

Detección de eventos anómalos en un entorno industrial mediante el uso de técnicas de Federated Learning.

**Darío Martín García Carretero**

Máster Universitario en Ciencia de Datos

Área 2

**Raúl Parada Medina**

**Jordi Casas Roma**

08/01/2020

Creative Commons License

This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).

**FICHA DEL TRABAJO FINAL**

|  |  |
| --- | --- |
| **Título del trabajo:** | *Detección de eventos anómalos en un entorno industrial mediante el uso técnicas de Federated Learning.* |
| **Nombre del autor:** | *Darío Martín García Carretero* |
| **Nombre del consultor/a:** | *Raúl Parada Medina* |
| **Nombre del PRA:** | *Jordi Casas Roma* |
| **Fecha de entrega (mm/aaaa):** | 01/2020 |
| **Titulación:** | *Máster en Ciencia de Datos* |
| **Área del Trabajo Final:** | *Área 2* |
| **Idioma del trabajo:** | *Español* |
| **Palabras clave** | *eventos anómalos, entorno industrial, federated learning* |
| **Resumen del Trabajo (máximo 250 palabras):** *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.* | |
| Un evento anómalo es aquel que se produce de forma repentina y sin previsión. En un entorno industrial estos eventos (generalmente debidos a algún error) pueden provocar grandes daños por lo que su detección puede ayudar a prevenir situaciones críticas.  El objetivo del proyecto es detectar las posibles anomalías que se pudieran producir en los equipos industriales mediante el uso de un modelo de aprendizaje automático. Para enseñar a distinguir a un modelo entre un comportamiento normal y otro anormal se necesitan datos, contra más datos, mejor.  Hoy en día la mayoría de los componentes dentro de un entorno industrial están constantemente monitorizados mediante el uso de dispositivos de medición especializados. Debido a esto, dispondremos de los datos suministrados por nuestros dispositivos de medición para entrenar al modelo, pero ¿y si pudiéramos disponer de más datos? Muchos equipos industriales son de uso genérico y pueden ser usados para muchas tareas en muchas compañías diferentes. Si pudiéramos tener acceso a los datos de todos esos dispositivos podríamos crear un modelo mucho mejor. Por diversas razones a las compañías no les gusta compartir sus datos, es por este motivo por lo que en este trabajo se propone el uso del Federated Learning (FL). Con el FL se pueden construir modelos aprovechando toda la información disponible manteniendo la privacidad de los datos. | |
| **Abstract (in English, 250 words or less):** | |
| An anomalous event is one that occurs suddenly and without foresight. In an industrial environment, these events (normally due to some error) can cause great damage, so their detection can help to prevent critical situations.  The objective of the project is to detect possible anomalies that will occur in industrial equipment using a machine learning model. To teach model how to distinguish between a normal and an abnormal behaviour, data is needed, the more data, the better.  Today, most components within an industrial environment are constantly monitored using specialized measuring devices. Because of this, we will have the data provided by our measuring devices for the model, but what if we could have more data? Many industrial equipment is of generic use and can be used for many tasks in many different companies. If we could have access to the data of all those devices, we could create a much better model. For various reasons, companies do not like to share their data, which is why this work proposes the use of Federated Learning (FL). With FL you can build models taking advantage of all the available information and you can keep the privacy of the data. | |

**Índice**

[1. Introducción 1](#_Toc27392476)

[1.1 Contexto y justificación del Trabajo 1](#_Toc27392477)

[1.2 Objetivos del Trabajo 2](#_Toc27392478)

[1.3 Enfoque y método seguido 2](#_Toc27392479)

[1.4 Planificación del Trabajo 3](#_Toc27392480)

[1.5 Breve sumario de productos obtenidos 4](#_Toc27392481)

[1.6 Breve descripción de los otros capítulos de la memoria 4](#_Toc27392482)

[2. Estudio del estado del arte 5](#_Toc27392483)

[2.1 Detección de eventos anómalos 5](#_Toc27392484)

