 Resultados de la detección de fallos mediante RNN Variacional creciente-decreciente activación sigmoide

3.1.6.2 ACTIVACION RELU

A continuación, se va a desarrollar otra RNN variacional realizado de la misma manera que la RNN anterior, pero utilizando como función de activación la función Relu para intentar conseguir mejores resultados.

En la Ilustración 55 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante la RNN Variacional creciente-decreciente con activación Relu en el que se detecta el fallo mediante el estadístico T² en la observación 168 (8 observaciones

después de que se produzca) y mediante el estadístico Q en la observación 164 (4 observaciones después de que se produzca el fallo).

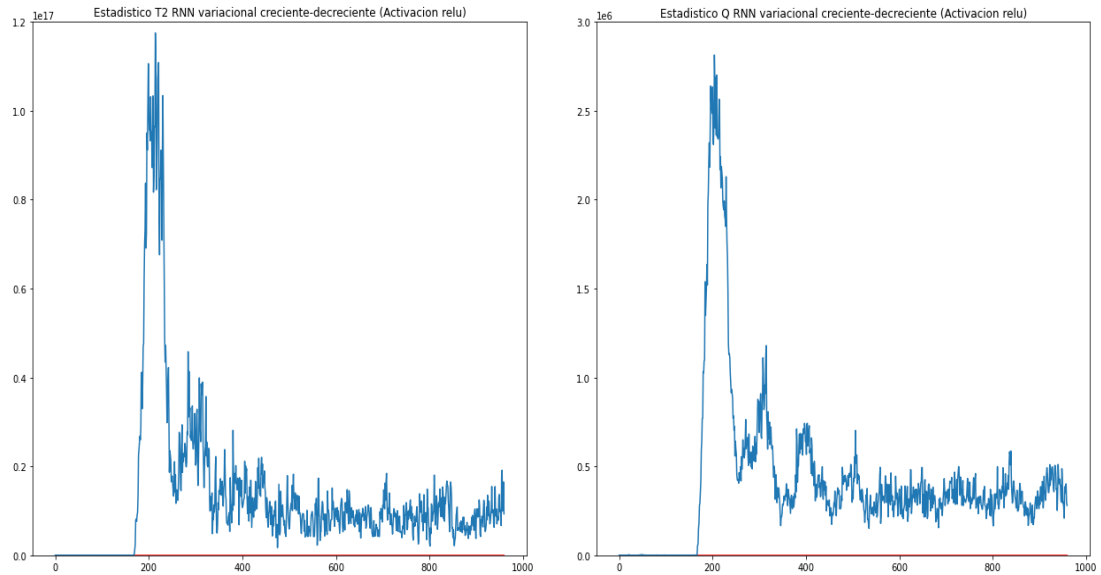


Ilustración 55 Detección del fallo 1 mediante RNN Variacional creciente-decreciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

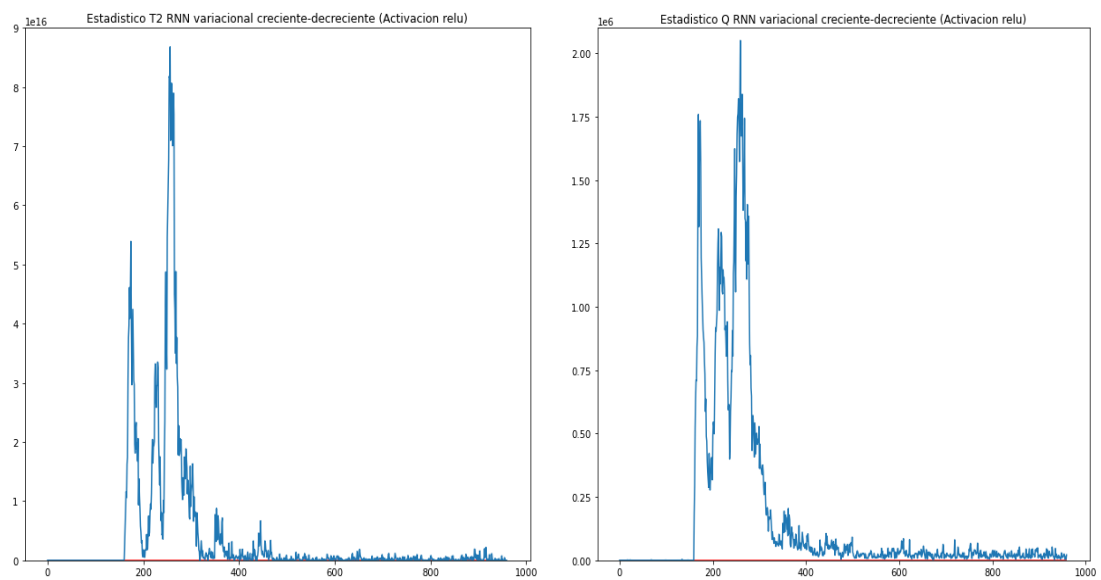


Ilustración 56 Detección del fallo 7 mediante RNN Variacional creciente-decreciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

Como se observa en la Ilustración 56, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

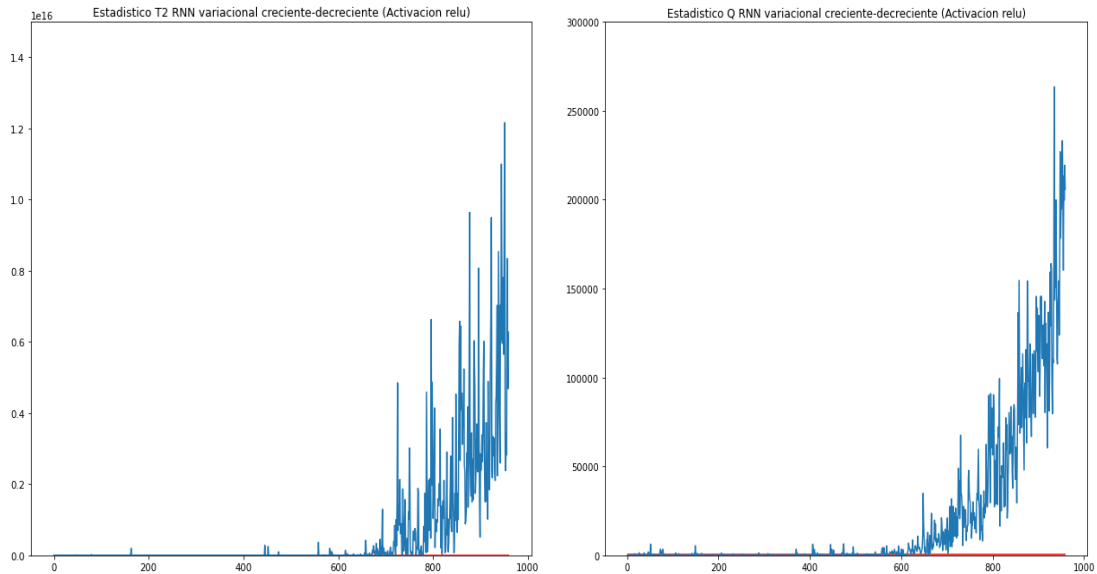


Ilustración 57 Detección del fallo 21 mediante RNN Variacional creciente-decreciente activación ReLu. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

En la Ilustración 57 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 528 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 415 observaciones desde que ocurre el fallo en detectarlo.

