ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

APLICACIÓN DE TÉCNICAS DE ANÁLISIS DE DATOS PARA LA GESTION DE INFRAESTRUCTURAS IoT

(Application of Data Analytics for the management of IoT Infrastructures)

Para acceder al Titulo de

Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: Iulian Antonov

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Iulian Antonov
Director del TFG: Luis Sánchez González
Título: "Aplicación de Técnicas de Análisis de Datos para la Gestión de Infraestructuras IoT"
Title: "Application of Data Analytics for the management of IoT Infrastructures"

Presentado a examen el día: 24 de octubre de 2019

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): José Basterrechea Verdeja
Secretario (Apellidos, Nombre): Jorge Lanza Calderón
Vocal (Apellidos, Nombre): Luis Sánchez González

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº (a asignar por Secretaría)

Índice

ÍNDICE DE FIGURAS

Resumen

En este documento se presenta un estudio de la aplicación de técnicas de análisis de datos para la gestión de infraestructuras de la Internet de las Cosas (IoT).

Una infraestructura IoT se caracteriza por estar compuesta de una gran cantidad de dispositivos que en su mayoría disponen de bajas capacidades. Además de una enorme cantidad de datos que generan estos dispositivos, es habitual que presenten comportamientos anómalos, sean temporales o permanentes. Asimismo, teniendo en cuenta del hecho que gran parte de estos están expuestos ante condiciones atmosféricas, estos dispositivos con frecuencia generan información sesgada. Ante estos retos, es necesario que las plataformas que gestionan dichos dispositivos dispongan de mecanismos de gestión y eliminación de información tendenciosa en la medida de lo posible.

Para un estudio y diseño de posibles soluciones, se tendrán en cuenta las características de este tipo de infraestructuras y las soluciones que puedan aprovecharse del paradigma del Fog Computing para una mejora de latencia y ahorro de recursos en la red. La validación y pruebas de dichos sistemas se hará en una infraestructura real como es SmartSantander.

Palabras Clave – SmartSantander, IoT, análisis de datos.

Abstract

This document presents a study of the application of data analysis techniques for the management of Internet of Things (IoT) infrastructure.

An IoT infrastructure is characterized by a large number of devices, most of which have low capacities. In addition to the enormous amount of data generated by these devices, it is common for them to exhibit anomalous behaviour, whether temporary or permanent. Also, given the fact that many of these are exposed to atmospheric conditions, these devices often generate biased information. Faced with these challenges, it is necessary for the platforms that manage these devices to have mechanisms for managing and suppressing tendentious information as far as possible.

For a study and design of possible solutions, the characteristics of this type of infrastructure will be taken into account, as well as the solutions that can take advantage of the paradigm of fog computing to improve latency and save resources on the network. The validation and testing of these systems will be done in a real infrastructure such as SmartSantander.

Keywords – SmartSantander, IoT, data analytics

Lista de Acrónimos

API – Application Programming Interface (Interfaz de Programación de Aplicaciones).

FIFO – First In First Out (Primero en entrar, primero en salir)

HF – High Frequency (Alta Frecuencia)

IoT – Internet of Things (Internet de las Cosas)

KDD – Knowledge Discovery in Database (Descubrimiento de Conciemiento en Bases de Datos)

KNN – K-Nearest Neighours (K vecinos más cercanos)

LF – Low Frequency (Baja Frecuencia)

MD – Mahalanobis Distance (Distancia de Mahalanobis)

M2M – Machine to Machine (máquina a máquina, comunicación entre máquinas)

NaN – Not a Number (no es un número)

OLAP- On-Line Analytical Processing (procesamiento analítico en línea)

QR – Quick Response Code (código de respuesta rápida)

RFID – Radio Frequency Identification (Identificación por Radiofrecuencia)

SDO – Sistema de Detección de Outliers

SGD – Sistema de Gestión de Disponibilidad

SVM – Support-Vector Machines (Máquinas de Vectores de Soporte)

UHF – Ultra High Frequency (Frecuencia Ultra Alta)

VA – Video Analytics (Análisis de Vídeo)

Capítulo 1: Introducción

1.1 Introducción

La Internet de las cosas (IoT) prevé una red mundial interconectada de entidades físicas inteligentes. Estas entidades físicas generan una gran cantidad de datos en funcionamiento y, a medida que la IoT gana impulso en términos de despliegue, la escala combinada de esos datos parece destinada a seguir creciendo. Cada vez más, las aplicaciones de la IoT implican análisis. La analítica de datos es el proceso de derivar conocimiento de los datos, generando valor como percepciones procesables a partir de ellos.

La IoT ha ido cobrando impulso tanto en la industria como en las comunidades de investigación debido al increíble aumento del número de dispositivos móviles, sensores inteligentes y a las aplicaciones potenciales de los datos producidos en un amplio espectro de ámbitos. En su informe de 2013, McKinsey [1] señala un crecimiento del 300% en dispositivos IoT conectados en los últimos cinco años y califica el impacto económico potencial de la IoT de 2,7 a 6,2 trillones de dólares anuales para 2025. Estas cifras aumentaron a 4 trillones de dólares y 11 trillones de dólares en 2015 [2]. Otra tecnología interesante que excede el impulso de la IoT en el ciclo de exageración es la del big data, del que la IoT sirve de fuente y sumidero.

Big data se define como datos demasiado grandes (volumen), demasiado rápidos (velocidad) y demasiado diversos (variedad). En el contexto de la IoT, vemos un ejemplo de volumen en el Gran Desafío DEBS 2014 [3], donde los datos de 40 casas con enchufes inteligentes produjeron 4.000 billones de eventos en un mes, dado que un censo del año 2011 mostró que había 26,4 millones de hogares en el Reino Unido [4], el tamaño de los datos proyectado de 2,64 cuatrillones (escala corta) por mes si cada casa tuviera un medidor, es un buen ejemplo de datos demasiado grandes. En los casos de uso de la IoT de sistemas de transporte inteligentes y de telecomunicaciones, los flujos de datos pueden llegar demasiado rápido para ser procesados, lo que representa un problema de velocidad de datos. Por último, el término global utilizado para describir la presencia de fuentes de datos heterogéneas en la IoT es demasiado diverso, lo que dificulta el análisis de las herramientas existentes. En una encuesta realizada en 2014 a científicos de datos, el 71% de los entrevistados dijo que el análisis se está volviendo cada vez más difícil debido a la variedad y los tipos de fuentes de datos [5]. Un ejemplo es el caso del uso de la asistencia sanitaria personal de la IoT, en el que las historias clínicas electrónicas textuales no estructuradas, los dispositivos móviles conectados y los sensores contribuyen al problema de la variedad [5].

El motivo para implementar tal infraestructura reside en la eficiencia, control, seguridad y calidad de vida de los ciudadanos en general. Esto es posible debido a la implementación de sistemas que administran y controlan los sensores instalados según qué propósito. Muchos de estos sistemas se usan para gestionar una cadena de

producción, la seguridad de un coche, el tráfico de las ciudades, cálculos de rutas alternativas con la intención de evitar atascos o ahorro de combustible y mucho más.

Por lo general una vez recibidas las mediciones de los dispositivos el sistema que los administra suele guardar los datos sin ningún tipo de análisis. Esto implica que entre los datos guardados se encuentran mediciones falsas debido a un mal calibrado o suciedad de los sensores. Algunos de los principales problemas al implementar este tipo de sistemas son la interacción inteligente, el almacenamiento y la organización de datos, la gestión y el análisis. Por lo tanto, incluso si el sistema se implementa, la pregunta principal podría ser cómo manipular esa cantidad de datos y cómo hacerlo de la manera más eficiente, de modo que los recursos que se utilizarán desde la red principal sean mínimos.

Como los principales problemas que describo son un tema de la junta [6], me concentré en el desarrollo e implementación de diferentes aplicaciones de gestión de datos en la infraestructura del proyecto SmartSantander. Esta plataforma ofrece una instalación para la investigación experimental a escala de ciudad en apoyo de aplicaciones y servicios típicos para una ciudad inteligente. Es lo suficientemente grande, abierta y flexible para el desarrollo de nuevas aplicaciones por parte de los usuarios de varios tipos, incluida la investigación experimental avanzada sobre tecnologías IoT.

Algunas integraciones de servicios que gestiona SmartSantander son el alumbrado público, parques y jardines, policía local, gestión de residuos, servicio de transportes, etc. En consecuencia, esta plataforma gestiona una gran cantidad de sensores que están instalados por toda la ciudad. Debido a un gran número de sensores se necesita una aplicación que gestione los dispositivos que están activos, que han dejado de funcionar o que envían datos erróneos para así poder ejercer un mantenimiento eficiente del proyecto. Otro problema para abordar es la sobrecarga que sufre la red al estar enviando gran cantidad de datos de forma constante que a su vez se procesan en los servidores centrales en la nube. Una posible solución podría ser el procesado de los datos en los gateways, antes de ser enviados a la nube. Esto aliviaría la sobrecarga en los servidores aumentando la velocidad de respuesta de estos ante situaciones de emergencia, así mismo como el ahorro de recursos de la red.

1.2 Objetivos del proyecto

El presente Trabajo de Fin de Grado tiene como objetivo fundamental el estudio de la aplicación de técnicas de análisis de datos en las labores de gestión y supervisión de infraestructuras de la IoT a gran escala. Normalmente, este tipo de infraestructuras está compuesto por una gran cantidad de dispositivos que, en su mayoría, disponen de bajas capacidades. Además, su gran número y la ingente cantidad de información que generan continuamente son importantes condicionantes para tener en cuenta a la hora de plantear soluciones para su gestión. Es habitual que los dispositivos que componen estas infraestructuras presenten comportamientos anómalos (temporales o permanentes) y la información que generan se vea comprometida por ello. Ante estos retos es necesario que las plataformas que gestionan estas infraestructuras y exportan la información que generan, dispongan de mecanismos que monitoricen el estado de los dispositivos IoT que las integran, alerten acerca del comportamiento anómalo de estos dispositivos y filtren la información errónea que puedan enviar.

En este trabajo de fin de grado se abordarán estas problemáticas y para ello se diseñarán, implementarán y validarán una serie de mecanismos basados en análisis de datos que se exponen a continuación:

- Diseño y desarrollo de un sistema para la evaluación de la disponibilidad de dispositivos IoT.
- Diseño de una arquitectura de Fog Computing para la supresión de outliers y reducción de la dimensionalidad en infraestructuras IoT.
- Implementación y evaluación de técnicas basadas en el algoritmo K-Nearest Neighbors (KNN) para la supresión de outliers.
- Implementación y evaluación de técnicas basadas en Support-vector machine (SVM) para la reducción de la dimensionalidad.

Para el diseño de las soluciones se tendrán en cuenta las características de este tipo de infraestructuras y en particular se analizarán soluciones que puedan aprovecharse del paradigma del Fog Computing y de las ventajas que este ofrece en cuanto a menor latencia y reducción de la sobrecarga de datos en la red.

La validación de los mecanismos se realizará empleando para ello una infraestructura IoT real como es la plataforma SmartSantander.

1.3 Resumen Ejecutivo

Tras la introducción realizada en este capítulo, donde se han presentado la motivación y los objetivos del proyecto, el resto del documento sigue una estructura que se comenta a continuación.

En el capítulo 2 se describen las bases del funcionamiento de la plataforma SmartSantander así como una introducción a las técnicas de análisis de datos para las infraestructuras IoT. Además, se hace una descripción detallada de los métodos KNN y SVM que se han empleado como técnicas de análisis durante la realización de este trabajo. En el capítulo 3 se presenta un sistema de gestión de disponibilidad para los dispositivos de una infraestructura IoT. Se describe la arquitectura del sistema de gestión y se detalla la implementación de éste así como su integración y las pruebas de validación a las que se le sometió. En el capítulo 4 se describe un sistema para la eliminación de outliers y reducción de la cantidad de datos a enviar. Además de la descripción de la arquitectura del sistema se introduce la solución implementada para la eliminación de medidas erróneas, denominadas outliers, mediante el uso de técnicas de KNN. Del mismo modo, se describe la solución para permitir la reducción del volumen de datos generados mediante el uso de técnicas basadas en SVM y el paradigma del Fog Computing. Por último, en el capítulo 5 se analizan los resultados definidos con el fin de poder alcanzar una serie de conclusiones acerca de la consecución de los objetivos que nos marcamos al inicio del TFG. Igualmente se presentan las que pudieran ser líneas futuras de actuación de cara a extender el trabajo y corregir y ampliar sus resultados.

Capítulo 2: Marco de Desarrollo

En este capítulo se describen las bases de funcionamiento de la plataforma SmartSantander así como una introducción a las técnicas de análisis de datos para las infraestructuras IoT.

2.1 SmartSantander

SmartSantander es un proyecto de investigación científica perteneciente al 7º Programa Marco de la Comisión Europea, en el que se han diseñado, desplegado y validado una plataforma para la experimentación en la IoT compuesta por más de 20.000 dispositivos (sensores, captadores, actuadores, cámaras, terminales móviles, etcétera) por toda la capital cántabra, formando un espacio virtual donde los objetos se comunican entre sí y trasmiten información para las personas con el fin de mejorar su bienestar (calidad de vida).

El campo de pruebas que la ciudad ofrece, crea un esquema de colaboración, entre lo público y lo privado, cuestión que ha sido desde el inicio, uno de los motores del proyecto SmartSantander, no ya sólo en el Proyecto Europeo que le dio nombre, sino en el global de la idea de Santander como ciudad inteligente

SmartSantander se encuadra dentro del reto de las Redes Ubicuas y Confiables e Infraestructuras de Servicio, con el objetivo destinado a desarrollar la investigación, en relación a la Internet del Futuro, basada en la experimentación sobre infraestructuras reales.

El núcleo principal de las instalaciones que comprenden más de 20000 dispositivos, se localiza en la ciudad de Santander y sus alrededores, incluyendo puntos singulares de la Comunidad de Cantabria.

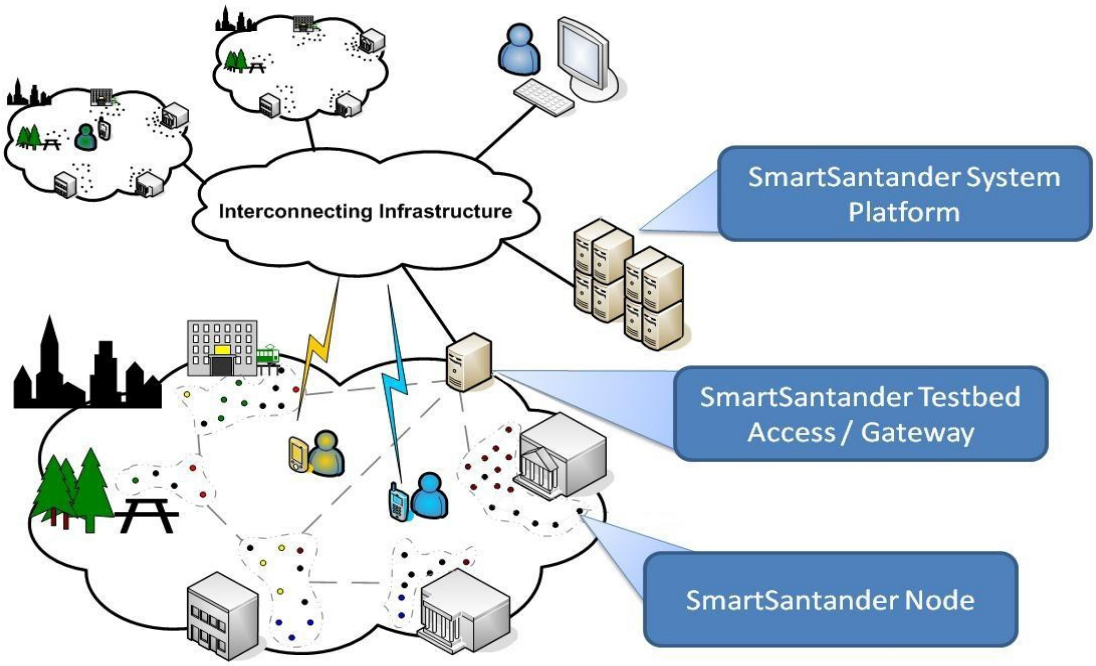


Figura 1Arquitectura de SmartSantander

SmartSantander es una aplicación real y a gran escala de la llamada "computación ubicua", y esta computación es un paradigma de la telemática que tiene por objeto la integración de pequeños dispositivos y sensores en todo los aparatos y objetos de nuestra vida cotidiana, de tal forma que se puedan interconexionarse entre ellos, pudiéndose intercambiar de este modo información útil entre ellos. Lo que se ha venido también a llamar comunicación Máquina a Máquina (M2M).