[2.2 Federated Learning 10](#_Toc27392485)

[2.3 Novedades propuestas 11](#_Toc27392486)

[3. Diseño e implementación del trabajo 12](#_Toc27392487)

[3.1 Consideraciones previas 12](#_Toc27392488)

[3.2 Preparación del entorno de trabajo 12](#_Toc27392489)

[3.3 Implementación del software de simulación 13](#_Toc27392490)

[3.3.3 Introducción 13](#_Toc27392491)

[4. Conclusiones 14](#_Toc27392492)

[4. Glosario 15](#_Toc27392493)

[5. Bibliografía 16](#_Toc27392494)

[6. Anexos 17](#_Toc27392495)

**Lista de figuras**

[Fig. 1 Los fallos no controlados pueden tener resultados catastróficos 1](file:///D:\projects\uoc\masterdatacience\doc\TFM_PEC3_v0.docx#_Toc27401332)

[Fig. 2 IRESE framework 5](#_Toc27401333)

[Fig. 3 Arquitectura de la red propuesta 6](#_Toc27401334)

[Fig. 4 Proyecciones PCA de los datos temporales 7](#_Toc27401335)

[Fig. 5 Arquitectura GAN propuesta 7](#_Toc27401336)

[Fig. 6 Modelo híbrido propuesto 8](#_Toc27401337)

[Fig. 7 Identificación de una instancia normal (izquierda) y una anormal (derecha) 9](#_Toc27401338)

[Fig. 8 Arquitectura de una Replicator Neural Network (izquierda) y función de activación de la capa intermedia (derecha) 9](#_Toc27401339)

[Fig. 9 Cada dispositivo personaliza el modelo localmente (A). Las actualizaciones de muchos usuarios se agregan (B) para actualizar el modelo global que se compartirá con todos los usuarios (C), después de lo cual se repite el proceso. 10](#_Toc27401340)

[Fig. 10 Arquitectura de la U-Net utilizada 11](#_Toc27401341)

**Lista de ecuaciones**

[Ecuación1 13](#_Toc27401051)

**Lista de tablas**

[Tabla 1ejemplo 13](#_Toc27401266)

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Un evento anómalo es aquel que se produce de forma repentina y sin previsión. En un entorno industrial estos eventos (generalmente debidos a algún error) pueden provocar grandes daños por lo que su detección puede ayudar a prevenir situaciones críticas.

El objetivo del proyecto es detectar las posibles anomalías que se pudieran producir en los equipos industriales mediante el uso de un modelo de aprendizaje automático. Para enseñar a distinguir a un modelo entre un comportamiento normal y otro anormal se necesitan datos, contra más datos, mejor. Hoy en día la mayoría de los componentes dentro de un entorno industrial están constantemente monitorizados mediante el uso de dispositivos de medición especializados. Debido a esto, dispondremos de los datos suministrados por nuestros dispositivos de medición para entrenar al modelo, pero ¿y si pudiéramos disponer de más datos? Muchos equipos industriales son de uso genérico y pueden ser usados para muchas tareas en muchas compañías diferentes. Si pudiéramos tener acceso a los datos de todos esos dispositivos podríamos crear un modelo mucho mejor. Por diversas razones a las compañías no les gusta compartir sus datos, es por este motivo por lo que en este trabajo se propone el uso del Federated Learning (FL). Con el FL se pueden construir modelos aprovechando toda la información disponible manteniendo la privacidad de los datos.

Fig. 1 Los fallos no controlados pueden tener resultados catastróficos

## 1.2 Objetivos del Trabajo

Debido a la dificultad para obtener un conjunto de datos real por ser este tipo de datos muy sensible para las compañías no se utilizarán datos reales. En su lugar, se utilizará un software que nos permitirá la simulación de las instalaciones industriales que necesitemos.

Nuestro objetivo es la creación de un modelo de Machine Learning compatible con el uso del Federated Learning (En principio cualquier algoritmo basado en algún tipo de actualización de sus parámetros es compatible con el uso del FL [1]) que sea capaz de detectar de forma eficaz los eventos anómalos producidos en los entornos industriales simulados.