A continuación, se representará la tabla de resultados (Tabla 14) de detección de fallos mediante la RNN variacional creciente-decreciente con activación ReLu.

En los datos de comportamiento normal para el estadístico T^2 hay 1.25% de falsas alarmas y para el estadístico Q hay un 3.75% de falsas alarmas.

Los resultados de la RNN variacional creciente-decreciente con activación ReLu son mucho peores que mediante la activación sigmoide. Mediante el estadístico T^2 se dejan de detectar los fallos 4, 9 y 19 que, si se detectaban mediante la RNN variacional con activación sigmoide, además de que la mayoría de fallos se tardan más en detectar. Mediante el estadístico Q mejora ya que el fallo 9 se detecta en cuanto ocurre, aunque el tiempo de detección de los fallos es peor en la mayoría de los fallos. Además, el número de falsas alarmas aumenta considerablemente.



Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	8	99.125	3.125	4	99.625	13.125
2	27	97.875	1.875	12	98.5	6.875
3	No detección del fallo	5.875	4.375	No detección del fallo	17.75	11.875
4	No detección del fallo	22.625	0.625	55	62.99	8.75
5	12	25.25	0.625	0	99.99	8.75
6	4	99.875	1.875	0	99.99	5.625
7	0	98.875	1.875	0	99.99	6.25
8	25	92.99	2.5	15	98.25	9.375
9	No detección del fallo	5.125	7.5	0	18	20.625
10	534	26.99	1.25	46	68.875	8.125
11	383	31.625	0.625	18	69.25	10.625
12	32	93.625	1.25	1	98.99	13.75
13	47	94.375	0.625	36	95.99	4.375
14	3	95.125	3.75	0	99.99	6.25
15	No detección del fallo	4	1.875	642	19.75	8.125
16	623	15.5	5.625	10	69.5	29.375
17	28	76.99	1.25	24	89.75	10.625
18	87	89.75	3.75	86	92.375	15.625
19	No detección del fallo	7.625	4.375	71	40	10
20	86	29.5	1.875	74	69.875	4.375
21	528	39.125	6.25	415	54.99	18.125

Tabla 14 Resultados de la detección de fallos mediante RNN Variacional creciente-decreciente activación ReLu

3.1.7 AUTOENCODER VARIACIONAL RECURRENTE DECRECIENTE-CRECIENTE GRU

A continuación, se va a desarrollar un Autoencoder Variacional recurrente con ambas activaciones, tanto recurrente como la de propagación hacia adelante,



sigmoide ya que en los apartados anteriores se ha visto como para los Autoencoders variacionales funciona mejor la activación sigmoide.

La dimensión de este Autoencoder decreciente-creciente será la misma que para el Autoencoder simple 52-42-35-24-35-42-52 para así poder realizar la comparación entre ellos.

Antes de tener el Autoencoder completo se irán entrenando distintos Autoencoders capa a capa antes de juntarlo en el Autoencoder completo de la manera:

- Autoencoder 1: Dimensiones 52-42-52
- Autoencoder 2: Dimensiones 42-35-42

A diferencia que el Autoencoder simple no se entrena un tercer Autoencoder por separado para entrenar previamente la capa oculta, sino que la capa oculta se consigue directamente al unir los Autoencoders entrenados por separado en el Autoencoder variacional recurrente final.

Una vez entrenados estos Autoencoders por separado se juntan en el Autoencoder completo con la estructura final 52-42-35-24-35-42-52.

Otros parámetros que se utilizan para la compilación y entrenamiento del Autoencoder Variacional recurrente GRU son:

- N.º de iteraciones (épocas) = 500
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM

En la Ilustración 58 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante el Autoencoder Variacional recurrente decreciente-creciente con activaciones sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 166 (6 observaciones después de que se produzca) y mediante el estadístico Q en la observación 162 (2 observaciones después de que se produzca el fallo).

Como se observa en la Ilustración 59, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

En la Ilustración 60 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 564 observaciones desde que ocurre el fallo en

detectarlo y mediante el estadístico Q se tarda 459 observaciones desde que ocurre el fallo en detectarlo.

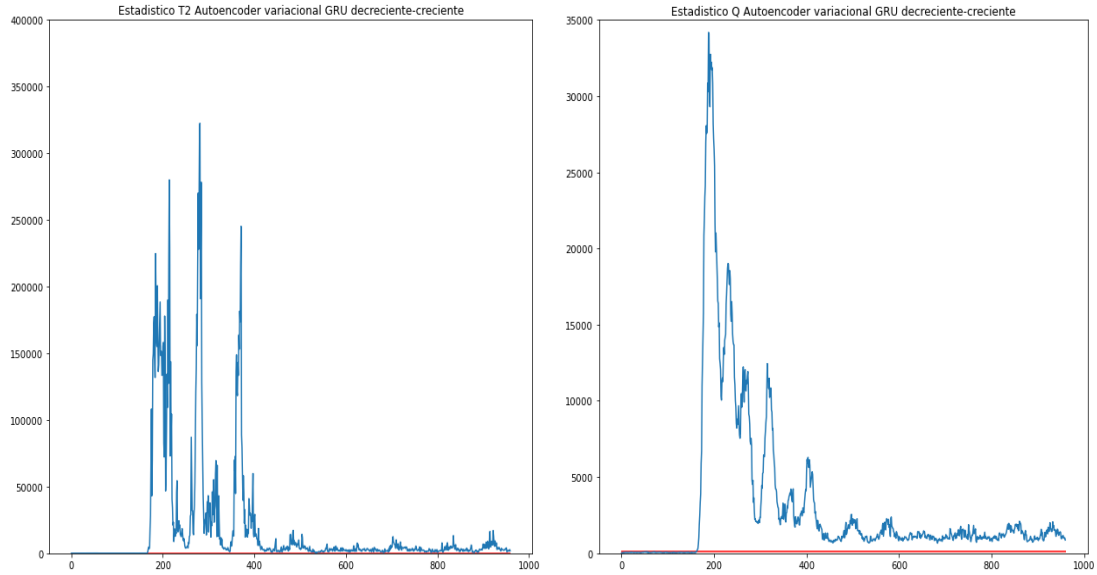


Ilustración 58 Detección del fallo 1 mediante Autoencoder Variacional decreciente-creciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