Como vemos en la Figura 1, los dispositivos que utiliza son de muy pequeño tamaño (nodos), con una capacidad de computación muy limitada, pero a los que es posible acoplar sensores de diversa naturaleza para que puedan captar datos del entorno. Estos datos fluyen a través de la red hasta un punto de concentración donde la información que captan los sensores es tratada con el objeto de obtener un servicio útil.

Para la consecución de los objetivos del proyecto, la plataforma software y la infraestructura física desplegada deben ofrecer una serie de características necesarias para que la futura experimentación pueda realmente suponer un avance cualitativo en la investigación e innovación de los campos de la IoT, las Ciudades Inteligentes y la Internet del Futuro.

2.1.1 Plataforma SmartSantander

La plataforma SmartSantander, se basa en una arquitectura formada por tres niveles:

- Nivel de Dispositivo IoT, que proporciona el sustrato necesario compuesto por los propios dispositivos; estos son recursos limitados y exportan datos fiables que aseguran con una serie de medidas.
- El nivel de Gateway IoT, que enlaza los dispositivos IoT en los bordes de la red a una infraestructura de red central.
- El nivel de Servidor, que dispone de dispositivos de gran capacidad, los cuales son conectados directamente con la infraestructura de red central. Los servidores pueden usarse como repositorios de datos IoT, y como servidores de aplicación

que pueden ser configurados para ofrecer una gran variedad de diferentes servicios IoT y aplicaciones.

2.1.2 Análisis de generación de datos de implementación de ciudades

Dada la gran cantidad de datos que se generan y deben ser manejados, almacenados y puestos a disposición , en esta sección se resumirá la forma en que se generan y las cifras aproximadas de datos a tratar.

Patrones de generación de datos

El patrón de generación de datos se define por el servicio para el que está destinado. Podemos identificar dos patrones: el de generación de observación periódica y de basada en eventos.

Los dispositivos IoT programados con el patrón de observación periódica informarán una observación que contiene la información detectada en una base de frecuencia configurable (Dispositivos para servicios de vigilancia ambiental, de intensidad del tráfico y de riesgos de parques y jardines). Sin embrago los dispositivos de IoT que se dedican al estacionamiento al aire libre funcionan en forma de evento, así solo informan sobre la detección de un cambio en el parámetro que están monitorizando (Sensibilización Participativa).

Cantidad de datos generados

Uno de los objetivos de la plataforma SmartSantander era apoyar la experimentación avanzada de la IoT, y para ello y dado que el patrón de generación de observación periódica es configurable, el objetivo era establecer una alta frecuencia.

Teniendo en cuenta sólo las necesidades de servicio, la frecuencia seleccionada conduce a una situación de sobremuestreo. Sin embargo, esto permitió una experimentación más amplia. Para la mayoría de los dispositivos, el período de tiempo utilizado para informar de las nuevas observaciones se fijó en cinco minutos. Para los dispositivos que usan el patrón de generación de observación basado en eventos, el número de observaciones reportadas depende solamente del uso real del servicio.

La Figura 2 resume el número de observaciones generadas diariamente dentro del lecho de pruebas SmartSantander durante marzo de 2014.

Figura 2 Número de Observaciones generadas diariamente en el banco de pruebas de SmartSantander

Servicio	Observaciones diarias
Monitoreo ambiental	139.370
Irrigación de Parques y Jardines	8,365
Monitorización Ambiental Móvil	82.726
Ocupación de aparcamiento	13.489
La gestión del tráfico	54.720
Detección Participativa	6.352
Realidad aumentada	1.489

Este Proyecto, es único en el mundo y ofrecerá un excelente campo de experimentación a la comunidad científica europea, puesto que SmartSantander

establecerá las bases para la comunicación entre elementos heterogéneos, permitiendo la federación con otras redes de similar naturaleza en el resto de Europa y el mundo. Este Proyecto, hace posible que con la gran cantidad de sensores de que dispone, se cumpla el objetivo de construir una “Ciudad Inteligente”.

2.2 Análisis de datos para gestión de infraestructuras IoT

La analítica es la ciencia o método de usar el análisis para examinar algo complejo [7]. Cuando se aplica a los datos, la analítica es el proceso de derivar (el paso del análisis) el conocimiento y las percepciones de los datos (algo complejo). La evolución del concepto de analítica que vemos hoy en día se remonta a 1962. Tukey [8]definió en primer lugar el análisis de datos como procedimientos de análisis de datos, técnicas de interpretación de los resultados, reunión de datos que facilitan el análisis, lo hacen más preciso y detallado y, por último, toda la maquinaria relacionada y los métodos estadísticos utilizados. En 1996, Fayyad et al. [9] publicaron un artículo explicando el Descubrimiento de Conocimientos en Bases de Datos (KDD) como "el proceso general de descubrir conocimiento útil a partir de datos" donde la minería de datos sirve como un paso en este proceso. "la aplicación de algoritmos específicos para extraer patrones de los datos". En 2006, Davenport [10] introdujo el análisis como modelo cuantitativo, estadístico o predictivo para analizar problemas de negocio como el rendimiento financiero o las cadenas de suministro y destacó su aparición como una herramienta de toma de decisiones para las empresas. En 2009, Varian [11] destacó la capacidad de tomar datos y "comprenderlos, procesarlos, extraer valor de ellos, visualizarlos y comunicarlos", como una habilidad de gran importancia en la próxima década. En 2013, Davenport [12] introdujo los conceptos de Analytics 1.0, analítica tradicional, 2.0, el desarrollo de la tecnología de grandes datos y 3.0, donde esta gran tecnología de datos se integra rápidamente con la analítica, produciendo una rápida comprensión e impacto en el negocio.

Una pregunta importante que debe hacerse siguiendo la definición de IoT y de su visión es la ventaja que ofrecen las "cosas" conectadas con respecto a los dispositivos aislados. Por ejemplo, ¿cuál es el beneficio de desplegar un sistema de aparcamiento inteligente en comparación con tener sensores aislados en un aparcamiento utilizando señales visuales de color verde o rojo en el techo para indicar si un aparcamiento está vacío u ocupado? El análisis añade valor a los datos y el contexto integrado de IoT, lo que produce una mayor comprensión del valor. El sistema de aparcamiento inteligente de análisis tiene un espacio de observación mucho más amplio y también guía al usuario hacia el aparcamiento disponible de forma eficiente, sin intervención humana, reduciendo el tráfico y la contaminación.

2.2.1 Tipos de análisis y su importancia

En este apartado se presenta una categorización de las capacidades analíticas de la literatura analítica de negocios, de la cual proviene el término analítica. Bertolucci et al. [13] proponen categorías descriptivas, predictivas y prescriptivas mientras que Gartner [14] propone la categoría extra de analítica diagnóstica. Finalmente, Corcoran et al. [15] introducen la categoría adicional de análisis de descubrimiento. Hall et al. [17] se basan en ellos para formar una clasificación completa de las capacidades analíticas que consta de cinco categorías: descriptiva, diagnóstica, de descubrimiento, predictiva y de análisis prescriptivo.

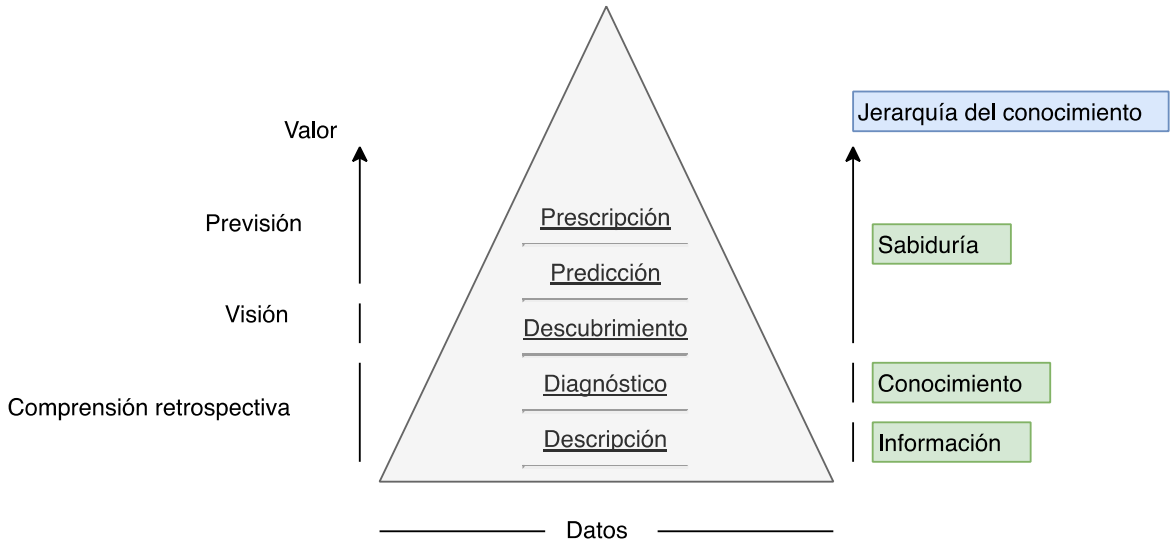


Figura 3 La Analítica de las jerarquías de Conocimiento y Valor

La Figura 3 muestra cómo encaja cada capacidad analítica dentro de la Jerarquía del Conocimiento, que es un marco común utilizado en el dominio de la Gestión del Conocimiento. Esta categorización de las capacidades analíticas nos permite establecer cuál es el objetivo del análisis y relacionarlo con la visión del despliegue del IoT tal y como se expresa a menudo en las hojas de ruta de la investigación. El valor de cada capacidad también se resalta en la Figura 3.

La jerarquía de conocimientos comienza con los datos de la base, cuyos ejemplos son hechos, cifras y observaciones (por ejemplo, los datos brutos producidos por las "cosas" de la IoT). La información se interpreta como datos con contexto, por ejemplo, la temperatura representada por el análisis descriptivo: una media de un mes o una descripción categórica del día soleado y cálido.

El conocimiento es información dentro de un contexto con mayor comprensión y significado, quizás posibles razones para la alta temperatura media de este mes. Por último, la sabiduría es el conocimiento con perspicacia, por ejemplo, descubrir una tendencia particular en la temperatura y proyectarla a lo largo de los próximos meses, al tiempo que se proporcionan soluciones de gestión de la energía que ahorran costes para calentar una vivienda inteligente basadas en estas predicciones. Cada componente de la jerarquía de conocimiento se basa en el nivel anterior y podemos ver algo similar con capacidades analíticas. Para añadir una visión práctica de la literatura de gestión empresarial a nuestra discusión, una revisión de las organizaciones que adoptan la analítica las categorizó como Aspirantes, Experimentadas y Transformadas. Se consideró que las organizaciones con aspiraciones utilizaban el análisis en retrospectiva como justificación de las acciones, utilizando los niveles de datos, información y conocimiento en el proceso. Las organizaciones experimentadas utilizaron los conocimientos para guiar sus decisiones y las organizaciones transformadas se caracterizaron por su capacidad de utilizar el análisis para prescribir sus acciones, aplicando eficazmente la previsión en su proceso de toma de decisiones.

2.2.2 Cinco categorías de capacidades analíticas

1. Análisis Descriptivo. Nos ayuda a responder a la pregunta, "¿qué pasó?". Puede adoptar la forma de describir, resumir o presentar los datos brutos de IoT que se han recopilado. Los datos son decodificados, interpretados en contexto, fusionados y luego

presentados para que puedan ser entendidos y puedan tomar la forma de un gráfico, un informe, estadísticas o alguna agregación de información.

2. Análisis Diagnóstico. Es el proceso de entender por qué ha ocurrido algo. Esto va un paso más allá de la analítica descriptiva en el sentido de que tratamos de encontrar la causa de fondo y las explicaciones para los datos de IoT. Tanto el análisis descriptivo como el diagnóstico nos dan una visión retrospectiva de qué y por qué han sucedido las cosas.

3. Descubrimiento en Analítica. Mediante la aplicación de inferencia, razonamiento o detección de información no trivial a partir de datos IoT brutos, tenemos la capacidad de Descubrimiento en Analítica. Dado el grave problema de volumen que presentan los grandes datos, el descubrimiento en analítica es también muy valioso para reducir el espacio de búsqueda de las aplicaciones. El descubrimiento en análisis de datos trata de responder a la pregunta de qué pasó que no conocemos y el resultado es la comprensión de lo que pasó. Lo que lo diferencia de los tipos anteriores de análisis es utilizar los datos para detectar algo nuevo, novedoso o diferente (por ejemplo, tendencias, excepciones o conglomerados) en lugar de describirlo o explicarlo.

4. Análisis Predictivo. Para las dos últimas categorías de análisis, pasamos de la retrospectiva y la perspicacia a la previsión. El análisis predictivo intenta responder a la pregunta: "¿Qué es lo que puede pasar?". Utiliza datos y conocimientos del pasado para predecir resultados futuros y proporciona métodos para evaluar la calidad de estas predicciones

5. Análisis Prescriptivo. Examina la cuestión de qué debo hacer con respecto a lo que ha ocurrido o es probable que ocurra. Permite a los responsables de la toma de decisiones no sólo mirar hacia el futuro sobre las oportunidades (y temas) que existen potencialmente, sino que también presenta el mejor curso de acción para actuar sobre la prospectiva de manera oportuna con la consideración de la incertidumbre. Esta forma de capacidad analítica va estrechamente unida a la optimización, respondiendo a preguntas de tipo "qué pasaría si" para evaluar y presentar la mejor solución.

2.2.3 Tipos específicos de análisis

Una vez examinadas las capacidades analíticas que ayudan a definir los objetivos de la analítica, nos fijamos en la analítica específica que puede guiar a las partes que intervienen en el despliegue de la analítica en las aplicaciones IoT.

- Analítica visual. La analítica visual combina visualizaciones interactivas con técnicas de análisis de datos "para una comprensión, razonamiento y toma de decisiones efectivas sobre la base de conjuntos de datos muy grandes y complejos". Por lo tanto, el análisis visual puede contribuir no sólo a describir y diagnosticar lo que sucedió, sino también a ayudar a los usuarios a descubrir nuevas percepciones. En el trabajo de Zhang et al., vemos cómo se aplica la analítica visual a los datos sanitarios y se describe, mediante la respuesta a preguntas como "¿Cuál es la distribución de la edad de gestación?
- Minería de datos. La minería de datos es parte del proceso de descubrimiento de conocimientos a partir de datos (KDD) en el que se descubren patrones y conocimientos interesantes a partir de grandes cantidades de datos. El IoT es una fuente de una gran cantidad de datos en la que se pueden aplicar las técnicas de minería de datos.

Éstos incluyen:

El resumen de datos multidimensionales se asocia a menudo con operaciones de procesamiento analítico en línea (OLAP) que utilizan el conocimiento de fondo del dominio para permitir la presentación de datos en diferentes niveles de abstracción. Por ejemplo, se pueden desglosar y transferir datos para presentarlos en diferentes grados de integración.

Asociación y correlación es el proceso de encontrar la relación entre dos variables que varían de acuerdo con algún patrón. Esto podría permitirnos averiguar si la compra del producto A condujo a la compra del producto B con cierto grado de confianza y apoyo.

La clasificación es el proceso de encontrar algún modelo o función que tenga la capacidad de distinguir entre clases de datos o conceptos.

El agrupamiento es el proceso de agrupar objetos de datos en clases sin etiquetas. Los objetos de datos agrupados tienen una similitud máxima con los objetos de la clase y una similitud mínima entre los objetos de otras clases.

El descubrimiento de patrones es el proceso de detectar y extraer patrones interesantes de los datos, un ejemplo de los cuales son los conjuntos de elementos frecuentes, un conjunto de elementos que a menudo aparecen juntos en un conjunto de datos transaccionales. La detección de anomalías se refiere al problema de "encontrar patrones en los datos que no se ajustan al comportamiento esperado".