## 1.3 Enfoque y método seguido

El método que se seguirá se describe a continuación:

**Preparación del entorno de trabajo**

Se instalarán las herramientas necesarias para la implementación de la simulación, la gestión del conjunto de datos y para la creación del modelo.

**Implementación del software de simulación**

Se implementará un software que permitirá simular un entorno industrial. De esta simulación será de donde se obtendrán los datos necesarios para la creación del modelo.

**Análisis de los datos generados**

**Elección del framework Federated Learning**

Actualmente existen dos alternativas: TensorFlow y PySyft (basada en Torch) se evaluará cual es la más conveniente y se instalará el framework elegido en el entorno de trabajo.

**Elección del modelo a utilizar**

Posteriormente se decidirá sobre el tipo de modelos que se usarán basándonos en la información obtenida del análisis del estado del arte.

**Selección de los criterios de evaluación del modelo**

Se seleccionarán los métodos más convenientes para el caso de estudio (Precision, F-Score, Accuracy, etc.).

**Creación y evaluación de los modelos**

Se crearán tres tipos de modelos en función del conjunto de datos del que se disponga:

* Un modelo con todos los datos disponibles, simulando la existencia de un repositorio central para todos los datos.
* Varios modelos utilizando únicamente un pequeño conjunto de datos, simulando la existencia de “islas de información”.
* Un modelo Federated Learning con todos los datos de forma distribuida.

Una vez construidos estos modelos se analizará y se compararán sus rendimientos.

**Discusión de los resultados y análisis de posibles extensiones**

## 1.4 Planificación del Trabajo

La planificación temporal del trabajo es la siguiente:

* **18 septiembre – 20 octubre**:
  + Definir el alcance del TFM.
  + Decidir el conjunto de datos a usar.
  + Decidir el tipo de modelos/ técnicas a usar en base al estudio del estado de arte.
* **21 octubre – 21 diciembre:**
  + Instalación del entorno de trabajo
  + Obtención del conjunto de datos
  + Procesado del conjunto de datos
  + Generación de los modelos
  + Análisis de la calidad de los modelos
  + Análisis de los resultados y estudio de posibles extensiones
* **22 diciembre – 8 enero:**
  + Redacción de la memoria
  + Creación de la presentación
* **9 enero – 22 enero:**
  + Defensa pública del trabajo

A continuación, se muestra un diagrama con la programación establecida:



Figura 1. Planificación del trabajo

## 1.5 Breve sumario de productos obtenidos

No hay que entrar en detalle: la descripción detallada se hará en el resto de capítulos.

## 1.6 Breve descripción de los otros capítulos de la memoria

Explicación de los contenidos de cada capítulo y su relación con el trabajo en global.

# 2. Estudio del estado del arte

Este trabajo aborda dos temáticas que, aunque relacionas entre sí, puede ser estudiadas de forma independiente. Por este motivo se realizará el estudio del arte de manera independiente.

## 2.1 Detección de eventos anómalos

Un evento anómalo es una observación (o pequeño conjunto de observaciones) que ocurre infrecuentemente y que se desvía o es inconsistente con respecto al resto de observaciones lo suficiente como para indicar que se está produciendo una anomalía en el conjunto de datos.

La detección de este tipo de eventos se ha estudiado ampliamente en la literatura científica desde multitud de enfoques diferentes. A continuación, se muestran algunos de ellos:

**IRESE: An intelligent rare-event detection system using unsupervised learning on the IoT edge** [2]

Este documento propone un sistema de detección de eventos anómalos basado en el uso de modelos de Machine Learning no supervisados (clustering) en tiempo real. El procedimiento se divide en cuatro fases:

1. Captura de los datos.
2. Extracción de características.
3. Clustering online, se aplica el algoritmo BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) para generar micro-clusters.
4. Clustering offline, se utiliza Agglomerative Clustering que finalmente genera dos macro-clusters, uno que contiene los eventos normales y otro con los eventos raros.