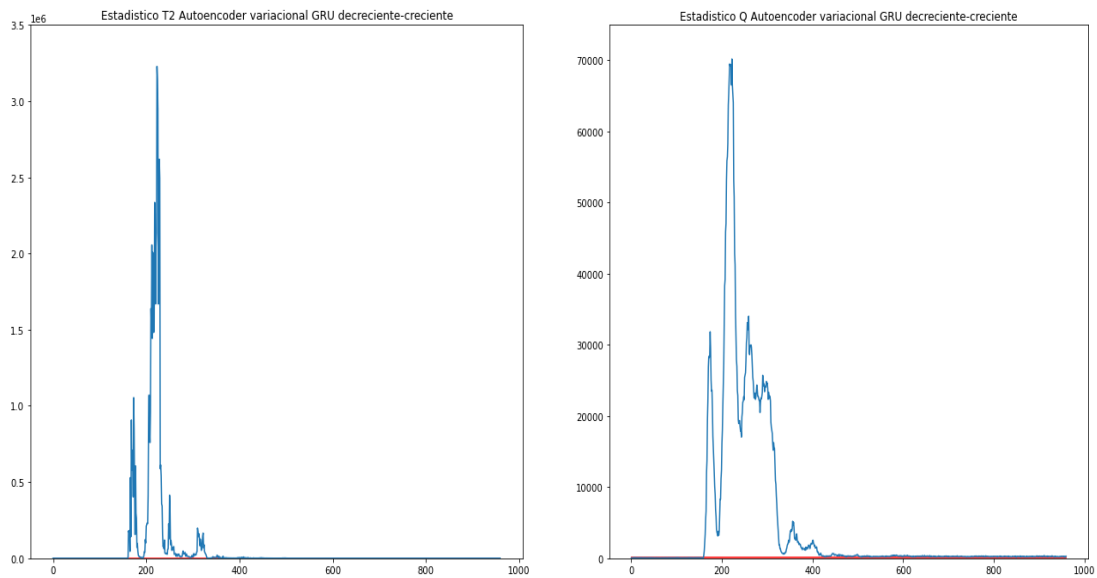


Ilustración 59 Detección del fallo 7 mediante Autoencoder Variacional decreciente-creciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

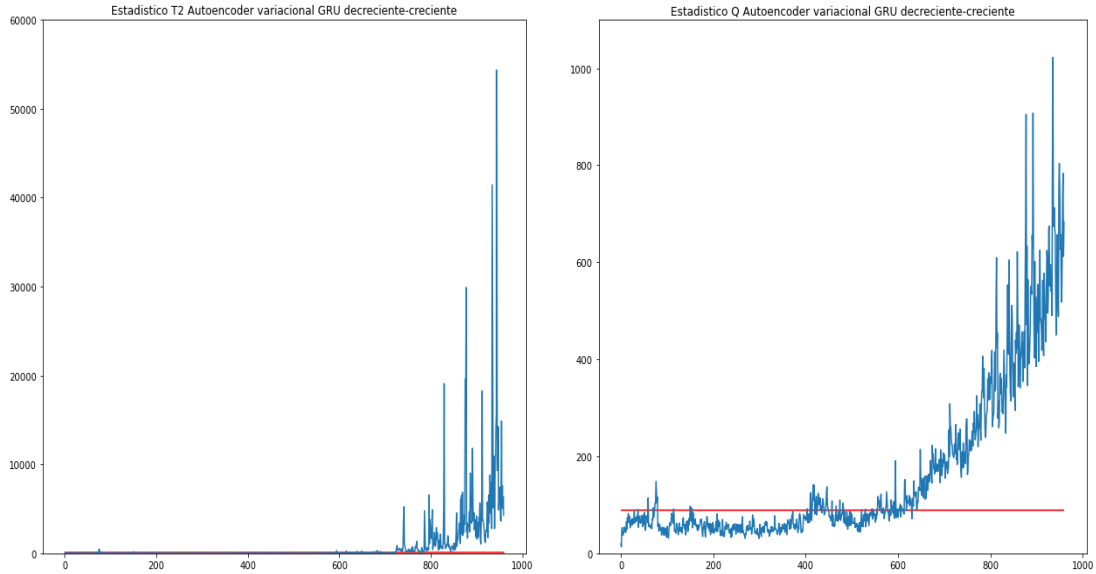


Ilustración 60 Detección del fallo 21 mediante Autoencoder Variacional decreciente-creciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

A continuación, se representará la tabla de resultados (Tabla 15) de detección de fallos mediante el Autoencoder variacional recurrente GRU decreciente-creciente con activaciones, tanto la de propagación hacia adelante como la recurrente, sigmoide.

En los datos de comportamiento normal para el estadístico T² hay 0% de falsas alarmas y para el estadístico Q hay un 0.625% de falsas alarmas.

Mediante este tipo de Autoencoder se puede realizar una comparación con los tres Autoencoders decreciente-creciente realizados anteriormente, el Autoencoder simple, Autoencoder recurrente GRU (con ambas activaciones sigmoide) y el Autoencoder variacional.

Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	6	99.25	0	2	99.75	1.25
2	14	98.25	0	12	98.5	0.625
3	No detección del fallo	3.875	3.125	396	7.25	4.375
4	No detección del fallo	13	0.625	0	99.99	1.875
5	12	25.125	0.625	0	99.99	1.875
6	6	99.25	0	0	99.99	0
7	0	59.5	0	0	99.99	0
8	22	96.375	0	19	98.5	0.625
9	No detección del fallo	3.125	7.5	No detección del fallo	7.875	18.125
10	35	39.875	0.625	24	86.375	0
11	301	30.875	0	6	76.99	1.875
12	21	96.75	1.25	2	99.75	5.625
13	45	93.75	0	37	95.5	0
14	12	94.375	0	0	99.99	1.25
15	No detección del fallo	5.625	0	629	13	0.625
16	416	19.25	15	7	92.25	20
17	25	84.625	0	21	96.125	0
18	91	88.75	0	83	90.625	1.25
19	No detección del fallo	1.5	0	9	85.99	0
20	80	33.375	0	69	84.125	1.25
21	564	31.125	4.375	459	51.875	8.125

Tabla 15 Resultados de la detección de fallos mediante Autoencoder Variacional recurrente GRU decreciente-creciente

En comparación con el Autoencoder simple, mediante el estadístico T² se consigue detectar el fallo 16 que no se conseguía detectar mediante el Autoencoder simple, los fallos 3, 4, 9, 15 y 19 siguen sin conseguir detectarse. Además, se reduce el tiempo de detección en todos los fallos de una manera notable. El número de falsas alarmas aumenta en baja medida. Mediante el estadístico Q se consigue detectar el fallo 3 que no se conseguía detectar mediante el Autoencoder simple, el fallo 9 sigue sin detectarse. El tiempo de detección para casi todos los fallos es muy similar, aunque se consigue reducir



el número de falsas alarmas. Por lo que se puede concluir que mejora al Autoencoder simple sobre todo respecto al estadístico T^2 .