- Análisis de contenido y texto. La analítica de contenidos es el área más amplia en la que se aplican las técnicas analíticas a los contenidos digitales. El análisis de texto es la derivación de información de alta calidad a partir de texto no estructurado, por ejemplo, extrayendo entidades y relaciones nombradas, analizando opiniones, extrayendo información de eventos y series temporales, etc.
- Análisis de vídeo. Video Analytics (VA) trata sobre el uso de software y hardware especializado "para analizar el vídeo capturado e identificar automáticamente objetos, eventos, comportamientos o actitudes específicas en las secuencias de vídeo en tiempo real".
- Análisis de tendencias. El análisis de tendencias se ocupa de examinar datos y eventos a través del tiempo, comprenderlos y hacer predicciones de tendencias futuras y proporcionar sistemas de alerta temprana. El análisis de tendencias también está estrechamente relacionado con el análisis de la información de las series temporales, en las que al observar una serie temporal intentamos encontrar un "cambio a largo plazo en el nivel medio".
- Análisis de negocios. Business Analytics es la práctica de utilizar los datos de una organización para obtener información a través de técnicas analíticas que pueden informar mejor las decisiones de negocio y automatizar y optimizar los procesos de negocio.

2.2.4 La infraestructura para la analítica en IoT

La infraestructura que permite el análisis en IoT son los componentes, las técnicas y la tecnología que contribuyen al proceso de utilización de los datos en las aplicaciones analíticas. La Figura 4 muestra el proceso de cómo los datos pasan por los pasos de generación y recolección, agregación e integración y finalmente se aplican en aplicaciones analíticas. El almacenamiento y el cálculo son procesos abstractos implicados en cada paso de este flujo de datos. En la práctica, los datos podrían canalizarse de una etapa a otra, por lo que no es necesario almacenarlos físicamente en

un lugar separado. El cálculo también puede realizarse en el dispositivo o en tránsito y no es necesario que implique un componente de cálculo separado.

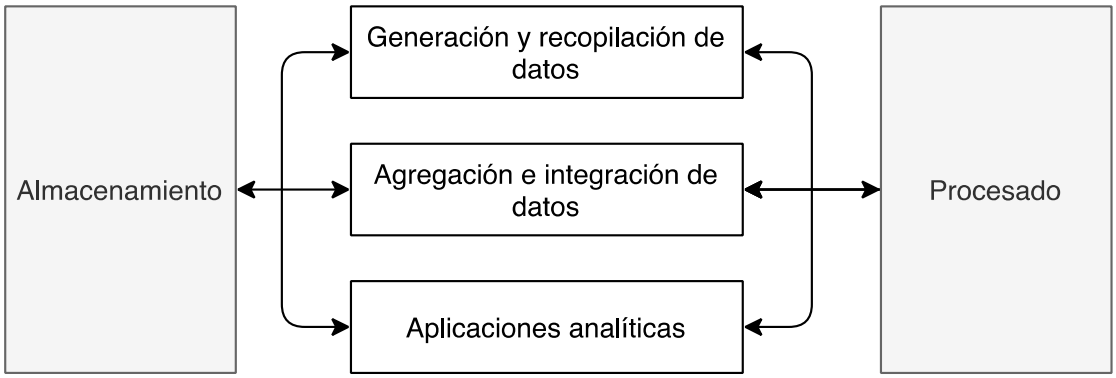


Figura 4 Generación de datos: sensores y etiquetas, hardware y sistema operativo, energía.

Una fuente importante de datos del IoT se genera a partir de dispositivos que incluyen muchos tipos de sensores ambientales, espaciales y de salud [18]. Las etiquetas (tags) también generan datos y pueden ser pasivas, como los códigos QR y los patrones de códigos de barras, que requieren que un dispositivo escanee o esté activo, como las tecnologías iBeacon [19] y UriBeacon [20], que proyectan señales a aplicaciones móviles. Las etiquetas RFID [21] pueden ser pasivas o activas, el tipo activo que requiere una fuente de alimentación para transmitir señales, y pueden ser UHF (Ultra High Frequency), HF (High Frequency), o LF (Low Frequency).

Los sensores IoT desplegados a distancia también requieren energía, especialmente para el proceso de transmisión inalámbrica de datos, que consume energía. Wolf [22]describe una serie de sistemas de recuperación de energía que aprovechan la energía del medio ambiente, mientras que las tecnologías de carga inalámbrica como el ubeam [23] y la carga por movimiento como el Ampy [24] son alternativas.

2.3 K- Nearest Neighbors (KNN)

K-Nearest Neighbor(KNN) es un algoritmo de Machine Learning muy simple, fácil de entender y versátil. Es idóneo para la clasificación de los datos debido a su alta precisión, teniendo en cuenta su simpleza. No es paramétrico, es decir, no hace suposiciones explicitas sobre la forma funcional de los datos. Por último, se caracteriza por ser insensible a valores atípicos [16].

En el reconocimiento de patrones, el algoritmo KNN es un método no paramétrico utilizado para la clasificación y regresión. En ambos casos, la entrada consiste en los ejemplos de entrenamiento más cercanos a k en el espacio de características. La salida depende de si se utiliza KNN para la clasificación o la regresión:

En la clasificación KNN, el resultado es una pertenencia a una clase. Un objeto es clasificado por un voto de pluralidad de sus vecinos, siendo el objeto asignado a la clase más común entre sus vecinos más cercanos (k es un entero positivo, típicamente pequeño). Si k = 1, entonces el objeto simplemente se asigna a la clase de ese vecino más cercano.

En la regresión KNN, la salida es el valor de propiedad del objeto. Este valor es el promedio de los valores de k de los vecinos más cercanos.

KNN es un tipo de aprendizaje basado en instancias, o aprendizaje perezoso, donde la función sólo se aproxima localmente y toda la computación se aplaza hasta la clasificación.

Tanto para la clasificación como para la regresión, una técnica útil puede ser asignar pesos a las contribuciones de los vecinos, de modo que los vecinos más cercanos contribuyan más al promedio que los más distantes. Por ejemplo, un esquema de ponderación común consiste en dar a cada vecino un peso de 1/d, donde d es la distancia al vecino.

Los vecinos se toman de un conjunto de objetos para los que se conoce la clase (para la clasificación KNN) o el valor de propiedad del objeto (para la regresión KNN). Esto se puede considerar como el conjunto de entrenamiento para el algoritmo, aunque no se requiere ningún paso de entrenamiento explícito. Una peculiaridad del algoritmo KNN es que es sensible a la estructura local de los datos.

Los ejemplos de formación son vectores en un espacio de características multidimensionales, cada uno con una etiqueta de clase. La fase de entrenamiento del algoritmo consiste únicamente en almacenar los vectores de característica y las etiquetas de clase de las muestras de entrenamiento.

En la fase de clasificación, k es una constante definida por el usuario, y un vector no identificado (una consulta o punto de prueba) se clasifica asignando la etiqueta más frecuente entre las muestras de entrenamiento k más cercanas a ese punto de consulta.

Una métrica de distancia comúnmente utilizada para las variables continuas es la distancia euclídea. Para las variables discretas, como para la clasificación de texto, se puede utilizar otra métrica, como la métrica de superposición (o distancia de Hamming).

A menudo, la precisión de clasificación de KNN puede mejorarse significativamente si la métrica de distancia se aprende con algoritmos especializados como el análisis de componentes de Large Margin Nearest Neighbor o Neighbourhood.

Una desventaja de la clasificación básica de "voto por mayoría" se produce cuando la distribución de clases es sesgada. Es decir, los ejemplos de una clase más frecuente tienden a dominar la predicción del nuevo ejemplo, porque tienden a ser comunes entre los vecinos más cercanos debido a su gran número [17]. Una forma de superar este problema es ponderar la clasificación, teniendo en cuenta la distancia desde el punto de prueba hasta cada uno de sus vecinos más cercanos. La clase (o valor, en problemas de regresión) de cada uno de los puntos k más cercanos se multiplica por un peso proporcional a la distancia inversa entre ese punto y el punto de prueba.

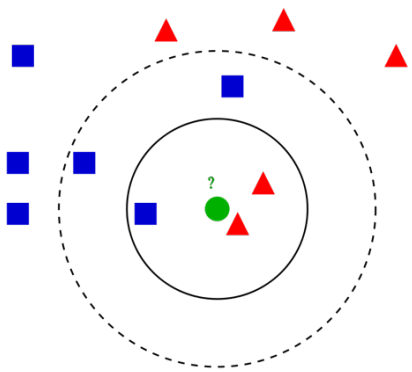


Figura 5 Ejemplo de clasificación k-NN. La muestra de prueba (punto verde) debe clasificarse en cuadrados azules o triángulos rojos. Si k = 3 (círculo de línea continua) se asigna a los triángulos rojos porque hay 2 triángulos y solo 1 cuadrado dentro del círculo [18].

La mejor elección de k depende de los datos; en general, los valores mayores de k reducen el efecto del ruido en la clasificación, pero hacen que los límites entre las clases sean menos distintivos. Una buena k puede ser seleccionada por varias técnicas heurísticas (ver optimización de hiperparámetros). El caso especial en el que se predice que la clase es la clase de la muestra de entrenamiento más cercana (es decir, cuando k = 1) se llama el algoritmo vecino más cercano.

La precisión del algoritmo k-NN puede verse gravemente degradada por la presencia de características ruidosas o irrelevantes, o si las escalas de características no son coherentes con su importancia. En los problemas de clasificación binaria (dos clases), es útil elegir k para que sea un número impar, ya que esto evita los votos empatados. Una forma popular de elegir la k empíricamente óptima en esta configuración es a través del método bootstrap [28].

El clasificador de tipo de vecino más cercano más intuitivo es el clasificador de vecino más cercano que asigna un punto x a la clase de su vecino más cercano en el espacio de características, es decir $C_n^{1\,nn}(x) = Y_{(1)}$.

A medida que el tamaño del conjunto de datos de entrenamiento se acerca al infinito, el clasificador vecino más cercano garantiza una tasa de error no peor que el doble de la tasa de error de Bayes [28] (la tasa de error mínima alcanzable dada la distribución de los datos).

La versión ingenua del algoritmo es fácil de implementar al calcular las distancias desde el ejemplo de prueba a todos los ejemplos almacenados, pero es computacionalmente intensiva para grandes conjuntos de entrenamiento. El uso de un algoritmo de búsqueda de vecino más cercano aproximado hace que KNN sea manejable computacionalmente incluso para grandes conjuntos de datos. Se han propuesto muchos algoritmos de búsqueda de vecinos más cercanos a lo largo de los años; estos generalmente buscan reducir el número de evaluaciones de distancia realmente realizadas.

KNN tiene algunos resultados de consistencia fuerte. A medida que la cantidad de datos se aproxima al infinito, el algoritmo KNN de dos clases garantiza una tasa de error no peor que el doble de la tasa de error de Bayes (la tasa de error mínima alcanzable dada la distribución de los datos).

2.4 Support-vector machine (SVM)

En este trabajo, las maquinas de vectores de soporte se emplearán para la predicción de datos. Se ha optado por SVM debido a su gran ventaja en términos de complejidad respecto a otros algoritmos. Se caracteriza por tener un mejor rendimiento en comparación con la regresión logística. Además de ser computacionalmente más barato, es rápido. Son características importantes para tener en cuenta a la hora de la implementación del sistema de reducción de dimensión que se explicará en el apartado 4.3.

En el aprendizaje automático, las máquinas de vectores de soporte (SVM, también redes de vectores de soporte) son modelos de aprendizaje supervisados con algoritmos de aprendizaje asociados que analizan los datos utilizados para el análisis de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento, cada uno marcado como perteneciente a una u otra de dos categorías, un algoritmo de entrenamiento SVM construye un modelo que asigna nuevos ejemplos a una categoría u otra, convirtiéndolo en un clasificador lineal binario no probabilístico. Un modelo SVM es una representación de los ejemplos como puntos en el espacio, mapeados para que los ejemplos de las categorías separadas se dividan por un espacio claro que sea lo más amplio posible. Los

nuevos ejemplos se asignan a ese mismo espacio y se predice que pertenecen a una categoría basada en el lado de la brecha en la que caen.

Cuando los datos no están etiquetados, el aprendizaje supervisado no es posible, y se requiere un enfoque de aprendizaje no supervisado, que intente encontrar la agrupación natural de los datos en grupos, y luego asignar nuevos datos a estos grupos formados. El algoritmo de agrupación de vectores de soporte, creado por Hava Siegelmann y Vladimir Vapnik [19], aplica las estadísticas de los vectores de soporte, desarrollados en el algoritmo de máquinas de vectores de soporte, para clasificar datos no etiquetados. Es uno de los algoritmos de agrupación más utilizados para aplicaciones industriales.

La clasificación de datos es una tarea común en el aprendizaje automático. Supongamos que unos puntos de datos pertenecen cada uno a una de las dos clases, y el objetivo es decidir en qué clase estará un nuevo punto de datos. En el caso de las máquinas de vectores de soporte, un punto de datos se ve como un vector p – dimensional (una lista de p números), y queremos saber si podemos separar esos puntos con un [(p - 1) - hiperplano dimensional]. Esto se llama un clasificador lineal. Hay muchos hiperplanos que pueden clasificar los datos. Una opción razonable como el mejor hiperplano es la que representa la mayor separación, o margen, entre las dos clases. Por lo tanto, elegimos el hiperplano para maximizar la distancia desde el punto de datos más cercano a cada lado. Si existe tal hiperplano, se conoce como el hiperplano de margen máximo y el clasificador lineal que se define se conoce como clasificador de margen máximo [20].

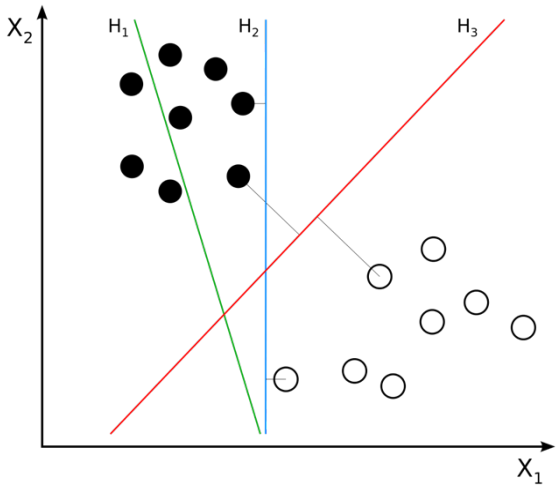


Figura 6 H1 no separa las clases. H2 sí, pero solo con un pequeño margen. H3 los separa con el margen máximo [21].

Más formalmente, una máquina de vectores de soporte construye un hiperplano o un conjunto de hiperplanos en un espacio de dimensión alta o infinita, que puede usarse para clasificación, regresión u otras tareas como detección de valores atípicos. Intuitivamente, se logra una buena separación mediante el hiperplano que tiene la mayor distancia al punto de datos de entrenamiento más cercano de cualquier clase (denominado margen funcional), ya que en general cuanto mayor es el margen, menor es el error de generalización del clasificador.

Mientras que el problema original puede plantearse en un espacio de dimensión finita, a menudo sucede que los conjuntos a discriminar no son linealmente separables en ese espacio. Por esta razón, se propuso que el espacio original de dimensiones finitas se mapee en un espacio de dimensiones mucho más altas, presumiblemente haciendo la separación más fácil en ese espacio. Para mantener la carga computacional razonable, los

mapeos utilizados por los esquemas SVM están diseñados para garantizar que los productos de puntos de pares de vectores de datos de entrada se puedan calcular fácilmente en términos de las variables en el espacio original, definiéndolos en términos de una función de núcleo k (x, y) seleccionado para adaptarse al problema.

Las SVM son útiles en la categorización de texto e hipertexto, ya que su aplicación puede reducir significativamente la necesidad de instancias de entrenamiento etiquetadas tanto en la configuración inductiva estándar como en la transductiva. Algunos métodos para el análisis semántico superficial se basan en máquinas de vectores de soporte.

La clasificación de imágenes también se puede realizar utilizando SVM. Los resultados experimentales muestran que los SVM logran una precisión de búsqueda significativamente mayor que los esquemas tradicionales de refinamiento de consultas después de solo tres o cuatro rondas de comentarios relevantes. Esto también es cierto para los sistemas de segmentación de imágenes, incluidos los que usan una versión SVM modificada que usa el enfoque privilegiado como lo sugiere Vapnik.