Fig. 2 IRESE framework

Las principales ventajas que ofrece este framework son dos, una es que gran parte del proceso se lleva a cabo en dispositivos locales (edge devices) lo que reduce la sobrecarga en la red y la latencia del sistema. La otra es que no se necesita etiquetar los eventos anómalos debido al uso de algoritmos no supervisados. En cuanto a las desventajas, la principal sería la extracción de características ya que este procedimiento es muy dependiente del tipo de datos que se quieren procesar.

En cuanto a los resultados, el articulo menciona que se han obtenido valores de precision y de recall por encima del 90% en algunos casos.

**Anomaly detection in ECG time signals via deep long short-term memory networks** [3]

Este trabajo utiliza la detección de eventos anómalos con fines médicos. En particular se analiza el uso de redes neuronales LSTM (Long Short-Term Memory) profundas para la detección de anomalías en las señales EGC (electrocardiogramas).



Fig. 3 Arquitectura de la red propuesta

Si denotamos una señal EGC como X = {x(0),…, x(i), …, x(n)} donde cada x(i) representa un valor del EGC en el tiempo i. El procedimiento a grandes rasgos es el siguiente:

1. Se entrena la red con el siguiente objetivo: predecir los k siguientes valores del EGC para cada x(i).
2. A partir de la red construida se obtiene un conjunto de vectores de error para cada x(i). Con estos vectores de error se ajusta una distribución normal multivariante.
3. A partir de esa distribución se calcula la probabilidad de ocurrencia del vector de error obtenido y en función de un threshold se determina si ese punto anómalo o no.

Este método presenta varias ventajas sobre otros métodos:

1. No se requiere apenas preprocesamiento de los datos.
2. No se necesita saber a priori el tipo de anomalías que pueden aparecer.

El método proporciona resultados bastante buenos llegando valores para el

F-score por encima de 0.9 para varias clases de anomalías.

**Diagnosing Network-Wide Traffic Anomalies** [4]

El documento explora el uso del método PCA (Principal Analysis Components) para la detección de eventos anómalos en una red de comunicación. La idea consiste en dividir el espacio original en dos subespacios, uno que captura el comportamiento normal y otro el anómalo. El espacio normal estará formado por las componentes que capturan más varianza mientras que el espacio anómalo estará formado por los componentes que capturan menos varianza.



Fig. 4 Proyecciones PCA de los datos temporales

Gracias a esta división podremos descomponer cada instancia en sus componentes normales y anormales. Para determinar cuándo se ha producido un evento extraño se utiliza la norma del vector del espacio anómalo. Si esta norma no sobrepasa cierto threshold se considerará como un evento normal en otro caso se identificará como un evento anómalo.

Los resultados muestran altas tasas de detección y bajas tasas de falsos positivos.

**Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series** [5]

Este estudio utiliza una arquitectura GAN (Generative Adversarial Network) como mecanismo para la detección de eventos anómalos en un entorno industrial informatizado. Una arquitectura GAN está pensada como un sistema generativo (dada una distribución, el sistema es entrenado para generar datos que sigan esa distribución). Un sistema GAN esta formado por dos redes neuronales: una llamada generativa que es la encargada de generar datos que sigan la distribución dada y una llamada discriminante, que se encarga de ayudar al entrenamiento de la primera. El objetivo del método propuesto es entrenar la arquitectura GAN para que aprenda a generar situaciones de uso normal. A continuación, se muestra un esquema del sistema propuesto:



Fig. 5 Arquitectura GAN propuesta

Para determinar cuando una situación es anómala el sistema se basa en una puntuación (una combinación entre en la diferencia entre el input real y el generado y el valor de salida de la red discriminante).

Una de las principales ventajas de este método es que es no supervisado por lo que no es necesario el etiquetado de las muestras. Por otro lado, su principal desventaja es su alto coste computacional. Con respecto a su rendimiento, el documento menciona que el modelo mejora el desempeño de modelos anteriores en el conjunto de datos utilizado (Secure Water Treatment Testbed).