En comparación con el Autoencoder recurrente GRU, mediante el estadístico T^2 siguen sin detectarse los fallos 3, 4, 9, 15 y 19 y los tiempos de detección del resto de fallos son ligeramente mejores, en contra posición el número de falsas alarmas aumenta ligeramente. Mediante el estadístico Q se consigue detectar el fallo 3 que no se conseguía detectar mediante el Autoencoder recurrente GRU. Además, los tiempos de detección del resto de fallos son ligeramente mejores y se reducen el número de falsas alarmas. Por lo que se puede concluir que mejora al Autoencoder recurrente GRU.

En comparación con el Autoencoder Variacional, mediante el estadístico T^2 siguen sin detectarse los fallos 3, 4, 9, 15 y 19. Además, hay fallos que se detectan más rápidamente como el fallo 10 que se detecta 62 observaciones antes o el fallo 16 que se detecta 202 observaciones antes, pero también hay fallos que tardan más en detectarse mediante este Autoencoder como es el fallo 21 que tarda en detectarse 59 observaciones más que para el Autoencoder variacional simple. Mediante el estadístico Q no se detecta el fallo 9 que si se consigue detectar mediante el Autoencoder variacional simple. Además, los fallos se detectan ligeramente más lentos. Se puede concluir que el Autoencoder variacional realiza mejor la detección de fallos que el Autoencoder variacional recurrente, aunque se consigan reducir el número de falsas alarmas mediante este último.

3.1.8 RNN VARIACIONAL RECURRENTE CRECIENTE-DECRECIENTE GRU

A continuación, se va a desarrollar una red neuronal Variacional recurrente con ambas activaciones, tanto recurrente como la de propagación hacia adelante, sigmoide ya que en los apartados anteriores se ha visto como para las redes neuronales variacionales y los Autoencoders variacionales funciona mejor la activación sigmoide.

La dimensión de esta red neuronal variacional recurrente creciente-decreciente será la misma que para la RNN simple 52-80-110-150-110-80-52 para así poder realizar la comparación entre ellos.

Antes de tener la red completa se irán entrenando distintas redes capa a capa antes de juntarlo en la red completa de la manera:

- Red 1: Dimensiones 52-80-52
- Red 2: Dimensiones 80-110-80



A diferencia que la RNN simple y la RNN recurrente no se entrena una tercera red por separado para entrenar previamente la capa oculta, sino que la capa oculta se consigue directamente al unir las redes entrenadas por separado en la RNN variacional recurrente final.

Una vez entrenados estas redes por separado se juntan en la red neuronal completa con la estructura final 52-80-110-150-110-80-52.

Otros parámetros que se utilizan para la compilación y entrenamiento de esta red variacional recurrente son:

- N.º de iteraciones (épocas) = 250
- Tamaño de batch (número de muestras que se propagan a través de la red) = 64
- Optimizador: ADAM

En la Ilustración 61 se observa cómo se detecta el fallo 1 (Relación de flujo de alimentaciones A/C con la composición de B constante) mediante la RNN Variacional recurrente creciente-decreciente con activaciones sigmoide en el que se detecta el fallo mediante el estadístico T^2 en la observación 165 (5 observaciones después de que se produzca) y mediante el estadístico Q en la observación 162 (2 observaciones después de que se produzca el fallo).

Como se observa en la Ilustración 62, el fallo 7 (Pérdida de presión en la corriente C) se detecta rápidamente al igual que mediante todas las técnicas anteriores ya que se detecta en cuanto ocurre.

En la Ilustración 63 se observa la detección del fallo 21 (Fallo desconocido) y lo tardía que es la detección mediante ambos estadísticos. Mediante el estadístico T^2 se tarda 494 observaciones desde que ocurre el fallo en detectarlo y mediante el estadístico Q se tarda 454 observaciones desde que ocurre el fallo en detectarlo.

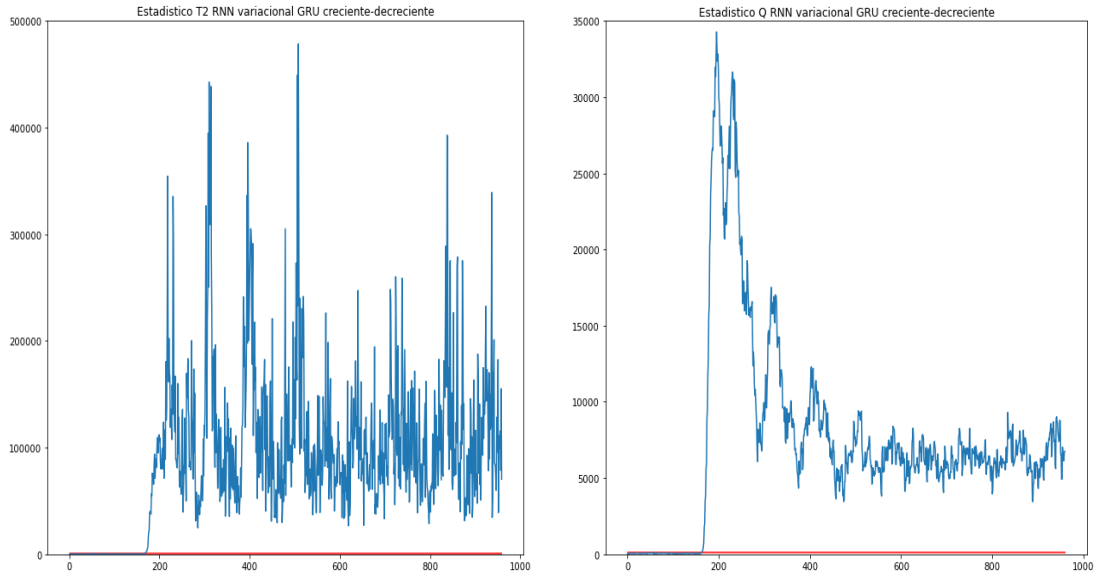


Ilustración 61 Detección del fallo 1 mediante RNN Variacional creciente-decreciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

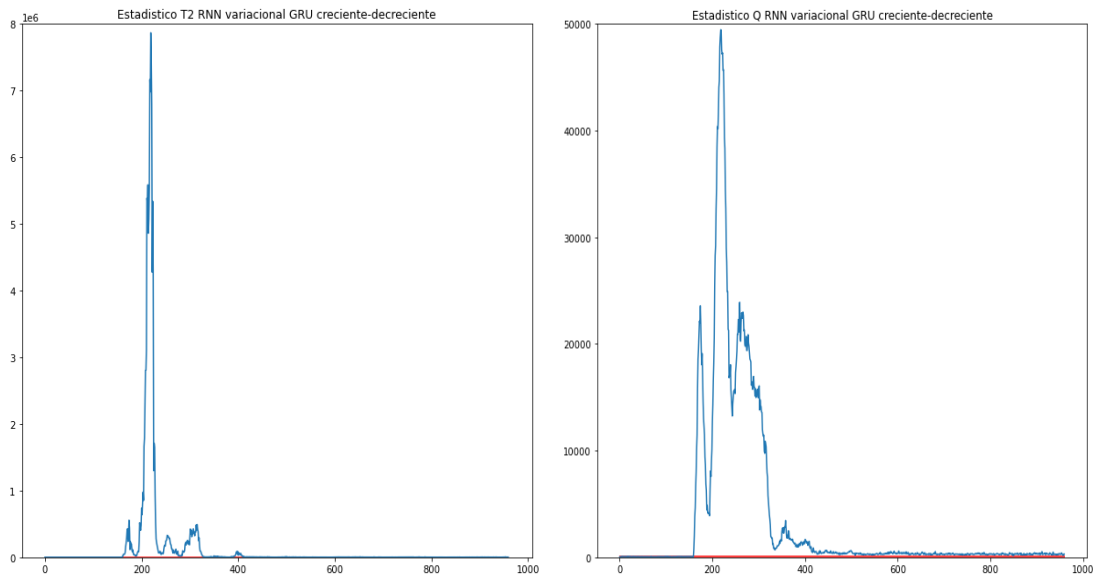


Ilustración 62 Detección del fallo 7 mediante RNN Variacional creciente-decreciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