Capítulo 3: Sistema de Gestión de la Disponibilidad de la Infraestructura de SmartSantander

En este capítulo se presenta el procedimiento y enfoque con el que se ha abordado la solución para conocer en todo momento la disponibilidad de los sensores en la plataforma SmartSantander.

3.1 Descripción y arquitectura del sistema

Uno de los aspectos fundamentales en la gestión de una infraestructura IoT es asegurarse de la integridad del sistema y el funcionamiento de los sensores que están desplegados en el terreno. Para ello, hay que comprobar que todos los dispositivos que conforman la infraestructura están conectados y en estado funcional. Para llevar a cabo dicha comprobación, el Sistema de Gestión de la Disponibilidad (SGD) que se ha implementado en este TFG hace uso del API de la plataforma SmartSantander. A través de este API, el SGD se subscribe a todos los sensores desplegados de forma que cada vez que un sensor genera una nueva medida, la plataforma de SmartSantander se la notificará. El SGD además de procesar estas medidas de manera online, las almacena en una base de datos para su procesado offline.

Una vez que se dispone de permisos de seguridad, se puede hacer una conexión con el API de la plataforma. Mediante la conexión a la plataforma, se puede crear una suscripción a todos los sensores de la infraestructura. Al crear una suscripción, se asigna un identificador a cada dispositivo que hace peticiones de información en el API. Dicho identificador forma parte de una secuencia arbitraria de letras y números.

Cuando se inicia el SGD se comprueba la existencia del identificador, que tiene que estar guardado en una colección de la base de datos creada para este propósito. El API de la plataforma SmartSantander asigna un identificador distinto cada vez que se crea una suscripción para la petición de datos. El motivo de guardar la identificación de la suscripción es para poder reanudar la suscripción ya creada y evitar crear una nueva. Sabiendo que las suscripciones al API caducan cada 7 días, se puede afirmar que no tendría un impacto grave en la plataforma, pero simplifica las cosas a la hora de comprobar si la suscripción ha caducado o si se quiere consultar el API a cerca de que dispositivos reciben información. Para comprobar la validez de la suscripción hay que crear una petición para dicho propósito en la que indicar el identificador que se utiliza. Para este propósito se hace uso de la base de datos para no perder dicha información si por algún motivo se para el SGD. Ante una situación de pruebas, se puede predecir el hecho de hacer conexiones iterativas a la plataforma, lo que implica la creación de un identificador distinto cada vez que se establece la conexión. Para evitar dicha situación,

como se ha comentado con anterioridad, se guardará el identificador y se comprobará si la suscripción al API sigue válida cada 6 días.

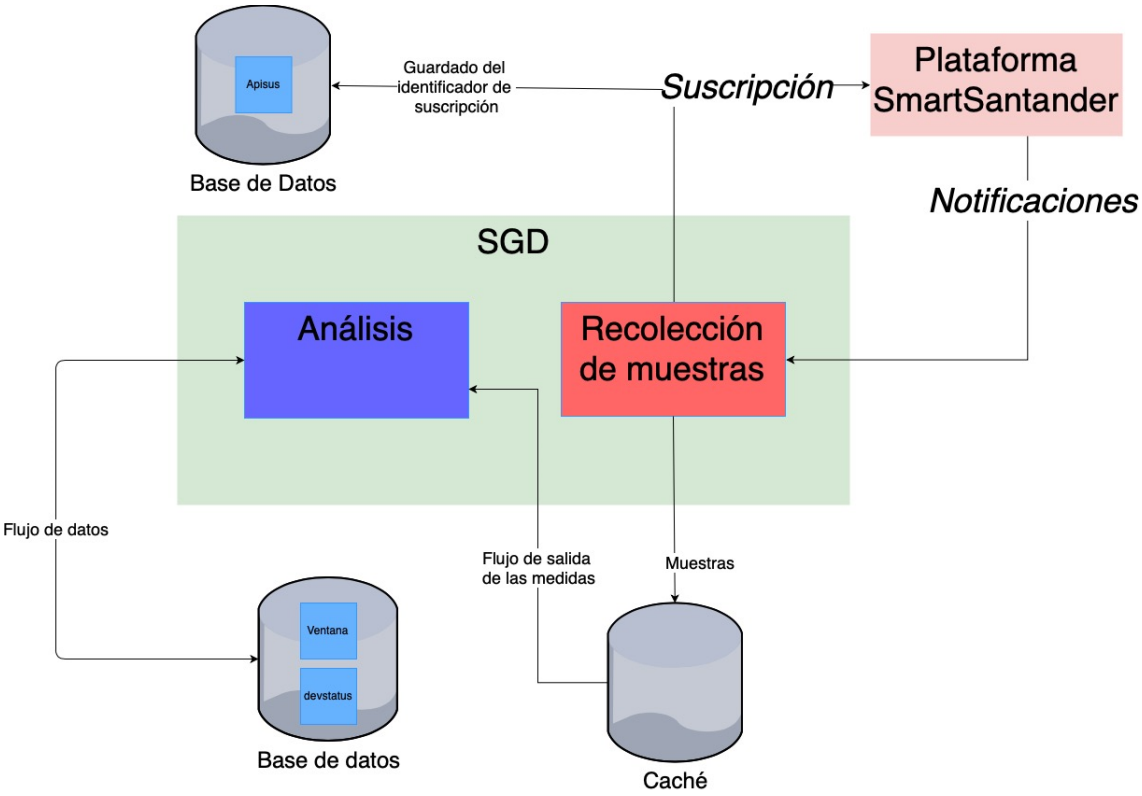


Figura 7 ARQUITECTURA DEL SISTEMA

El siguiente paso consiste en la petición de todas las medidas que los sensores envían al estar en funcionamiento. Los sensores son pequeños dispositivos capaces de detectar acciones o estímulos externos y responder en consecuencia. Es decir, son dispositivos que se encargan de medir las magnitudes físicas y transformarlas en señales eléctricas. Estos dispositivos estarán conectados a un nodo de la infraestructura.

Un nodo es un dispositivo que forma parte de la red de SmartSantander. Su propósito es el reenvío de las muestras recibidas de cada uno de los sensores que puede tener conectados al mismo. La conexión de los sensores con el nodo puede ser inalámbrica o mediante cable tal como se muestra en la Figura 10. Los nodos reenvían las muestras al API de la plataforma a través del gateway. Los gateways son dispositivos que gestionan a los nodos de la plataforma.

Es importante destacar que para la gestión de la disponibilidad de la infraestructura solo nos vamos a centrar en los metadatos. Los metadatos es toda información descriptiva que comprende una muestra enviada de un sensor (por ejemplo, la fecha de envío de una muestra, la fecha de creación de un archivo, la magnitud física que mide un sensor). Los metadatos de las medidas recibidas serán utilizados más adelante para los cálculos de frecuencia de envió, disponibilidad y varianza de cada dispositivo.

La nube (cloud) u ordenador central, como lo indica el nombre, es un ordenador que se encarga de recibir los datos reenviados por los nodos, procesarlos, guardarlos y ofrecer una disponibilidad de esta información mediante una conexión a internet. Ante la ausencia de cualquier gestión y solo mero reenvío de los datos una vez recibidos por el API, hay un gran riesgo de recibir la información a ráfagas. Esto es un problema debido a que el SGD comprende un procesado online que tiene su coste temporal. En

consecuencia, hay pérdida de información gracias al procesado online del que dispone. Como solución, se ha optado por la implementación de una base de datos NoSQL como caché. Una caché en el SGD es una componente software destinada para el almacenamiento de las muestras recibidas del API para su lectura y procesado en el futuro.

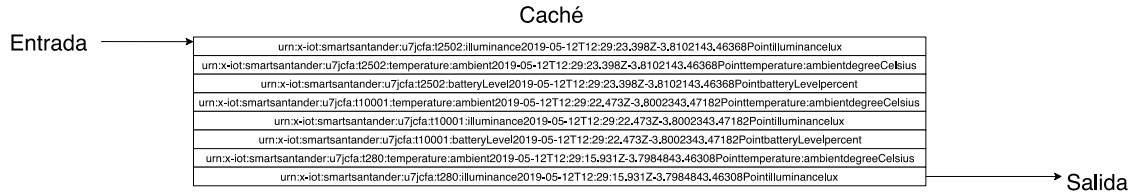


Figura 8 Flujo de datos en la caché.

Como se muestra en la Figura 8, el flujo de los datos en la caché sigue una lógica FIFO (First In, First Out). Es decir, de la caché se solicitará siempre la información que lleva más tiempo guardada en la misma o la información que entre primero. Esto evita la perdida de datos independientemente de la velocidad del flujo de información recibida.

Para el análisis y cálculo de la disponibilidad de cada dispositivo habilitado, se necesitan dos metadatos de cada muestra: identificador y timestamp. Un timestamp es una secuencia de caracteres o información codificada que identifica cuándo ocurrió un determinado evento, por lo general dando la fecha y la hora del día, a veces con una precisión de una pequeña fracción de segundo. El término se deriva de los sellos de goma utilizados en las oficinas para estampar la fecha actual y, a veces, la hora, en tinta en documentos en papel, para registrar cuándo se recibió el documento. Ejemplos comunes de este tipo de marca de tiempo son un matasellos en una letra o los tiempos de "entrada" y "salida" en una tarjeta de hora.

Al recibir una ráfaga de muestras, como primer paso se guardan en la caché. A continuación, por el orden de guardado, se retiran los datos y se procesan teniendo en cuenta el identificador de cada muestra. Las muestras de los sensores no tienen un identificador como tal. El identificador del que disponen pertenece al nodo al que están conectados dado que son los nodos los que envían la información por la red, no los sensores. Pero aún así es posible identificar cada muestra con su respectivo sensor de cada nodo. Al extraer los datos de la caché, podemos unir la dirección de cada nodo con la magnitud física del sensor. Esta información es comprendida en cada muestra que se recibe.

Por ejemplo:

Identificador del nodo 1: urn:x-iot:smartsantander:u7jcfa:f3004

Magnitud física del sensor 1 : temperature:ambient

Identificador de sensor 1: urn:x-iot:smartsantander:u7jcfa:f3004:temperature:ambient

Identificador de sensor 2: urn:x-iot:smartsantander:u7jcfa:f3004:relativeHumidity

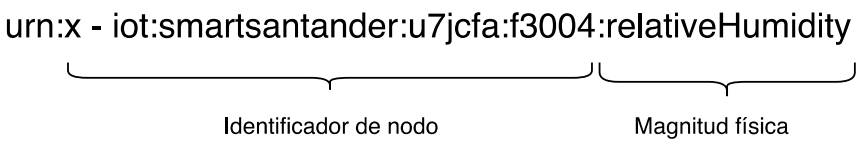


Figura 9 Explicación de la creación del identificador de un sensor

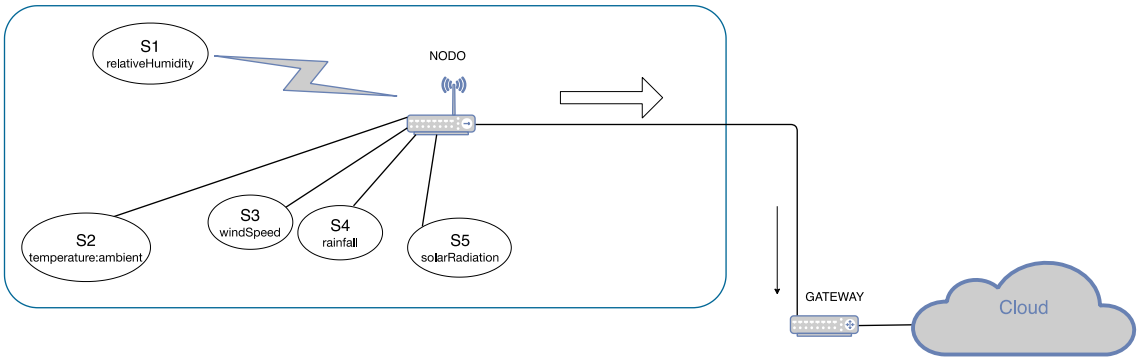


Figura 10 ARQUITECTURA SMARTSANTANDER

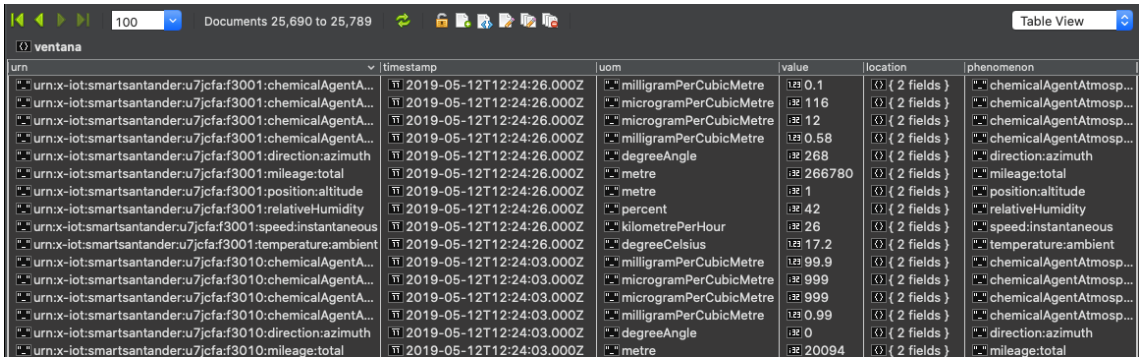


Figura 11Colección Ventana de la base de datos

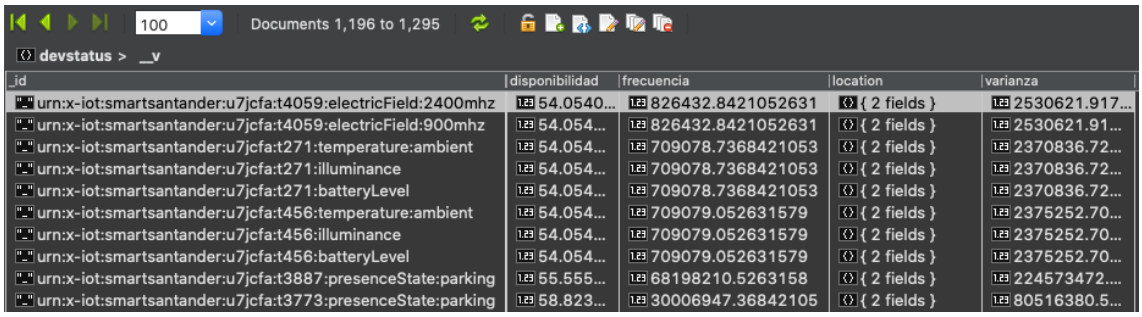


Figura 12 Colección devstatus de la base de datos

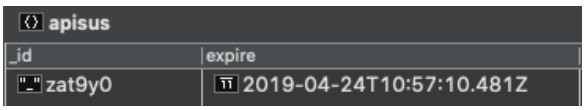


Figura 13 Colección apisus de la base de datos

SGD hará uso principalmente de dos colecciones en la base de datos. La colección "Ventana" esta pensada para guardar veinte muestras de cada identificador de sensor. Por cada muestra que se procese se guardarán los valores como la identificación, localización, medida, marca temporal, etc. Por lo tanto, aunque esta implementación solo se use para gestión de disponibilidad, los datos guardados se podrían utilizar para otros propósitos.

La estrategia de guardar la información de cada sensor por separado y de almacenar una serie de veinte muestras sirve para estudiar la secuencia de dichos envíos y poder ofrecer una respuesta relativamente válida a cerca de su disponibilidad y frecuencia de trabajo. Cabe destacar de que el valor de la dimensión de la ventana se puede cambiar ampliando el numero de muestras guardadas en la colección "Ventana" antes de calcular una respuesta. La dimensión de la ventana dependerá del marco temporal con el que se quiera trabajar. Por ejemplo, si queremos calcular el porcentaje de disponibilidad por hora, se necesitaría aumentar el tamaño de la ventana a un numero que comprenda las muestras de ese tiempo. En consecuencia, el SGD estaría recolectando datos durante una hora para hacer los cálculos. He aquí uno de los motivos por los que se ha decidido por una ventana de veinte muestras por sensor.