**High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning** [6]

Los algoritmos one-class SVM (1SVM) son una herramienta que demostrado ser efectiva en la tarea de la detección de valores anómalos. El problema con este tipo de modelos es que no escalan bien. En este trabajo se propone un método que permite solventar esa limitación gracias al uso de DBNs (Deep Belief Networks). La idea es poder sustituir los kernels no lineales de las 1SVM (muy costosos de calcular) por kernels lineales (menos costosos) gracias al uso de las DBN para la generación de vectores de características.



Fig. 6 Modelo híbrido propuesto

Los resultados experimentales demuestran que el modelo híbrido mejora a los modelos 1SVMs en aproximadamente un 20% en algunos conjuntos de datos. Comparado con los métodos basados en autoencoders la diferencia es prácticamente inapreciable sin embargo el método propuesto es tres veces más rápido en el entrenamiento y hasta 1000 veces más rápido en el testing.

**Isolation-Based Anomaly Detection** [7]

Lo que diferencia a esta propuesta del resto es que, en lugar de tratar de modelar las instancias normales para luego buscar los valores anómalos, el modelo explícitamente identifica las anomalías. Esto se consigue a través de los llamados Isolation Forests. Los Isolation Forest están formados por un conjunto de árboles de decisión. Cada árbol se crea de la siguiente manera, primero se selecciona un atributo aleatoriamente y luego se establece una partición también aleatoria entre el máximo y el mínimo del atributo seleccionado. La idea detrás de este algoritmo es que los puntos que están más aislados estarán en las hojas mas cercanas a nodo raíz.



Fig. 7 Identificación de una instancia normal (izquierda) y una anormal (derecha)

Este algoritmo puede tratar con conjuntos de datos grandes y de alta dimensionalidad. Sin embargo, este método presenta ciertos problemas debido a que las particiones se realizan únicamente de manera vertical u horizontal. Para solventar esos problemas se ha propuesto una modificación al algoritmo original en [8] llamado Extended Isolation Forest.

**Outlier Detection Using Replicator Neural Networks** [9]

La Replicator Neural Network utilizada en este artículo es MLP (Multi-Layer Perceptron) compuesto por tres capas ocultas. La función de esta red es la de reproducir los datos de entrada en la salida con el mínimo error posible.



Fig. 8 Arquitectura de una Replicator Neural Network (izquierda) y función de activación de la capa intermedia (derecha)

Una vez entrenada la red a cada punto se le asigna un valor que es la media de los errores de reconstrucción de cada una de las variables. Cuanto más grande sea el error más probable será que se trate de una anomalía. Este método es similar al uso de auto-encoders, la diferencia más llamativa es que este método utiliza una función de activación especial en forma de escalera la capa intermedia.

## 2.2 Federated Learning

El campo del Federated Learning es relativamente nuevo y se encuentra en pleno estudio es por este motivo que no existe tantos trabajos como en el caso del Machine Learning tradicional. Se han seleccionado tres trabajos, dos de definición e introducción a esta tecnología y uno centrado en un caso práctico de aplicación.

**Communication-Efficient Learning of Deep Networks from Decentralized Data** [10]

En este artículo se presenta un algoritmo que permite el uso de modelos de Machine Learning sobre conjuntos de datos descentralizados. En el documento no solo demuestra que esta idea es factible técnicamente presentando un algoritmo llamado *FederatedAveraging* sino que permite crear modelos de una alta calidad. Uno de los aspectos más importante de esta propuesta es que permite incrementar el nivel de privacidad de los clientes ya que en ningún momento se comparten datos entre clientes ni estos son enviados a un servidor central. A continuación, se muestra un esquema del modelo propuesto:



Fig. 9 Cada dispositivo personaliza el modelo localmente (A). Las actualizaciones de muchos usuarios se agregan (B) para actualizar el modelo global que se compartirá con todos los usuarios (C), después de lo cual se repite el proceso.