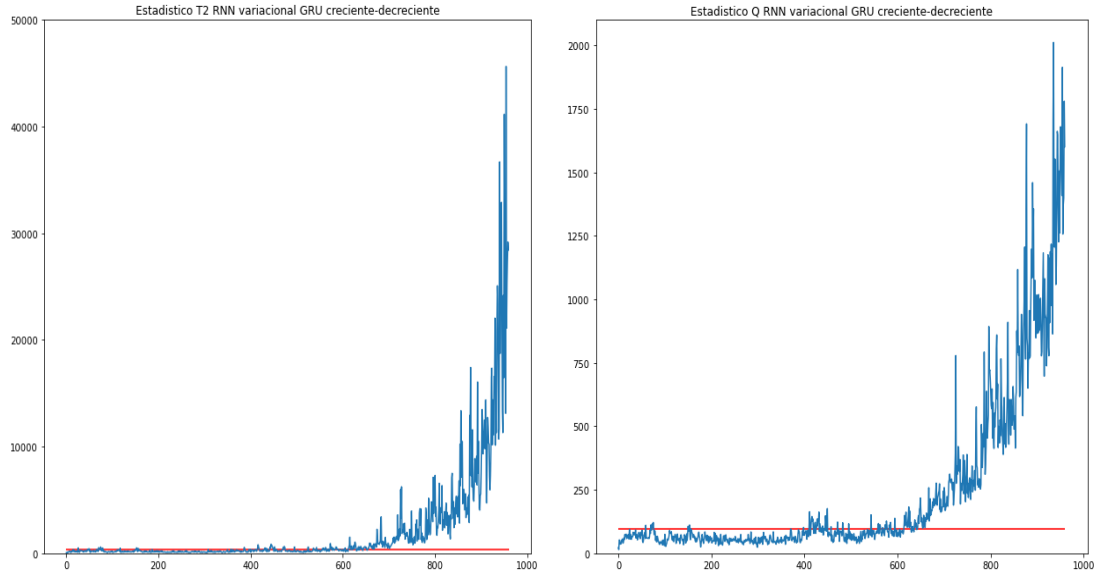


Ilustración 63 Detección del fallo 21 mediante RNN Variacional creciente-decreciente GRU ambas activaciones sigmoides. Izquierda, Detección mediante estadístico T2. Derecha, Detección mediante estadístico Q

A continuación, se representará la tabla de resultados (Tabla 16) de detección de fallos mediante la RNN variacional recurrente GRU creciente-decreciente con activaciones, tanto la de propagación hacia adelante como la recurrente, sigmoide.

En los datos de comportamiento normal para el estadístico T^2 hay 1.25% de falsas alarmas y para el estadístico Q hay un 0.625% de falsas alarmas.

Mediante este tipo de red neuronal se puede realizar una comparación con las tres RNN creciente-decreciente realizados anteriormente, la RNN simple, RNN recurrente GRU (con ambas activaciones sigmoide) y la RNN variacional.

Fallo	Tiempo de detección mediante T ²	FDR (% fallos detectados) con T ²	% Falsas alarmas previas al fallo con T ²	Tiempo de detección mediante Q	FDR (% fallos detectados) con Q	% Falsas alarmas previas al fallo con Q
1	5	99.375	3.125	2	99.75	2.5
2	11	98.625	0.625	11	98.75	4.375
3	No detección del fallo	10.5	8.75	No detección del fallo	10.375	8.75
4	70	37.99	3.125	0	99.99	1.25
5	10	31.625	3.125	0	99.99	1.25
6	5	99.375	1.25	0	99.99	0
7	0	99.75	0.625	0	99.99	0.625
8	22	97.375	4.375	18	98.125	1.25
9	No detección del fallo	8.625	13.75	228	8.375	18.75
10	77	47.5	0.625	18	85.99	1.25
11	191	57.125	1.875	6	76.25	1.875
12	2	99.125	5.625	2	99.75	9.375
13	35	95.75	3.125	40	95.25	1.875
14	0	99.99	2.5	0	99.99	1.875
15	No detección del fallo	8.625	1.875	626	16	1.25
16	195	30.625	23.125	9	89.375	25
17	23	93.125	3.125	21	96.375	1.875
18	87	90.375	4.375	83	90.75	2.5
19	No detección del fallo	15.25	2.5	10	82.875	1.875
20	80	52.5	1.25	66	81.375	2.5
21	494	50.5	10.625	454	48.75	6.25

Tabla 16 Resultados de la detección de fallos mediante RNN Variacional recurrente GRU creciente-decreciente

En comparación con la RNN simple, mediante el estadístico T² siguen sin detectarse los fallos 3, 9, 15 y 19 al igual que mediante la RNN simple. Además, se reduce el tiempo de detección en todos los fallos, en algunos de ellos de una manera notable, como por ejemplo el fallo 4 que pasa de detectarse a las 354 observaciones mediante la RNN simple a detectarse a las 70 observaciones mediante esta RNN. El número de falsas alarmas también se disminuye en comparación con la RNN simple. Mediante el estadístico Q se consigue detectar el fallo 9 que no se conseguía detectar mediante la RNN simple, el fallo 3 sigue sin detectarse. El tiempo de detección para casi todos los fallos es muy similar,



excepto los que se tardaban mucho en detectar mediante la RNN simple como son el fallo 15 y 21 que se reduce el tiempo de detección notablemente. El número de falsas alarmas en ambos RNN es muy similar. Por lo que se puede concluir que mejora a la RNN simple en el tiempo de detección de los fallos que tardan mucho en detectarse.

En comparación con la RNN recurrente GRU, mediante el estadístico T^2 se consiguen detectar los fallos 4 y 16 que no se conseguían detectar mediante la RNN simple. Los fallos 3, 9, 15 y 19 siguen sin detectarse. Además, los tiempos de detección del resto de fallos son mejores como por ejemplo en el fallo 11 que se consigue detectar 178 observaciones antes, además se reduce el número de falsas alarmas con respecto a la RNN recurrente. Mediante el estadístico Q se consigue detectar el fallo 9 que no se conseguía detectar mediante la RNN recurrente GRU. Además, los tiempos de detección del resto de fallos son ligeramente mejores. El número de falsas alarmas es muy similar para ambas RNN. Por lo que se puede concluir que mejora a la RNN recurrente GRU.