Al llenar la ventana de un sensor con N muestras, la siguiente muestra que se recibe del mismo sensor, acciona el calculo de una respuesta respecto a la ventana, antes de su modificación. La respuesta será el porcentaje de disponibilidad, frecuencia de envío de las muestras y la varianza del tiempo.

De manera genérica, teniendo en cuenta que disponemos de N muestras por sensor. El tiempo medio de envío entre dos envíos consecutivos, o dicho de otro modo, la periodicidad media de cada sensor será:

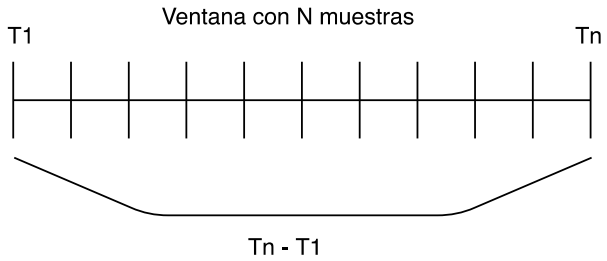


Figura 14 Ejemplo de una ventana de datos con N muestras

$$T_{med} = \frac{N}{T_N - T_1}$$

Donde T_N es el timestamp de n-ésima muestra y T_1 el de la primera.

A su vez, la varianza se calcularía de la siguiente manera:

$$Var\ (X) = \frac{\sum_1^n (xi - \bar{X})^2}{n}$$

n = numero de muestras -1
xi = diferencia tiempo entre muestras
$\bar{X}$ = tiempo medio de envío

Como ultimo parámetro, se calcula la disponibilidad del sensor según el algoritmo que se representa a continuación:

```
 let perd = 0;
   for (let i = 1; i < Numero_de_muestras; i++) {
     const diferencia_tiempo = tiempo_muestra[i] - tiempo_muestra[i - 1];
     if (tiempo_medio < diferencia_tiempo) {
       let ntram = Math.floor(diferencia_tiempo / tiempo_medio); // ntram es
el número de muestras perdidas
       perd += ntram; // muestras perdidas en total
     }
 Disponibilidad = (Numero_de_muestras / (perd + Numero_de_muestras)) * 100;
```

Básicamente, se calcula la diferencia de tiempo que hay entre una muestra y su precedente en la ventana de un sensor. Eso es (T(i+1) – Ti). Si el tiempo medio de envío de muestras es inferior al tiempo calculado entre una muestra y su precedente de la misma ventana, se puede afirmar que el sensor no trabaja de manera normal, por lo tanto tendría una disponibilidad inferior al 100%. En la condición de que el tiempo medio de la ventana sea mayor o igual a la diferencia de tiempos que hay entre una muestra y su precedente, el sensor tendría una disponibilidad del 100%.

Figura 15 Calculo de la diferencia de tiempo entre muestras.

Una vez calculados los valores de frecuencia de envío (de las muestras), varianza y disponibilidad del dispositivo, la información se almacenará en la segunda colección de la base de datos bajo el nombre “devstatus”. Además de la información calculada previamente, se guardan datos de posición, es decir, latitud y longitud. Estos datos pueden ser útiles a la hora de intentar actuar ante un incidente de baja disponibilidad. La base de datos se podría utilizar para mostrar la posición de los sensores en un mapa junto con una etiqueta, informando del estado de cada uno. Por ultimo, se actualiza la ventana del sensor siguiendo la lógica FIFO. Esto es, se hace avanzar todas las entradas de la colección eliminando la más antigua.

El SGD se encarga de proporcionar información acerca de dispositivos que ya han enviado datos alguna vez en la red. Esto implica que a no ser que se tenga una lista con cada sensor que en teoría deben funcionar en la red, no se va a poder saber si un sensor esta dañado si nunca ha enviado datos.

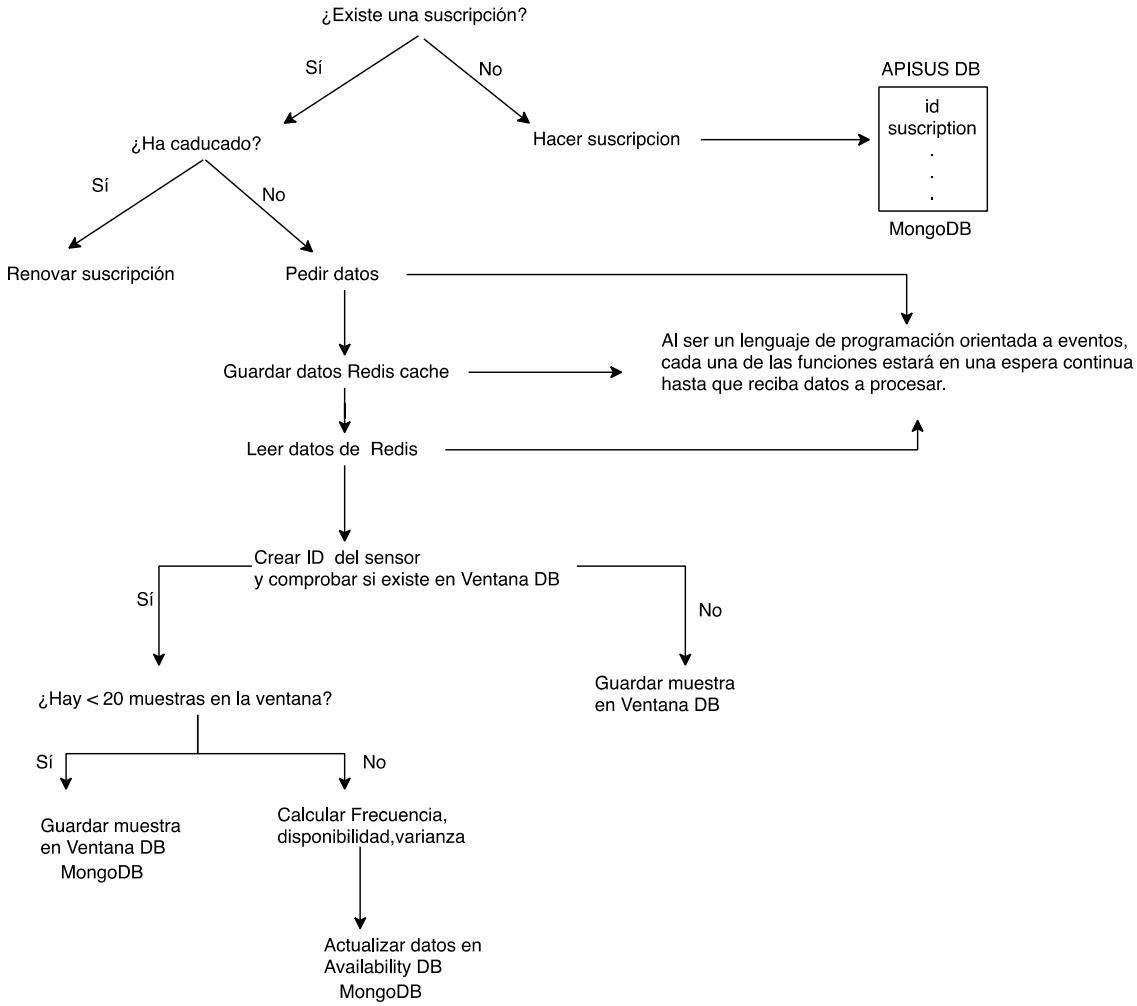


Figura 16 Planificación de funcionamiento de SGD.

3.2 Implementación del sistema

Para la implementación del SGD, se ha utilizado el entorno Node.js, las bases de datos Redis y Mongodb, así como un editor de código fuente, Visual Studio Code.

Node.js es un entorno JavaScript del lado del servidor, basado en eventos. Es idóneo para la implementación de la aplicación al ser asíncrono. A diferencia de un código síncrono, este no espera a las instrucciones diferidas y continúa con su ejecución. Tal y como se ha comentado, el SGD, tras suscribirse a todos los sensores, comenzará a recibir las notificaciones con cada nueva medida. Este comportamiento asíncrono hace que Node.js se adecúe perfectamente a las necesidades del SGD. Cuando Node.js necesita realizar una operación de E / S, como leer desde la red, acceder a una base de datos o al sistema de archivos, en lugar de bloquear el hilo y desperdiciar los ciclos de la CPU en espera, continúa con su operativa. Cuando se completa la operación. La lógica del sistema también continúa. Esto permite que maneje miles de conexiones concurrentes con un solo servidor sin introducir la carga de administrar la concurrencia de subprocesos, lo que podría ser una fuente importante de errores.

Por su parte, Redis es un almacén de estructura de datos en memoria, de código abierto, que se utiliza como agente de base de datos, caché y mensaje. En este caso se va a utilizar solo como caché. Por último, MongoDB es una base de datos orientada a documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON. Una de las diferencias más importantes con respecto a las bases de datos

relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.

3.3 Integración y pruebas del sistema

La integración del SGD se hizo en una maquina virtual con un entorno Linux. Tras instalar tanto las librerías Node.js que utiliza el SGD como las bases de datos Redis y MongoDB, se lanzó el SGD mediante el administrador de procesos PM2 de forma que en caso de caída accidental del sistema, este se reiniciase automáticamente.

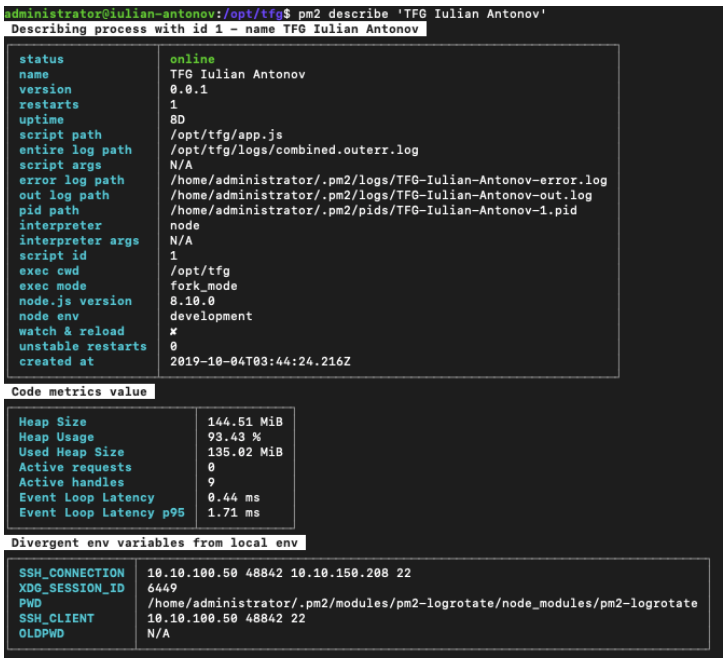


Figura 17 Descripción del estado de funcionamiento del SGD en PM2.

Como se ha mencionado anteriormente, los resultados de este sistema de gestión podrían utilizarse para detectar los sensores que han perdido carga, han dejado de funcionar o están perdiendo datos. Este sistema al recoger varias muestras de un sensor antes de ofrecer una respuesta como la disponibilidad de este, tiene la ventaja de que se podría arrancar independientemente del día o la hora. A la vez, esto implica que no se puede saber desde un principio cuantos sensores están conectados a la infraestructura si no han enviado datos. Por lo tanto, en la situación de que una vez arrancado el sistema de gestión haya sensores que estén dañados y por lo tanto no envíen datos, el sistema no podría detectarlo. Esto se podría solventar si se dispusiera de un listado de todos los nodos instalados, de forma que también se pudieran detectar los que no envían medidas.

Capítulo 4: Sistema de Eliminación de Outliers y Reducción de la Dimensionalidad

En este capítulo se van a explicar los procedimientos que se han aplicado para el análisis de datos en una estructura IoT. El análisis de este sistema está enfocado en la eliminación de outliers. Se basa en un estudio de los valores medidos por los sensores de la infraestructura y un procesado de estos para la reducción de errores y trafico en la red. Este enfoque podría proporcionar una menor latencia en la red así como un ahorro de recursos de esta y facilitar así su escalado.

4.1 Descripción y arquitectura del sistema

Antes de comenzar con la explicación de la arquitectura, cabe destacar que el estudio de los outliers que se ha realizado es aplicable a cualquier medida que, en condiciones normales, conlleve un cabio de valores progresivo. Esto implica que su aplicación es adecuada para sensores como los de temperatura, radiación solar, humedad, luminosidad, etc. No sería un enfoque recomendable para sensores cuyos valores de las medidas pueden cambiar notablemente en cuestión de segundos, como por ejemplo, los sensores de viento, aparcamiento, ruido. Por lo tanto, este análisis se va a centrar en el procesamiento de datos de los sensores de temperatura de la plataforma SmartSantander.

Estos dispositivos se caracterizan por tener un margen relativamente amplio de tiempo en cuanto a los cambios de los valores medidos, lo que implica una detección más sencilla de los valores erróneos en el dataset. Los datos que se necesitan para utilizar el sistema serían la identificación de cada sensor, y sus valores medidos en un tiempo determinado.

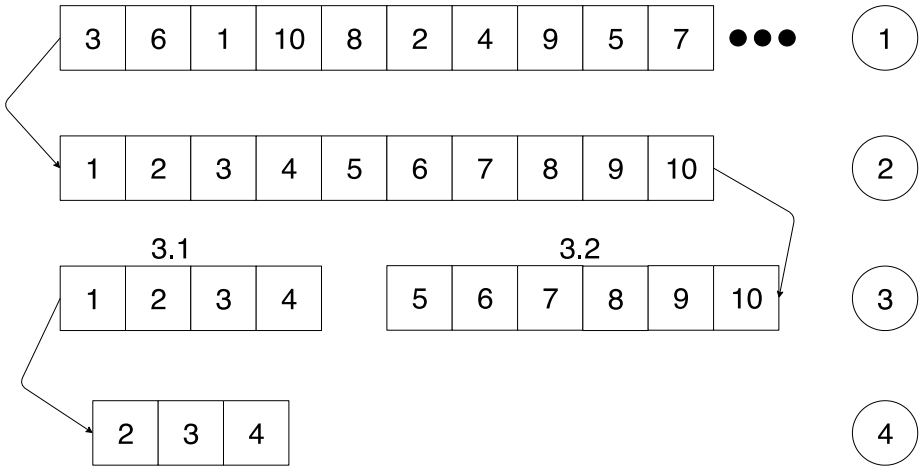


Figura 18 Representación abstracta del procesado de datos.

Para la realización de este análisis disponemos de una base de datos con la información de los identificadores de cada sensor, valores medidos y marca temporal para cada una de las muestras que además están desordenados cronológicamente (punto 1 Figura 18). Como un primer paso, se ordena el dataset cronológicamente (punto 2 Figura 18). A continuación, para tener un punto de referencia en cuanto a la toma de decisión si las muestras tienen valores verdaderos o no, se calcula la gráfica con la media de la variación respecto al tiempo entre muestras en función de los cambios de los valores. Esta grafica nos permite decidir la cantidad de muestras que se debe tener en cuenta para un tiempo determinado. El número de muestras se usará para crear una ventana deslizante (punto 3.1 Figura 18). La ventana deslizante permite tener un punto de referencia para la comparación de las muestras y decisión de resultados (punto 3.2 Figura 18). Según la Figura 19, se van a tener en cuenta 150 muestras, que son las que se envían de media en dos horas. El tamaño de la franja temporal que se considera depende de la variación de las muestras con el tiempo. Básicamente, con esta decisión se está fijando el límite de correlación temporal en esas dos horas aproximadamente.