**Federated Machine Learning: Concept and Applications** [11]

Este documento presenta los conceptos básicos del Federated Learning sus principales técnicas y arquitecturas. Expone su potencial en varias aplicaciones y compara sus características con otros sistemas de aprendizaje distribuidos (Distributed Machine Learning, Federated Database Systems, etc.). Adicionalmente introduce el concepto de Data Alliance of Enterprises. Una Data Alliance of Enterprises es un conjunto de organizaciones que, mediante el uso de sus datos (sin compartirlos directamente con los demás miembros) colaboran para crear un modelo que aporte beneficios a todos los participantes. Este tipo de alianzas están llamadas a ser el futuro del Machine Learning y se espera constituyan todo un nuevo modelo de negocio [12].

**Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation** [13]

En este documento se presenta el primer uso del Federated Learning para el tratamiento de imágenes para el diagnóstico médico. En particular se usa esta tecnología para la creación de un modelo distribuido de segmentación de imágenes (U-Net) sin compartir datos de pacientes entre instituciones.



Fig. 10 Arquitectura de la U-Net utilizada

El estudio demuestra que el modelo basado en Federated Learning consigue un rendimiento prácticamente equivalente al modelo creado con el uso de datos compartidos.

## 2.3 Novedades propuestas

Aunque la detección de eventos anómalos ha sido estudiada en profundidad desde múltiples perspectivas nunca se había abordado el problema de forma colaborativa y descentralizada. Esta es la principal novedad que pretende aportar este trabajo.

# 3. Diseño e implementación del trabajo

## 3.1 Consideraciones previas

El objetivo del trabajo es mostrar cómo construir un modelo de aprendizaje automático federado que pueda ser aplicado en una situación real para detección precoz de potenciales fallos en máquinas dentro de un entorno industrial monitorizado. Este tipo de entornos monitorizados hoy en día están muy extendidos gracias al auge de la llamada Industria 4.0 y la implementación de dispositivos IoT (Internet of Things) en las fábricas. Sin embargo, como ya se mencionó en apartados anteriores es prácticamente imposible obtener un conjunto de datos real por ser este tipo de datos muy sensible para las compañías. Por este motivo no se utilizarán datos reales. En su lugar, se ha construido un software que nos permitirá la simulación de estos datos.

## 3.2 Preparación del entorno de trabajo

La mayor parte del desarrollo del proyecto se llevará a cabo con el lenguaje de programación Python. Python ofrece, entre otras, las siguientes ventajas [14]:

* **Simplicidad.** Python es conocido por hacer que los programas funcionen en la menor cantidad de líneas de código. Identifica y asocia automáticamente los tipos de datos y, en general, resulta un lenguaje fácil de usar y toma menos tiempo en la codificación.
* **Compatibilidad.** Muchas de las tecnologías actuales relacionadas con el Machine Learning están pesadas para ser utilizadas con este lenguaje.
* **Facilidad de aprendizaje.** En comparación con otros lenguajes, Python es fácil de aprender incluso para los programadores con menos experiencia. Principalmente por tres razones:
  + Cuenta con amplios recursos de aprendizaje.
  + Garantiza un código legible.
  + Se rodea de una gran comunidad.
* **Variedad de paquetes.** Python tiene un poderoso conjunto de paquetes para una amplia gama de necesidades de análisis y ciencia de datos( numPy, Pandas, Scipy, Scikit-learn, etc.
* **Visualización de datos.** Aunque no es el mejor lenguaje en lo que respecta a la visualización de datos. Actualmente existen APIs que pueden ofrecer resultados bastante buenos.