En comparación con la RNN Variacional, mediante el estadístico T^2 no se detectan los fallos 9 y 19 que si se conseguían detectar mediante la RNN variacional simple. Los fallos 3 y 15 siguen sin detectarse. EL tiempo de detección de los fallos es más rápido como, por ejemplo, el fallo 11 que se detecta 100 observaciones antes o el fallo 16 que se detecta 118 observaciones antes. Las falsas alarmas que se producen antes del fallo son ligeramente menores con esta RNN. Mediante el estadístico Q se detecta el fallo 9 que no se conseguía detectar mediante la RNN variacional simple. El fallo 3 sigue sin detectarse. Además, los tiempos de detección de los fallos es muy similar para ambas RNN. Las falsas alarmas que se producen antes del fallo son ligeramente mayores con esta RNN. Se puede concluir que la RNN variacional creciente- decreciente realiza mejor la detección de fallos que la RNN variacional recurrente creciente-decreciente, sobre todo mediante el estadístico T^2 .



CAPITULO V: CONCLUSIONES Y TRABAJO FUTURO





1. CONCLUSIONES

En este trabajo se han desarrollado distintas técnicas de detección de fallos para los fallos posibles que pueden ocurrir en un proceso basado en la planta Tennessee Eastman.

En primer lugar, se ha desarrollado una aplicación basada en el control estadístico de procesos como es el Análisis de Componentes Principales mediante el cual se ha buscado detectar los 21 fallos posibles de la planta. Los resultados han sido satisfactorios ya que se consiguen detectar 17 de los 21 fallos que ocurren en el proceso, es decir, un 81% de los fallos posibles. Hay fallos que se detectan en cuanto ocurren como el fallo 7 y el fallo 14 mediante el estadístico T^2 , y los fallos 4, 6 y 7 mediante el estadístico Q .

A continuación, se desarrollan redes de propagación hacia adelante y Autoencoders con el objetivo de conseguir una detección más rápida de los fallos y conseguir detectar los fallos que no se detectan mediante PCA.

Hay distintos criterios para concluir que red o Autoencoder realiza mejor la función de detección de fallos ya que hay redes que detectan más fallos, pero produciendo más falsas alarmas o redes que detectan menos fallos, pero detectando los demás fallos con más rapidez.

De todas las técnicas desarrolladas, los peores resultados los produce el Autoencoder variacional simple decreciente-creciente con activación ReLu ya que mediante el estadístico T^2 no se detectan los fallos 3, 4, 9, 11, 15, 16 y 19 y mediante el estadístico Q no se detectan los fallos 3, 9, 15 y 19.

1.1 TABLA COMPARATIVA DE RESULTADOS ESTADISTICO T^2

A continuación, se va a presentar una tabla comparativa del tiempo de detección y tasa de detección de cada fallo mediante el estadístico T^2 para las redes con los mejores resultados, comparándolo con PCA.



Fallo	Tiempo detección PCA	Tiempo detección RNN simple creciente- decreciente	Tiempo detección RNN creciente- decreciente GRU (tanh- sigmoid)	Tiempo detección RNN variacional creciente- decreciente	Tiempo detección RNN variacional creciente- decreciente GRU
1	6	6	4	5	5
2	14	11	11	14	11
3	No detección del fallo	No detección del fallo	No detección del fallo	No detección del fallo	No detección del fallo
4	63	354	55	310	70
5	10	6	0	10	10
6	6	9	4	5	5
7	0	0	0	0	0
8	22	22	17	22	22
9	No detección del fallo	No detección del fallo	No detección del fallo	730	No detección del fallo
10	96	145	14	76	77
11	187	188	46	291	191
12	6	21	6	21	2
13	44	43	40	45	35
14	0	0	0	0	0
15	No detección del fallo	No detección del fallo	577	No detección del fallo	No detección del fallo
16	196	310	195	313	195
17	26	8	24	24	23
18	87	87	87	91	87
19	No detección del fallo	No detección del fallo	74	75	No detección del fallo
20	86	84	80	80	80
21	514	510	484	506	494
MEDIA	80	106	90	137	77
MEDIA SIN LOS FALLOS 3,9 Y 15	80	106	63	104	77

Tabla 17 Tabla comparativa de resultados tiempo de detección mediante Estadístico T²



En la Tabla 17 se ha realizado la comparación con PCA de las redes que no empeoran los resultados de PCA mediante el estadístico T^2 , se llega a la conclusión de que únicamente algunas de las redes neuronales desarrolladas con el encoder creciente y el decoder decreciente igualan o mejoran los resultados de PCA en cuanto a la detección de fallos, los Autoencoders con el encoder decreciente y el decoder creciente desarrollados empeoran los resultados de PCA.

Como se observa en la Tabla 17 los mejores resultados se obtienen mediante la RNN recurrente GRU con activación tangente hiperbólica y la activación recurrente sigmoide ya que se detectan todos los fallos excepto los fallos 3 y 9 y mediante la RNN variacional con activación sigmoide ya que se detectan todos los fallos excepto los fallos 3 y 15.

La RNN recurrente también es la que detecta más rápido los fallos en la mayoría de los fallos que se producen, aunque también aumenta el número de falsas alarmas.

Otro criterio que se ha dispuesto en la Tabla 17 es la media del tiempo de detección de los fallos que se detectan cuantificando los fallos 3, 9 y 15 y sin cuantificar dichos fallos, en el que se comprueba que el único que mejora a PCA es la RNN variacional creciente-decreciente GRU que mejora en 3 observaciones de media el tiempo de detección de los fallos si cuantificamos estos fallos. Si no cuantificamos estos fallos la RNN recurrente GRU creciente-decreciente mejora en 17 observaciones de media el tiempo de detección de los fallos.

A continuación, también se va a realizar una tabla comparativa de la tasa de detección de las mismas redes de la tabla anterior.

Como se observa en la Tabla 18 los mejores resultados se obtienen mediante la RNN recurrente GRU con activación tangente hiperbólica y la activación recurrente sigmoide ya que es con el que se obtiene la mayor tasa de detección de los fallos con una media de un 70.43% de los fallos. Además, también es el que mayor tasa de detección tiene desechando los fallos 3,9 y 15. Se comprueba también como se produce que la RNN variacional creciente-decreciente detecta todos los fallos excepto el 3 y el 9 pero es el que menos tasa de detección de fallos tiene, incluso siendo menor que PCA si desechamos los fallos 3,9 y 15.

Se representa haciendo las dos medias como los fallos 3, 9 y 15 condicionan mucho los resultados.