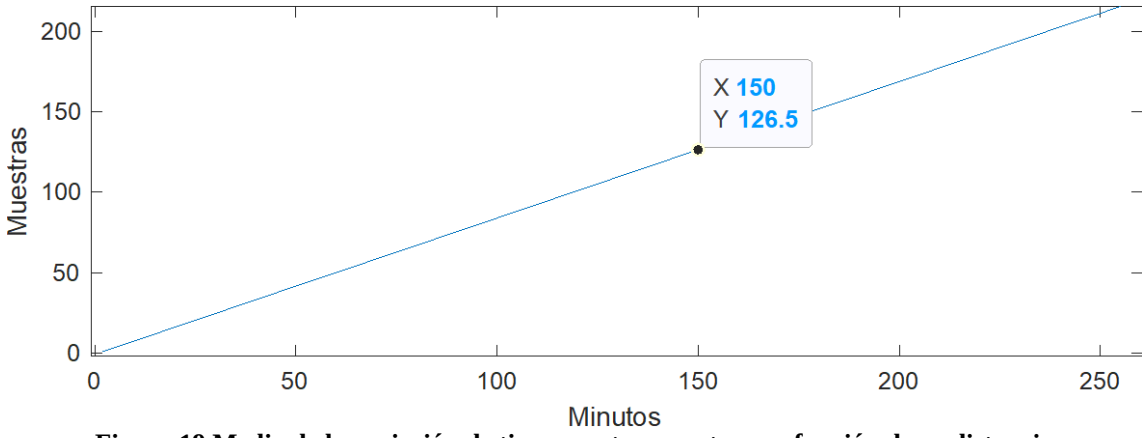


Figura 19 Media de la variación de tiempo entre muestras en función de su distancia

Una vez conocido el tamaño de la ventana, se comienza por crear dicha ventana con las N primeras muestras de nuestro flujo de datos ordenado. Para ese grupo de datos se eliminarán las muestras con un valor 'NaN', mayores que 40 grados centígrados y menores que 0 grados centígrados (punto 4 Figura 18). Estos valores límite de temperatura que se utilizan como primer filtro, se deben considerar en función de las temperaturas promedio en Santander durante el año. Para obtener dicha información se puede hacer un estudio en profundidad si la intención es basarse en valores muy técnicos. En este caso, se pretende asegurar el funcionamiento y la posibilidad de implementación con la condición de usar la menor cantidad de recursos en el procesado.

Uno de los objetivos de la implementación de este sistema es comprobar su funcionamiento así como plantear una posible solución ante una implementación en los nodos en la red. Por lo tanto, utilizar datos muy técnicos respecto al primer filtro de la ventana es poco relevante.

Al filtrar los primeros valores de la ventana nos aseguramos de obtener las primeras muestras para el procesado de datos. Esta primera ventana podría tener outliers entre sus valores que no se detectarían en un principio, pero suponiendo que la dimensión de ésta es relativamente pequeña respecto a la gran cantidad de datos que se van a procesar, la pérdida de la información se podría despreciar. Es más, al utilizar una ventana deslizante, las muestras que se utilizan en un principio irán cambiando, por lo tanto, los valores medios de dicha ventana serán más fiables.

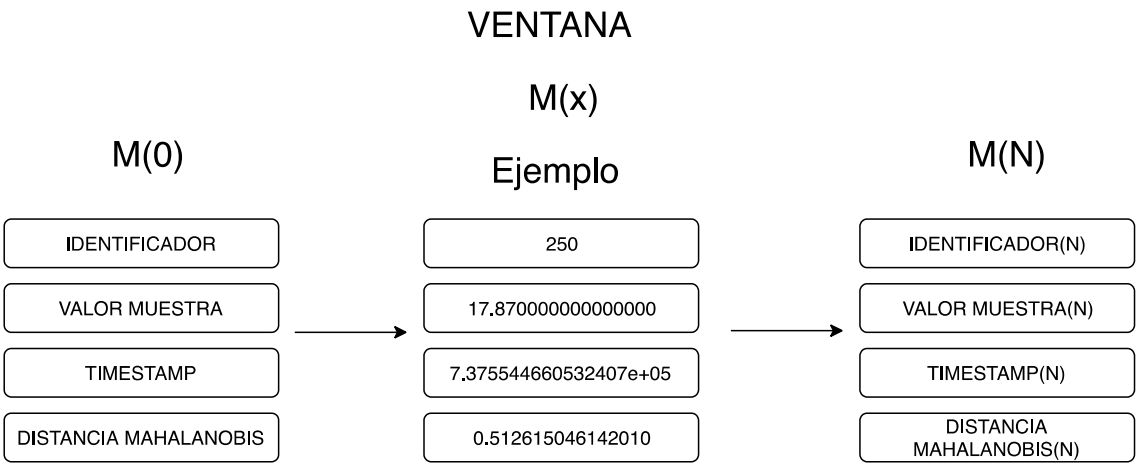


Figura 20 Información que abarca la matriz ventana.

Los resultados del análisis se guardan en una matriz respuesta. En cada columna de dicha matriz se escribirán los resultados de las muestras que se han procesado. Las columnas de la matriz respuesta abarcan la información del identificador, valor corregido, timestamp, clasificación de la muestra y el valor leído de la base de datos.

De esta manera se puede analizar los resultados y comprobar el funcionamiento correcto del proceso.

Figura 21 Representación gráfica del contenido de una matriz resultado.

Si el estado de la muestra es:

- 0 → El valor de la muestra es considerado un valor verdadero, por lo tanto el valor leído concuerda con el valor correcto.

- 1 → El valor de la muestra fue erróneo. El valor corregido se asigna al valor medio de los valores que en ese momento están dentro de la ventana.

- 2 → El valor de la muestra fue un valor 'NaN'. El valor corregido se asigna al valor medio de los que en ese momento están dentro de la ventana.

El análisis consiste en un ciclo de lectura de las muestras y decisión de su estado. Sea este un valor verdadero, un error, o un valor nulo (NaN). En el caso de que se procese un valor nulo (NaN), los datos que se guardan en la matriz respuesta sería la identificación, valor medio de la ventana deslizante, timestamp, clasificación de la muestra y por último el valor que se había leído en el momento. Si el valor leído no es nulo, se procede a los

cálculos de la distancia de Mahalanobis de cada valor en la ventana deslizante respecto del mismo conjunto. Además, se calcula la misma distancia del valor leído con respecto a la ventana deslizante. Esto nos permite comparar las muestras y decidir si el valor leído tiene una distancia menor al conjunto de muestras de la ventana.

Como modo de detección de outliers en el flujo de datos, se utilizará el algoritmo KNN. Según se ha mencionado con anterioridad, para la comparación de los valores y por lo tanto su clasificación, vamos a necesitar una métrica. En este caso se va a utilizar la distancia de Mahalanobis.

La distancia Mahalanobis (MD) [22] es la distancia entre dos puntos en un espacio multidimensional. En un espacio euclídeo regular, las variables (por ejemplo, x, y, z) se representan mediante ejes trazados perpendicularmente entre sí. La distancia entre dos puntos cualesquiera se puede medir con una regla. Para las variables no correlacionadas, la distancia euclídea es igual a la MD. Sin embargo, si dos o más variables están correlacionadas, los ejes ya no están en ángulo recto, y las mediciones se hacen imposibles con una regla. Además, si tienes más de tres variables, no puedes dibujarlas en un espacio 3D normal. MD resuelve este problema de medición, ya que mide distancias entre puntos, incluso puntos correlacionados para múltiples variables.

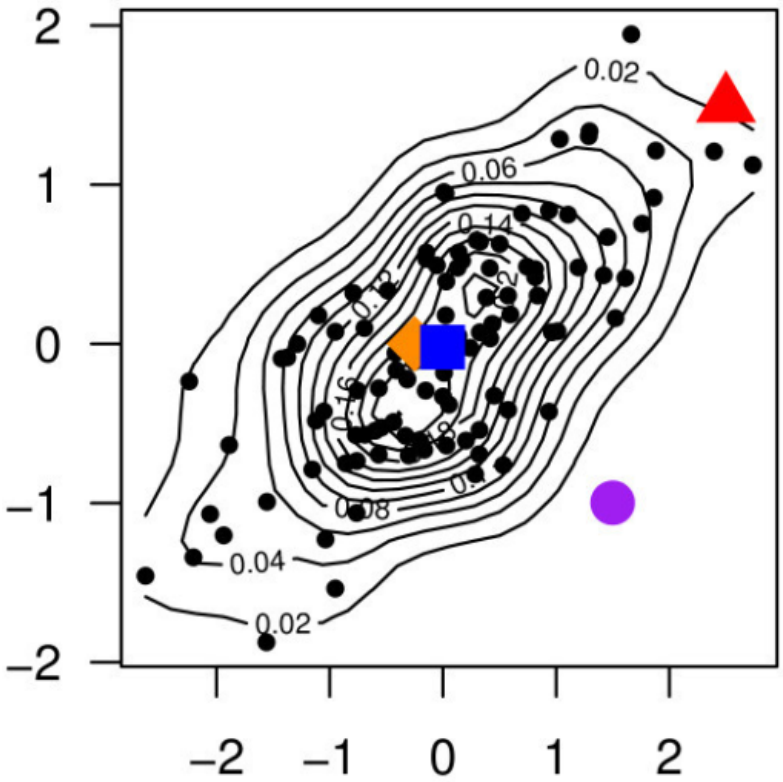


Figura 22 Ejemplo de diagrama de distancia de Mahalanobis.

En la Figura 22 se muestra una gráfica de contorno que superpone la gráfica de dispersión de 100 muestras aleatorias de una distribución normal en dos dimensiones con media cero, varianza uno y correlación del 50%. El centroide definido por el medio marginal está marcado por un cuadrado azul.

La distancia de Mahalanobis mide la distancia relativa al centroide - una base o punto central que puede ser considerado como una media global para datos con múltiples dimensiones. El centroide es un punto en el espacio donde se cruzan todos los medios de

todas las variables. Cuanto más grande es el MD, más lejos del centroide se encuentra el punto de datos.

Es una generalización multidimensional de la idea de medir cuántas desviaciones estándar hay entre P y la media de D. Esta distancia es cero si P está en la media de D, y crece a medida que P se aleja de la media a lo largo de cada eje principal del componente. En este caso, se va a calcular la distancia Mahalanobis en un espacio de dimensión 1.

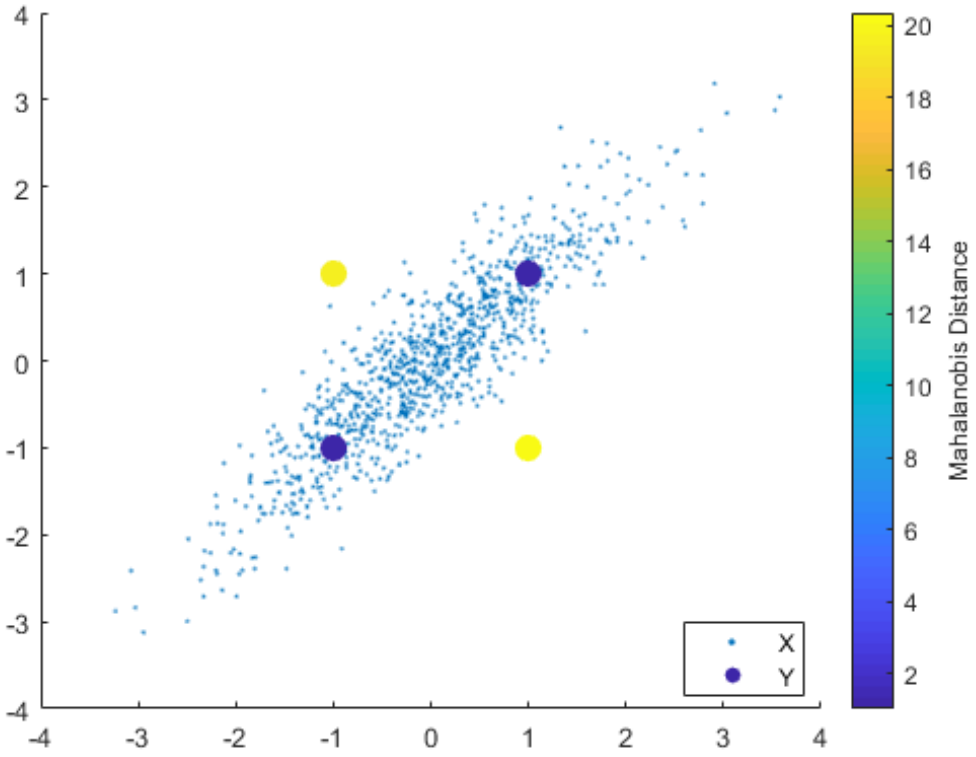


Figura 23 Ejemplo distancia Mahalanobis representado en un diagrama con distancia euclídea.

Si la distancia de Mahalanobis de la muestra a comparar es menor que la distancia máxima de cada uno de los valores del conjunto de la ventana deslizante, la muestra a procesar se clasificaría como un valor verdadero. En caso contrario, se calcula la desviación estándar de la muestra para medir la dispersión en relación con el conjunto de elementos de la ventana deslizante.

La desviación estándar, se define como la raíz cuadrada de la varianza (medida de dispersión de datos, el cuadrado del dato original y por ende el cuadrado de su unidad). Junto con este valor, la desviación típica (estándar) es una medida (cuadrática) que comunica la media de distancias que poseen los datos a proporción a su media aritmética, enunciada en las mismas unidades que la variable. De este modo, la desviación estándar hace referencia al cálculo medio o la media entre las diferencias relativas a datos y resultados y mientras más grande es la diferencia entre los datos, más grande va a resultar la desviación estándar o típica [23].

$$\text{Desviación estándar} = \sqrt{\frac{\sum|x-\bar{x}|^2}{n}}$$

Teniendo en cuenta esta definición, si representamos la distribución normal de la ventana deslizante y representamos la desviación estándar de la muestra leída podemos observar si la muestra pudiera ser un posible valor verdadero o no, según su distribución en múltiplos de la desviación estándar y de la verosimilitud del valor evaluado.

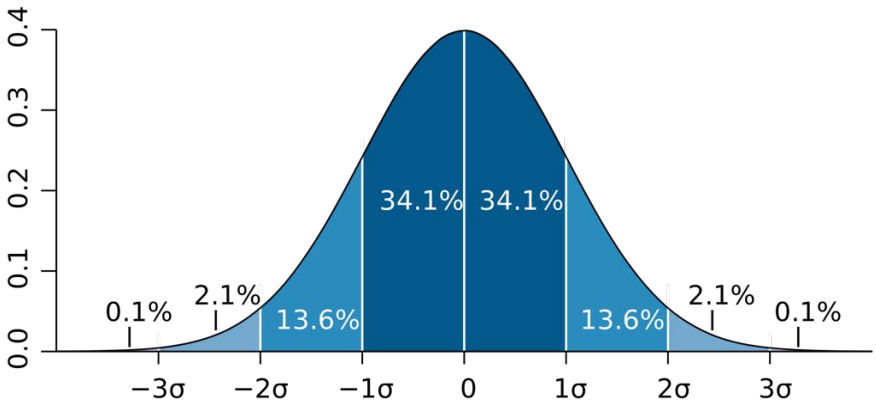


Figura 24 Representación de la desviación estándar.

σ = desviación estándar

Sabiendo:

- La media ± 1 * desviación estándar = cubre el 68,3% de los casos
- La media ± 2 * desviación estándar = cubre el 95,5% de los casos
- La media ± 3 * desviación estándar = cubre el 99,7% de los casos

Podemos calcular la desviación estándar [24] de la muestra al valor medio de la ventana deslizante. Si dicho valor supera el rango de ± 3 sigmas, entonces es calificado como outlier y se corrige en la matriz respuesta con el valor medio de la ventana. En el caso contrario, la muestra se califica como valor verdadero.

Una vez calificado el estado de la muestra que se lee, los valores calculados, excepto los nulos, se sustituyen en la ventana por la muestra cuyo valor de timestamp es el más antiguo. Esto permite mantener una correlación temporal de las muestras. Como resultado final, se obtiene una matriz con la información de identificación de los sensores, muestras corregidas, valores leídos y momento del envío.