Nuestro entorno de trabajo estará formado por las siguientes herramientas:

* **Microsoft Excel.** Software de hojas de cálculo
* **Python.** En su versión 3.6
* **PyCharm.** Editor para el lenguaje de programación Python

## 3.3 Implementación del software de simulación

### 3.3.1 Introducción

El objetivo de la implementación de este software es la simulación de un entorno industrial cuyas instalaciones se encuentran monitorizadas mediante dispositivos IoT. Debido a las características únicas de cada tipo de instalación no nos es posible generar un software que simule cada una de las diferentes máquinas que pudieran existir. Por lo tanto, consideraremos únicamente un tipo de máquina: máquinas rotatorias genéricas. Basándonos en conjuntos de datos ya existentes (por ejemplo, Turbofan Engine Degradation Simulation Data Set) se han elegido los siguientes parámetros de operacionales[[1]](#footnote-1):

* Velocidad rotacional
* Temperatura
* Presión

Además de estos parámetros se tendrán en cuenta variables relacionadas con el entorno. En particular la temperatura del entorno y la presión atmosférica.

### 3.3.2 Caracterización de la degradación de las máquinas

Con el propósito de una generación de datos inspirada en la física se utilizará una aproximación similar a la utilizada en [15]. En nuestro caso se usará la siguiente ecuación para simular la degradación de una máquina en función del tiempo de uso:

Ecuación 1 Degradación

Donde, *a* y *b* son coeficientes que determinan la forma de la curva y *d* es la degradación inicial. A continuación se muestra un ejemplo de estas curvas:



Figura 2 Curvas de degradación para diferentes parámetros

Podemos observar que la curva punteada representa una degradación repentina mientras que la curva continua representaría una degradación más constante ( aunque el fallo se produciría antes).

En nuestro caso se considerarán dos curvas de degradación para cada máquina. Esto equivaldría a decir que cada máquina puede fallar por dos motivos diferentes:

* Uno estrechamente relacionado con la presión y/o la temperatura (a esta curva la llamaremos ***h1).***
* Una más relacionada con la velocidad (que denotaremos por ***h2****)*

### 3.3.3 Evolución de los parámetros operacionales en función de la degradación

Para cada uno de los parámetros operacionales se propondrá una fórmula que nos dará su valor en el tiempo *t* en función de:

* Su valor en *t-1*
* Valores del resto de parámetros en *t-1*
* Valor de la curva de degradación en *t*
* Parámetros adicionales

Las fórmulas propuestas están basadas en las fórmulas presentadas en [16].

**Cálculo de la velocidad**

La velocidad en *t* será una media entre a velocidad en *t-1* y la velocidad objetivo ponderada con un coeficiente relacionado con la función de degradación***h2.***

A continuación se muestra el efecto de la degradación en la evolución de la velocidad:

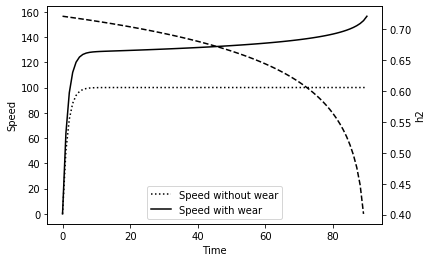


Figura 3 Efecto de la degradación en la velocidad

En este caso, en el que la velocidad objetivo era 100. Podemos ver como el efecto del desgaste afecta a la velocidad real. Al principio la diferencia se mantiene más o menos constante empezando a aumentar notablemente al final de la vida útil de la máquina.

Tabla 1ejemplo

|  |  |
| --- | --- |
| yu | uty |
| uty | uyt |

**Análisis de los datos generados**

**Elección del framework Federated Learning**

Actualmente existen dos alternativas: TensorFlow y PySyft (basada en Torch) se evaluará cual es la más conveniente y se instalará el framework elegido en el entorno de trabajo.

**Elección del modelo a utilizar**

Posteriormente se decidirá sobre el tipo de modelos que se usarán basándonos en la información obtenida del análisis del estado del arte.

**Selección de los criterios de evaluación del modelo**

Se seleccionarán los métodos más convenientes para el caso de estudio (Precision, F-Score, Accuracy, etc.).