Fallo	% Tasa de detección PCA	% Tasa de detección RNN simple creciente-decreciente	% Tasa de detección RNN creciente-decreciente GRU (tanh-sigmoid)	% Tasa de detección RNN variacional creciente-decreciente	% Tasa de detección RNN variacional creciente-decreciente GRU
1	99.375%	99.5%	99.5%	99.375%	99.375%
2	98.375%	98.875%	98.625%	98.375%	98.625%
3	4.25%	17.875 %	19.125%	11%	10.5%
4	59.75%	46%	82.125%	30.125%	37.99%
5	28.75%	39%	40.375%	35.75%	31.625%
6	99.25%	99.5 %	99.625%	99.375%	99.375%
7	99.99%	99.99%	99.99%	92.5%	99.75%
8	97.5%	98.125%	98.5%	97.625%	97.375%
9	5.375%	18.25%	18.5%	11.375%	8.625%
10	47.5%	44.375%	58.99%	47.125%	47.5%
11	57.25%	62.5%	73.25%	52.99%	57.125%
12	99%	98.625%	99.25%	98.375%	99.125%
13	95.375%	95.625%	95.5%	95.125%	95.75%
14	99.99%	99.75%	99.99%	99.99%	99.99%
15	7.875%	19.25%	19.5%	12.625%	8.625%
16	28.75%	31.875%	46.99%	27.25%	30.625%
17	82.5%	84.875%	93.75%	89.875%	93.125%
18	89.875%	91.625%	91.75%	90.99%	90.375%
19	13.5%	35.875%	32.875%	17.875%	15.25%
20	43.75%	48.5%	58.625%	50.25%	52.5%
21	39.875%	49.5%	52.375%	46.125%	50.5%
MEDIA	61.8%	65.69%	70.43%	62.09%	63.03%
MEDIA SIN LOS FALLOS 3,9 Y 15	71.13%	73.56%	79%	70.5%	72%

Tabla 18 Tabla comparativa de resultados tasa de detección mediante Estadístico T²

1.2 TABLA COMPARATIVA DE RESULTADOS ESTADISTICO Q

A continuación, se va a presentar una tabla comparativa del tiempo de detección y tasa de detección de cada fallo mediante el estadístico Q para las redes con los mejores resultados, comparándolo con PCA.

Mediante el estadístico Q todas las técnicas de aprendizaje automático desarrolladas, como mínimo, igualan a la técnica de PCA. En las tablas 19 y 20



se ha representado las que mejor funcionan y más mejoran a PCA, observando como algunas detectan todos los fallos.

Hay dos redes neuronales con el que se consiguen detectar todos los fallos mediante el estadístico Q, que son el Autoencoder recurrente GRU decreciente-creciente con activación de propagación hacia adelante tangente hiperbólica y activación recurrente sigmoide y el Autoencoder variacional simple decreciente-creciente con activación sigmoide. Comparando estos dos Autoencoders, se consiguen mejores resultados respecto al criterio del tiempo de detección con el Autoencoder recurrente, pero se produce un número de falsas alarmas mucho mayor que mediante el Autoencoder Variacional simple decreciente-creciente.

Otro criterio mostrado en las tablas es la media tanto del tiempo de detección como de la tasa de detección, en primer lugar, teniendo en cuenta el tiempo de detección, PCA es el que mejor tiempo de detección medio de los fallos tiene respecto a las técnicas de aprendizaje automático mostradas en la tabla 19. Aunque, desechando los fallos 3, 9 y 15 la detección media más rápida se produce mediante el Autoencoder variacional decreciente-creciente que es 29 observaciones más rápida en el tiempo de detección medio que PCA y se comprueba como todas las técnicas de aprendizaje automático mostradas en la tabla 19 son considerablemente más rápidas que PCA mediante el estadístico Q.

Teniendo en cuenta la tasa de detección media, todas las técnicas de aprendizaje automático mostradas en la tabla 20 detectan más los fallos que PCA, siendo el mejor el Autoencoder variacional decreciente-creciente con una tasa de detección media del 81.09%. Si desechamos los fallos 3, 9 y 15 todas las técnicas de aprendizaje automático mostradas en la tabla 20 superan el 90% de tasa de detección media.

Se representa haciendo las dos medias como los fallos 3, 9 y 15 condicionan mucho los resultados.



Fallo	Tiempo detección PCA	Tiempo detección Autoencoder decreciente-creciente GRU (tanh-sigmoid)	Tiempo detección RNN creciente-decreciente GRU (tanh-sigmoid)	Tiempo detección Autoencoder variacional decreciente-creciente	Tiempo detección RNN variacional creciente-decreciente GRU
1	2	0	3	2	2
2	10	10	10	11	11
3	No detección del fallo	82	354	394	No detección del fallo
4	0	0	2	0	0
5	2	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	17	19	13	15	18
9	No detección del fallo	2	No detección del fallo	227	228
10	98	13	15	20	18
11	55	10	9	5	6
12	22	2	2	2	2
13	41	36	36	35	40
14	1	0	0	0	0
15	No detección del fallo	630	630	626	626
16	192	9	10	9	9
17	21	23	23	21	21
18	83	78	83	8	83
19	No detección del fallo	10	59	9	10
20	80	69	71	69	66
21	482	453	484	454	454
MEDIA	65	68	90	90	79
MEDIA SIN LOS FALLOS 3,9 Y 15	65	41	45	36	41

Tabla 19 Tabla comparativa de resultados tiempo de detección mediante Estadístico Q



Fallo	%Tasa detección PCA	%Tasa detección Autoencoder decreciente- creciente GRU (tanh- sigmoid)	%Tasa detección RNN creciente- decreciente GRU (tanh- sigmoid)	%Tasa detección Autoencoder variacional decreciente- creciente	%Tasa detección RNN variacional creciente- decreciente GRU
1	99.75%	99.99%	99.625%	99.75%	99.75%
2	98.875%	98.875%	98.875%	98.625%	98.75%
3	3%	15.625%	14.75%	10.75%	10.375%
4	99.625%	99.875%	99.125%	99.99%	99.99%
5	26.5%	99.99%	70.625%	99.99%	99.99%
6	99.99%	99.99%	99.99%	99.99%	99.99%
7	99.99%	99.99%	99.99%	99.99%	99.99%
8	95.5%	98.375%	98.375%	98.375%	98.125%
9	3.75%	14.5%	12.75%	11.25%	8.375%
10	48%	85.375%	81.375%	85.375%	85.99%
11	54.875%	76.5%	76.875%	78.25%	76.25%
12	96.5%	99.625%	99.625%	99.75%	99.75%
13	95.125%	95.625%	95.625%	95.625%	95.25%
14	96.625%	99.99%	99.99%	99.99%	99.99%
15	6.25%	22.25%	19%	16.5%	16%
16	44.5%	89.25%	83.875%	92.25%	89.375%
17	95.75%	95.75%	94.625%	96.5%	96.375%
18	90.25%	91.625%	91.5%	91.125%	90.75%
19	16.25%	83.375%	68.99%	88.5%	82.875%
20	55.25%	80.99%	76.875%	87.125%	81.375%
21	52%	52.125%	47.75%	53.375%	48.75%
MEDIA	65.63%	80.93%	77.62%	81.09%	79.90%
MEDIA SIN FALLOS 3,9 Y 15	75.85%	91.51%	92.29%	92.47%	91.29%

Tabla 20 Tabla comparativa de resultados tasa de detección mediante Estadístico Q

2. TRABAJO FUTURO

Como trabajo futuro hay que buscar mejorar los resultados obtenidos en este trabajo. Para ello se pueden realizar técnicas utilizadas en este trabajo como los Autoencoders simples, pero con distintas funciones de activación que no sean la sigmoide.