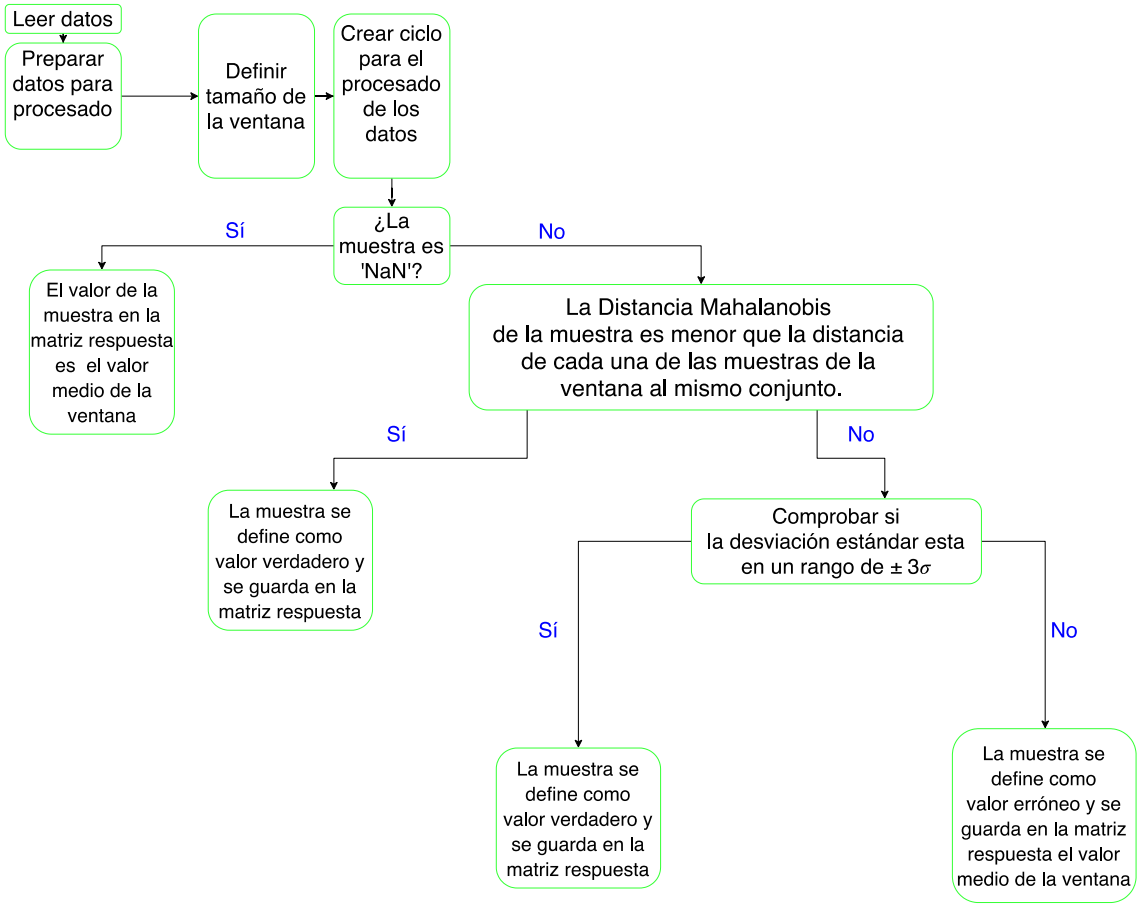


Figura 25 Planificación de funcionamiento del SDO.

El proceso de clasificación y detección de outliers que se ha explicado con anterioridad, puede aplicarse a un algoritmo llamado K-means clustering. Un cluster será la representación virtual de un grupo de nodos de la infraestructura. Este grupo de nodos se formará según su posición física en el mapa de Santander. Esto se ha implementado con la intención de que el SDO haga una mejor detección de las medidas erróneas de los sensores que están conectados a los nodos.

La agrupación de medidas en K grupos es un tipo de aprendizaje no supervisado, que se utiliza cuando se tienen datos no etiquetados (es decir, datos sin categorías o grupos definidos). El algoritmo funciona de manera iterativa para asignar cada punto de datos a uno de los grupos K en función de las características que se proporcionan. Los puntos de datos se agrupan en función de la similitud de las características.

Los centroides de los grupos K, que pueden ser usados para etiquetar nuevos datos. Etiquetas para los datos de formación (cada punto de datos se asigna a un único grupo).

En lugar de definir grupos antes de examinar los datos, la agrupación permite encontrar y analizar los grupos que se han formado orgánicamente.

Cada centroide de un cluster es un conjunto de valores de característica que definen los grupos resultantes. El estudio de las ponderaciones de las características del centroide puede utilizarse para interpretar cualitativamente qué tipo de grupo representa cada conglomerado.

El algoritmo de agrupamiento K-means [25] utiliza refinamiento iterativo para producir un resultado final. Las entradas del algoritmo son el número de clusters K y el conjunto de datos. El conjunto de datos es un conjunto de características para cada punto

de datos. Los algoritmos comienzan con las estimaciones iniciales para los centroides de K, que pueden generarse aleatoriamente o seleccionarse al azar del conjunto de datos. El algoritmo entonces itera entre dos pasos:

1. Paso de asignación de datos:

Cada centroide define uno de los grupos. En este paso, cada punto de datos se asigna a su centroide más cercano, basado en la distancia cuadrada de Euclides.

2. Paso de actualización del centroide:

En este paso, se vuelven a calcular los centroides. Esto se hace tomando la media de todos los puntos de datos asignados al cluster de ese centroide.

El algoritmo itera entre los pasos uno y dos hasta que se cumple un criterio de parada (es decir, ningún punto de datos cambia de grupo, la suma de las distancias se minimiza o se alcanza un número máximo de iteraciones).

Este algoritmo garantiza converger a un resultado. El resultado puede ser un óptimo local (es decir, no necesariamente el mejor resultado posible), lo que significa que la evaluación de más de una ejecución del algoritmo con centroides de inicio aleatorio puede dar un mejor resultado.

K-means encuentra los clusters y las etiquetas de conjuntos de datos para una K en particular preseleccionada. Para encontrar el número de clusters en los datos, el usuario necesita ejecutar el algoritmo para un rango de valores K y comparar los resultados. En general, no existe un método para determinar el valor exacto de K, pero se puede obtener una estimación precisa utilizando las siguientes técnicas.

Una de las métricas que se utiliza comúnmente para comparar los resultados a través de diferentes valores de K es la distancia media entre los puntos de datos y el centroide de su conglomerado. Dado que el aumento del número de clusters siempre reducirá la distancia a los puntos de datos, el aumento de K siempre disminuirá esta métrica, hasta el extremo de llegar a cero cuando K es el mismo que el número de puntos de datos. Por lo tanto, esta métrica no puede utilizarse como único objetivo. En su lugar, se traza la distancia media al centroide en función de K y se puede utilizar el punto donde la tasa de disminución se desplaza bruscamente, para determinar aproximadamente K.

Suponiendo que tenemos la información de unas muestras aleatorias representadas en un plano. Una vez establecido el número de grupos que queremos dividir nuestro dataset. El algoritmo se encarga de procesar la posición de los centroides y clasificar cada muestra en cada iteración.

Para el sistema de detección de outliers (SDO), el algoritmo K-means se va a utilizar con la intención de clasificar cada sensor de temperatura en un grupo según su localización geográfica. Si suponemos que para ciudades más grandes los valores de temperatura varían ligeramente según la zona de medida, este enfoque cobra sentido. La clasificación K-means puede ser la opción indicada para una mejora de precisión a la hora de procesar los datos.

En este caso, además de tener una correlación temporal en el momento de procesado, habrá una correlación espacial. Esto es debido a que los sensores formarán grupos según su posición geográfica. En consecuencia, las medidas que se van a procesar de cada cluster serán específicas de una zona geográfica.

Para ello se va a utilizar una base de datos con los detalles de longitud y latitud de cada sensor. Utilizando un convertidor de coordenadas esféricas a coordenadas

equidistantes azimutales, podemos proyectar dichos valores en un plano. A continuación, sabiendo que la infraestructura de SmartSantander dispone de 8 gateways, se va a utilizar el algoritmo K-means para la clasificación de los sensores en grupos con un valor de K = 8.

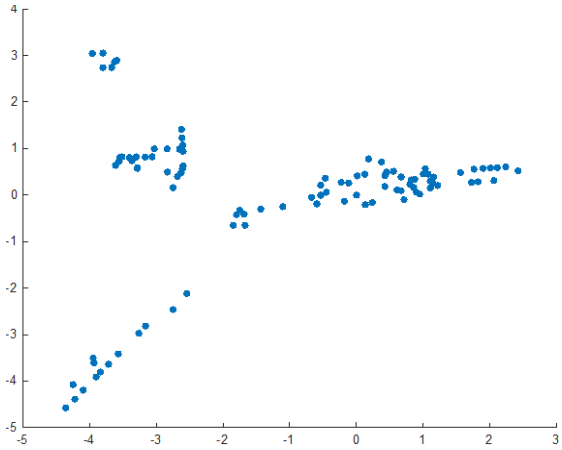


Figura 26 Representación de los sensores en un plano de dos dimensiones.

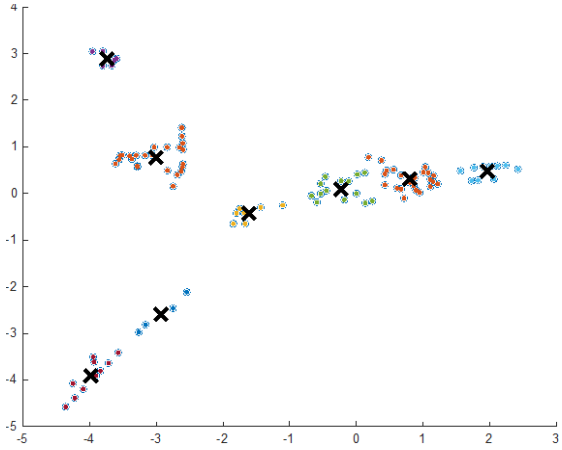


Figura 27 Representación gráfica de la clasificación del algoritmo K-means.

Para cada cluster formado se va a crear una ventana deslizante. Es importante mencionar que el tamaño de la ventana deslizante variará según el cluster. Abarcando un espacio temporal de una hora (este valor puede cambiar bajo cualquier criterio que se imponga). En cuanto al proceso de análisis, en cuanto se tengan los clusters formados, se dispondrá al filtrado del dataset para formar subsets de datos que representan dichos clusters. Una vez obtenidos los subgrupos de datos para cada cluster, se aplicará el SDO al igual que en el caso anterior, que fue para el dataset completo.

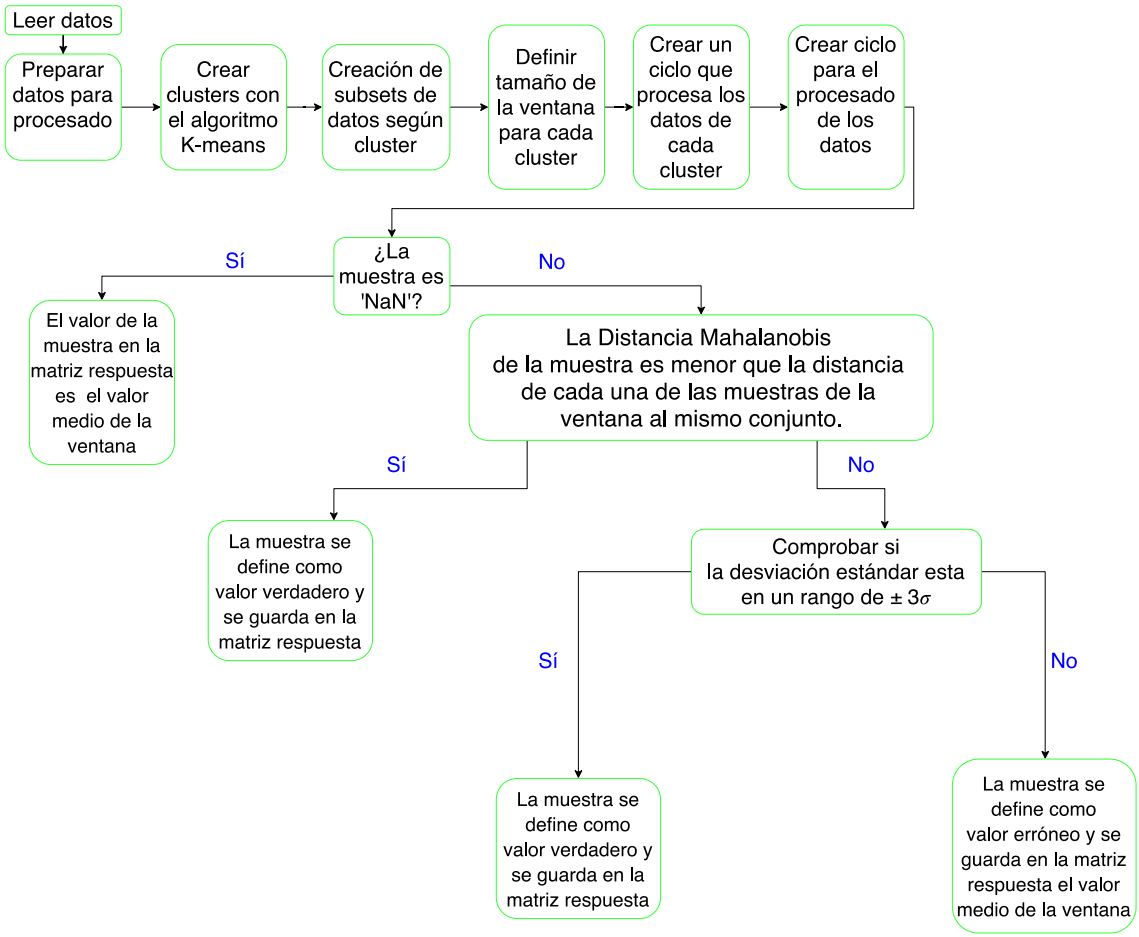


Figura 28 Planificación de funcionamiento de SDO con implementación de K-means

Partiendo del punto en el que tenemos los datos desorganizados en un solo dataset. Como primer paso organizamos dichas muestras en orden cronológico. A continuación mediante la implementación de K-means, creamos los subsets con las muestras de cada cluster. Calculando el tiempo medio de envío de muestras por segundo, deducimos la cantidad de muestras que se envían por hora según el cluster. El siguiente paso consiste en implementar un ciclo que aplicará el procesado de datos para la detección de outliers para cada cluster. En consecuencia se obtendrá K matrices respuesta de cada uno de los clusters de la infraestructura.

4.2 Uso de KNN para la eliminación de outliers

En este apartado de presentan los resultados del SDO así como la implementación adicional de K-means. Para las pruebas del SDO se necesitará una base de datos con las medidas recolectadas de la plataforma SmartSantander. En este caso, para la base de datos, la identificación de los sensores viene representada por un numero en concreto. Esto quizá simplifica nuestra implementación, pero es algo irrelevante a la hora de comprobar el funcionamiento de SDO. El dataset del que disponemos cubre los valores de quince días aproximadamente, del 23 de abril (13:53:29 pm) al 8 de mayo (13:52:00 pm) del 2019. El algoritmo de procesado para SDO, fue implementado para las pruebas utilizando la aplicación Matlab. Para este proceso se ha necesitado leer los datos de los ficheros que comprendían las muestras de los quince días que mencionamos con anterioridad. Una vez leídos los datos, se formará una tabla con el conjunto de muestras y se ordenará cronológicamente. Antes de utilizar el dataset me he dispuesto a cambiar el formato de la tabla que contenía las muestras a una matriz con los mismos datos pero

recodificados para facilitar el análisis. Por ejemplo, en la Figura 29 se muestra la tabla con los datos sin codificar, y en la Figura 30 se muestra la matriz preparada que contiene el dataset completo con los datos recodificados.

	1 ID	2 Temperatura	3 Timestamp
1	10003	'22,64'	23-Apr-2019 13:53:29
2	10008	'20,64'	23-Apr-2019 13:53:34
3	10015	'22,38'	23-Apr-2019 13:54:08
4	10007	'20,70'	23-Apr-2019 13:54:12
5	10004	'20,51'	23-Apr-2019 13:54:29
6	10006	'21,67'	23-Apr-2019 13:54:30
7	10002	'21,93'	23-Apr-2019 13:54:37
8	10009	'21,61'	23-Apr-2019 13:54:58
9	10005	'20,19'	23-Apr-2019 13:55:11
10	10012	'20,77'	23-Apr-2019 13:55:59

Figura 29 Muestras tabla del dataset

	1 ID	2 Temperatura	3 Timestamp
1	2	10.8300	7.3754e+05
2	5	11.8000	7.3754e+05
3	55	12.2500	7.3754e+05
4	165	11.6700	7.3754e+05
5	68	12.5100	7.3754e+05
6	73	11.8700	7.3754e+05
7	3	11.5400	7.3754e+05
8	85	12.7000	7.3754e+05
9	72	11.9300	7.3754e+05
10	66	226.3800	7.3754e+05

Figura 30 Muestras matriz del dataset

Al igual que en la Figura 30, disponemos de 3 columnas para nuestra base de datos a procesar (identificador, valor y timestamp). Al obtener dicha matriz con los datos preparados, nos disponemos a crear la ventana de datos teniendo en cuenta el estudio previo de la media de la variación de tiempo entre muestras en función de la distancia (Figura 19) o un histograma con la diferencia en segundos de tiempo entre muestras (Figura 31).