**Creación y evaluación de los modelos**

# 4. Conclusiones

Este capítulo tiene que incluir:

* Una descripción de las conclusiones del trabajo: Qué lecciones se han aprendido del trabajo?.
* Una reflexión crítica sobre el logro de los objetivos planteados inicialmente: Hemos logrado todos los objetivos? Si la respuesta es negativa, por qué motivo?
* Un análisis crítico del seguimiento de la planificación y metodología a lo largo del producto: Se ha seguido la planificación? La metodología prevista ha sido la adecuada? Ha habido que introducir cambios para garantizar el éxito del trabajo? Por qué?
* Las líneas de trabajo futuro que no se han podido explorar en este trabajo y han quedado pendientes.

# 4. Glosario

Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

# 5. Bibliografía

|  |  |
| --- | --- |
| [1] | F. Hartmann. [En línea]. Available: https://florian.github.io/federated-learning/. [Último acceso: 15 10 2019]. |
| [2] | Z. Janjua, M. Vecchio, M. Antonini y F. Antonelli, «IRESE: An intelligent rare-event detection system using unsupervised learning on the IoT edge,» *Engineering Applications of Artificial Intelligence,* vol. 84, pp. 41-50, 2019. |
| [3] | S. Chauhan y L. Vig, «Anomaly detection in ECG time signals via deep long short-term memory networks,» de *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Paris, 2015. |
| [4] | A. Lakhina, M. Crovella y C. Diot, «Diagnosing Network-Wide Traffic Anomalies,» *Computer Communication Review,* vol. 34, 2004. |
| [5] | D. Li, D. Chen, J. Goh y S. k. Ng, «Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series,» de *International Workshop on Big Data*, London, 2018. |
| [6] | S. M. Erfani, S. Rajasegarar, S. Karunasekera y C. Leckie, «High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning,» *Pattern Recognition,* vol. 58, pp. 121-134, 2016. |
| [7] | F. T. Liu, K. Ting y Z.-H. Zhou, «Isolation-Based Anomaly Detection,» *ACM Transactions on Knowledge Discovery From Data - TKDD,* vol. 6, pp. 1-39, 2012. |
| [8] | S. Hariri, M. Carrasco Kind y R. J. Brunner, «Extended Isolation Forest,» 2018. |
| [9] | S. Hawkins, H. He, G. Williams y R. Baxter, «Outlier Detection Using Replicator Neural Networks,» *Data Warehousing and Knowledge Discovery,* pp. 170-180, 2002. |
| [10] | H. B. McMahan, E. Moore, D. Ramage, S. Hampson y B. Agüera y Arcas, «Communication-Efficient Learning of Deep Networks from Decentralized Data,» 2016. |
| [11] | Q. Yang, Y. Liu, T. Chen y Y. Tong, «Federated Machine Learning: Concept and Applications,» 2019. |
| [12] | A. Gonfalonieri, «Towards Data Science,» [En línea]. Available: https://towardsdatascience.com/federated-learning-a-new-ai-business-model-ec6b4141b1bf. |
| [13] | M. J. Sheller, G. A. Reina, B. Edwards, J. Martin y S. Bakas, «Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation,» 2018. |
| [14] | U. I. d. Valencia, «www.universidadviu.es,» [En línea]. Available: https://www.universidadviu.es/python-para-big-data-motivos-para-elegirlo/. [Último acceso: 15 12 2019]. |
| [15] | K. G. D. S. N. E. A. Saxena, «Damage propagation modeling for aircraft engine run-to-failure simulation,» de *International Conference on Prognostics and Health Management,*, Denver, CO, USA, 2008. |
| [16] | Microsoft, «gallery.azure.ai,» [En línea]. Available: https://gallery.azure.ai/Collection/Predictive-Maintenance-Template-3. [Último acceso: 16 11 2019]. |

# 6. Anexos

Listado de apartados que son demasiado extensos para incluir dentro de la memoria y tienen un carácter autocontenido (por ejemplo, manuales de usuario, manuales de instalación, etc.)

Dependiente del tipo de trabajo, es posible que no haya que añadir ningún anexo.

1. Las unidades de medida serán: kPa para las presiones, ºC para las temperaturas y rpm para las velocidades [↑](#footnote-ref-1)