Universidad de Valladolid

También se pueden utilizar distintos Autoencoders recurrentes, como, por ejemplo, redes LSTM para intentar conseguir mejores resultados o, en las redes recurrentes GRU, combinar distintas funciones de activación en la función de propagación hacia adelante y en la función recurrente.

También se pueden utilizar redes convolucionales para comprobar si produce mejores resultados en la detección de fallos.

Por último, como en este trabajo se han combinado las redes variacionales y las redes recurrentes, se pueden buscar distintas combinaciones de redes neuronales para conseguir mejores resultados.



BIBLIOGRAFÍA

- [1] Pablo Orellana Nirian, Economipedia, Control de calidad <https://economipedia.com/definiciones/control-de-calidad.html>
[Último acceso: 26 Mar. 2022]
- [2] ISOTools Excellence, Industria 4.0 <https://www.isotools.org/2018/07/12/industria-4-0-que-debemos-saber/#:~:text=El%20concepto%20de%20industria%204.0,%2C%20ci%20berindustria%2C%20industria%20inteligente%20etc> [Último acceso: 26 Mar. 2022]
- [3] IBM, Industria 4.0 <https://www.ibm.com/es-es/topics/industry-4-0>
[Último acceso: 26 Mar. 2022]
- [4] Universidad Complutense de Madrid, Gestión de la calidad <https://webs.ucm.es/centros/cont/descargas/documento10123.pdf>
[Último acceso: 28 Mar. 2022]
- [5] A. Sánchez Fernández, (2020), “Métodos de detección y diagnóstico de fallos mediante aproximaciones distribuidas: modelos, métodos y computación. Tesis doctoral” EsDUVa, Universidad de Valladolid <https://uvadoc.uva.es/bitstream/handle/10324/47723/TESIS-1843-210720.pdf?sequence=1&isAllowed=y>
- [6] Measure control, Historia del SPC <https://measurecontrol.com/la-historia-del-spc/> [Último acceso: 29 Mar. 2022]
- [7] R. Carro Paz y D. González Gómez, “Control Estadístico de Procesos”, Universidad Nacional de Mar del Plata http://nulan.mdp.edu.ar/1617/1/12_control_estadistico.pdf [Último acceso: 29 Mar. 2022]
- [8] M. González Velázquez, (2020), “Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos. Trabajo Fin de Grado en Ingeniería en Organización Industrial”, EII, Universidad de Valladolid
- [9] Universidad Carlos III, Gráficos de control con memoria <http://halweb.uc3m.es/esp/personal/personas/kaiser/esp/calidad/capitulo6.pdf> [Último acceso: 31 Mar. 2022]



- [10] A. Pequeño Alonso, (2020), “Mejora del control de calidad de un proceso mediante técnicas de aprendizaje automático. Trabajo Fin de Grado en Ingeniería Química”, EII, Universidad de Valladolid
- [11] M.J. De la Fuente Aparicio, “Control Estadístico de Procesos”, Universidad de Valladolid, Ingeniería de Sistemas y Automática
- [12] Universidad Autónoma Agraria Antonio Narro de México, Tablas de distribución F <http://www.uaaan.mx/~jmelbos/tablas/distf.pdf> [Último acceso: 1 abr. 2022]
- [13] A. González, cleverdata <https://cleverdata.io/que-es-machine-learning-big-data/> [Último acceso: 1 abr. 2022]
- [14] <http://blog.bismart.com/diferencias-machine-learning-supervisado-no-supervisado> [Último acceso: 1 abr. 2022]
- [15] Revista Ingeniería y tecnología, (2021), Unlr <https://www.unir.net/ingenieria/revista/redes-neuronales-artificiales/> [Último acceso: 1 abr. 2022]
- [16] https://ml4a.github.io/ml4a/es/neural_networks/#:~:text=De%20clasificadores%20lineales%20a%20neuronas&text=Podemos%20interpretar%20que%20cada%20peso.0%20independiente%20de%20los%20pesos [Último acceso: 1 abr. 2022]
- [17] <https://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo4.html> [Último acceso: 1 abr. 2022]
- [18] D. Calvo, (2018), Perceptrón Multicapa – Red Neuronal <https://www.diegocalvo.es/perceptron-multicapa/> [Último acceso: 1 abr. 2022]
- [19] J.A Villamil Torres y J.A Delgado Rivera, (2007), “Entrenamiento de una red neuronal multicapa para la tasa de cambio euro-dólar”, Ingeniería e investigación vol. 27 n° 3. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-56092007000300012#:~:text=El%20entrenamiento%20de%20una%20red.el%20conjunto%20de%20datos%20de



- [20] C. Arana, (2021), “Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales”, Universidad del CEMA (Buenos Aires) <https://ucema.edu.ar/publicaciones/download/documentos/797.pdf>
- [21] J. Torres, (2019), “Deep Learning – Introducción práctica con Keras (2º parte, Capítulo 7), Editorial Marcombo <https://torres.ai/redes-neuronales-recurrentes/>
- [22] <https://programmerclick.com/article/3394701573/> [Último acceso 4 abr. 2022]
- [23] C. Junyoung, C. Gulcehre, C. KyungHyun, Y. Bengio, (2014), “Empirical evaluation of gated recurrent neural networks on sequence modeling”, NIPS 2014 Workshop on Deep Learning
- [24] A. Dertat, (2017), “Applied Deep Learning – Part 3: Autoencoders” Towards Data Science <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>
- [25] Li. Z, Tian. L, Jiang. Q, Yan. X, (2021), “Distributed ensemble stacked autoencoder model for non-lineal process monitoring”, Information Sciences, 542, pp. 302-316
- [26] F. Sancho Caparrini, (2020), “Variational Autoencoder”, Departamento de Ciencias de la computación e Inteligencia Artificial, Escuela Técnica Superior de Informática, Universidad de Sevilla <http://www.cs.us.es/~fsancho/?e=232> [Último acceso: 4 abr. 2022]
- [27] Lee. S, Kwak. M, Tsui. K, Bum Kim. S, (2019), “Process Monitoring using variational autoencoder for high-dimensional nonlinear process”, Engineering Applications of Artificial Intelligence, 83, pp. 13-27
- [28] <https://keras.io/api/layers/activations/#sigmoid-function> [Último acceso: 7 abr. 2022]