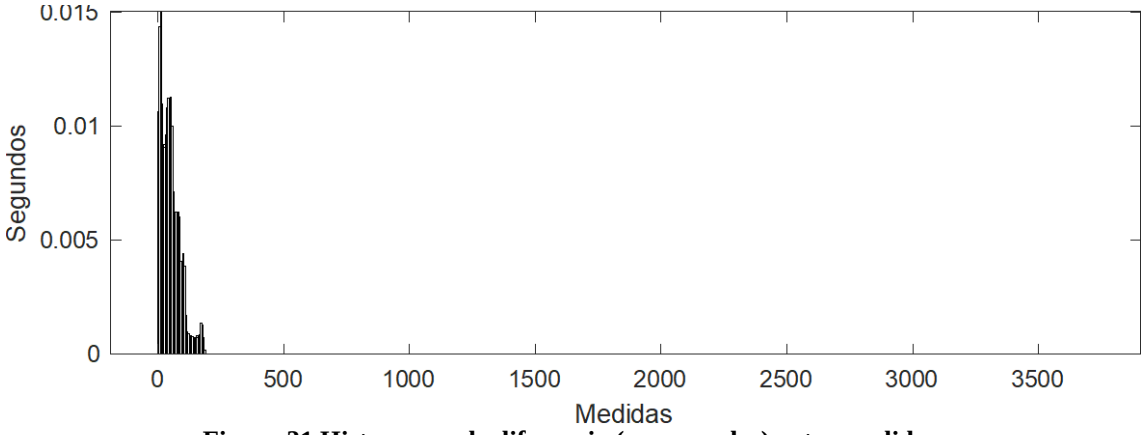


Figura 31 Histograma de diferencia (en segundos) entre medidas.

Después de analizar los valores, creamos una ventana con N datos para el análisis. Estos N datos serán los primeros N valores de la matriz dataset ordenada cronológicamente. La matriz ventana no tendrá N valores necesariamente en un principio, esto dependerá de si hay valores nulos, mayores de 40 o menores de 0 en dicha matriz. En mi caso, la matriz ventana tiene una dimensión real de 88 columnas en vez de 150 que se había comentado antes, debido a valores no válidos.

Antes de comenzar con el algoritmo del análisis de las muestras, nos disponemos a crear una matriz respuesta de 5 filas y (D-N) columnas, siendo D el número de muestras que tenemos en el dataset completo y N la dimensión de la ventana acordado. Al acabar dichos procedimientos, nos disponemos a la implementación del análisis tal y como se ha comentado con anterioridad.

Como respuesta general obtenemos la matriz según el esquema indicado en la Figura 21. Además, podemos proceder al estudio de dicha matriz para extraer valores como la cantidad de outliers total del dataset que se han detectado. Asimismo una lista de los sensores que han mandado dichas muestras (en total 63 sensores con valores erróneos).

1x63 double

	1	2	3	4	5	6	7	8	9
1	2	3	55	66	68	73	85	160	165

Figura 32 Muestra ejemplo de la matriz con los identificadores de los nodos emisores de outliers.

Para este análisis se ha calculado que de las 378127 muestras de las que se dispone, 220513 son valores correctos, 66432 outliers y 91182 valores nulos (NaN).

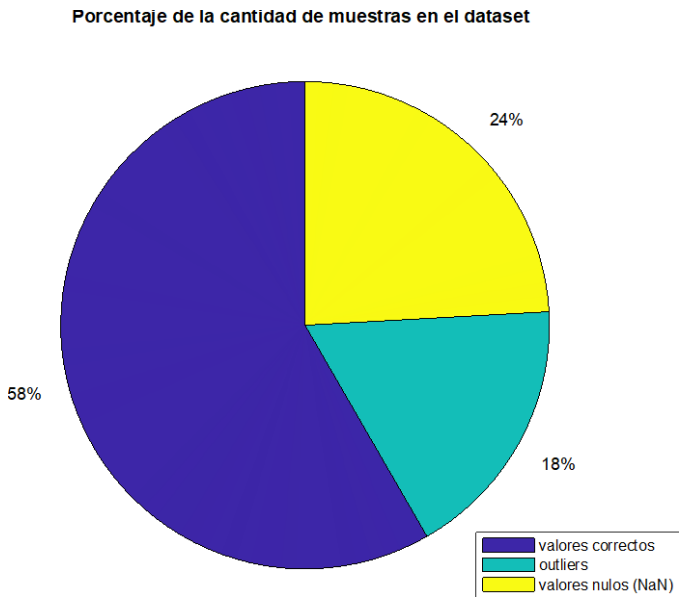


Figura 33 Representación gráfica de los resultados aplicando al dataset completo el SDO.

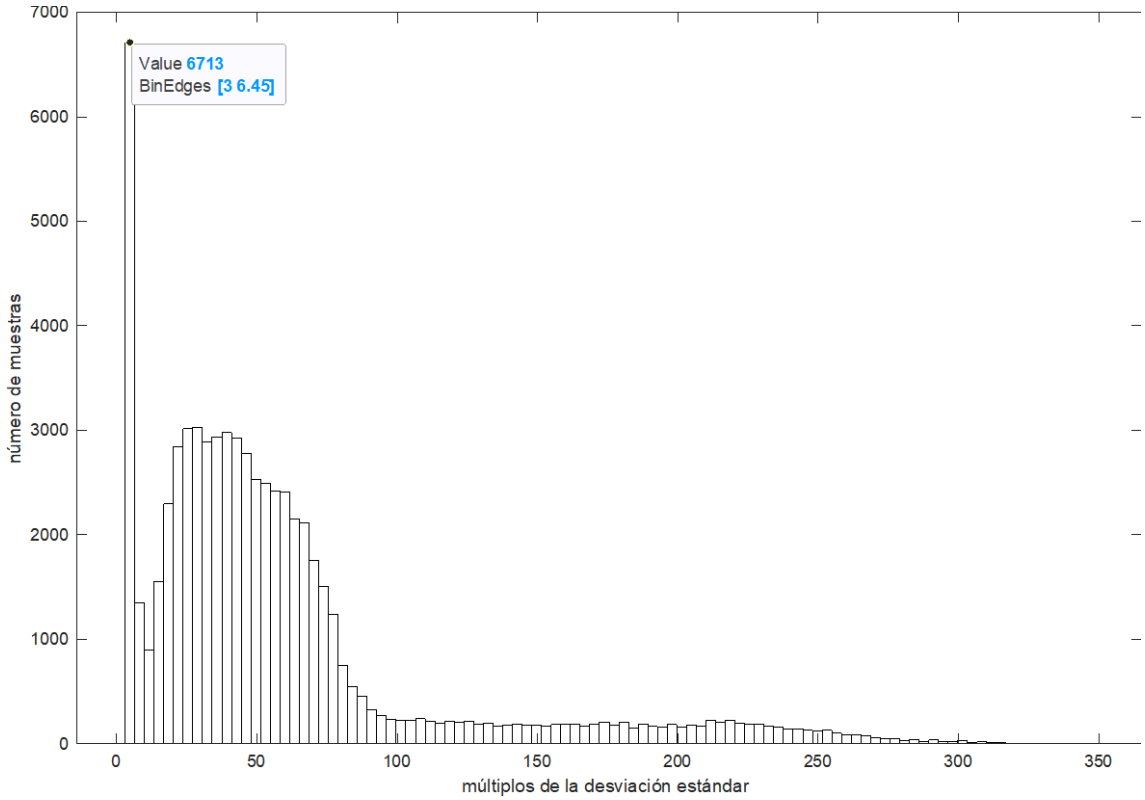


Figura 34 Histograma de los outliers y su distancia respecto a la desviación estándar.

Para el caso que abarca la implementación del algoritmo K-means en el SDO, el análisis de los datos se realizará en orden de clúster. Es decir, un orden arbitrario que elegí para marcar una secuencia de procesado de los datos. Los resultados de K-means se representan en una matriz que contiene la conversión de coordenadas esféricas a coordenadas azimutales equidistantes, la identificación de nodo, el número de cluster al que pertenece y la posición geográfica (latitud y longitud) tal como se muestra en la Figura 35 a continuación.

	1 X	2 Y	3 ID	4 Nr.Cluster	5 Latitud	6 Longitud
31	-3.7183	-3.6387	583	6	43.4557	-3.8116
32	-2.7503	-2.4610	615	7	43.4578	-3.8092
33	-3.8393	-3.8053	735	6	43.4554	-3.8119
34	-4.0974	-4.1942	736	6	43.4547	-3.8125
35	-4.2506	-4.0775	737	6	43.4549	-3.8129
36	-2.5446	-2.1166	612	7	43.4584	-3.8087
37	-3.5731	-3.4165	609	6	43.4561	-3.8112
38	-4.3596	-4.5775	606	6	43.4540	-3.8132
39	-3.1617	-2.8165	604	7	43.4571	-3.8102
40	-3.9360	-3.6109	599	6	43.4557	-3.8121
41	-3.9038	-3.9109	586	6	43.4552	-3.8120
42	-3.2625	-2.9721	593	7	43.4569	-3.8104
43	-3.9481	-3.5053	575	6	43.4559	-3.8121
44	-4.2225	-4.3886	594	6	43.4543	-3.8128

Figura 35 Matriz ejemplo con la información de los nodos respecto a los clusters.

La matriz de la Figura 35 nos lleva a crear un dataset para cada cluster. Para ello se filtra la matriz para crear los datasets de cada cluster por separado (Figura 36).

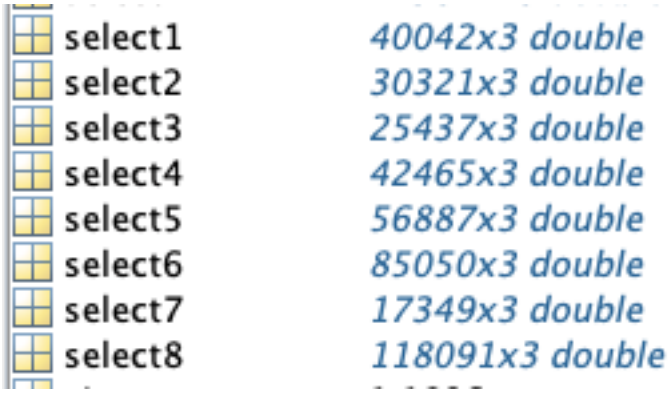


Figura 36 Matriz select-n contiene los datasets correspondientes a cada cluster.

Al tener preparadas las matrices ‘select-n’ nos disponemos a aplicar el SDO a cada uno de los datasets tal y como se ha explicado con anterioridad. Es decir, se calcula la dimensión de cada ventana (número de valores que cada ventana perteneciente a un select va a tener). En la Figura 37 podemos observar que no todas las ventanas tienen la misma dimensión. Esto es debido a que para cada cluster habrá un flujo distinto de datos, por lo tanto, para un cluster se pueden enviar más muestras que en otro en un tiempo determinado.

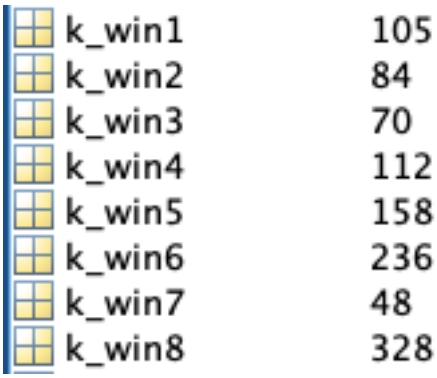


Figura 37 K_win - n es la variable que indica la dimensión de la ventana que se calcula.

En la Figura 38 podemos observar que las ventanas en un principio van a tener menos valores que los calculados anteriormente, esto es debido a que los valores nulos y aquellos que se eliminan con el primer filtrado, es decir los valores mayores de 40 y menores de 0, han sido ya eliminados. Como resultado, quedan unas ventanas redimensionadas con valores más fiables con los que se podría empezar el análisis del SDO.

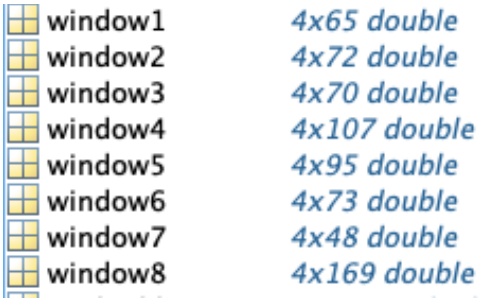


Figura 38 window-n representa la ventana de cada cluster que se va a procesar.

Al acabar el análisis del SDO, se obtienen las matrices respuesta de cada cluster. Estas matrices tienen la información acerca del análisis de las muestras de cada cluster.

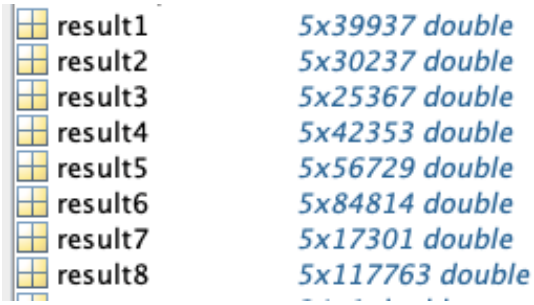
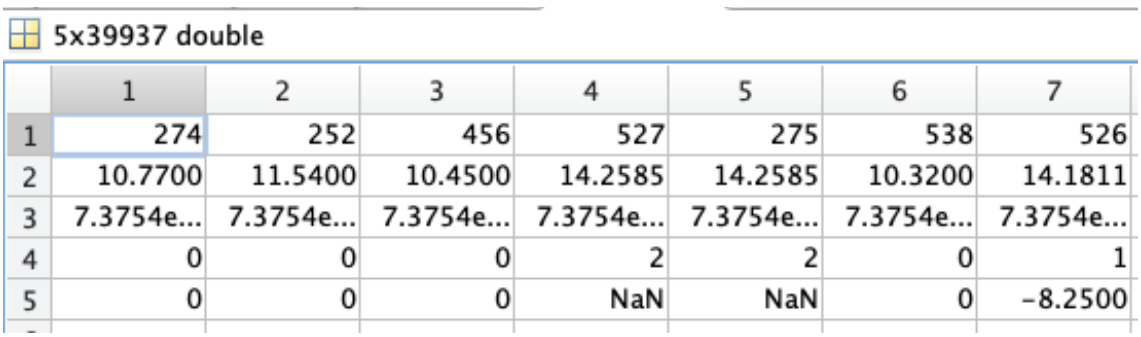


Figura 39 Result-n representación de la matriz respuesta de cada cluster.


5x39937 double

	1	2	3	4	5	6	7
1	274	252	456	527	275	538	526
2	10.7700	11.5400	10.4500	14.2585	14.2585	10.3200	14.1811
3	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...
4	0	0	0	2	2	0	1
5	0	0	0	NaN	NaN	0	-8.2500

Figura 40 Representación de los valores de una matriz respuesta.

outlier1	25081
outlier2	884
outlier3	401
outlier4	6660
outlier5	4939
outlier6	6803
outlier7	386
outlier8	42730

Figura 41 outlier-n indica la cantidad de outliers detectados en cada cluster.

empty1	9841
empty2	0
empty3	0
empty4	0
empty5	18208
empty6	54543
empty7	0
empty8	18216

Figura 42 empty -n contador de valores 'NaN' de cada cluster.

Si nos fijamos en las imágenes representadas en la Figura 36 y Figura 41, podemos deducir que los resultados obtenidos con la implementación de K-means no son los óptimos. Esto es debido a que cada cluster (Figura 49) representa un porcentaje distinto de datos del total. Esto da lugar a que algunos clusters tengan muchos datos con posibles errores (e.g. select8 Figura 36 y outlier8 Figura 41). Otros tendrán pocos nodos asignados al cluster, por lo tanto, carece de variedad en las muestras a la hora de analizar los datos (e.g. Figura 44). Si un cluster tiene pocos nodos asignados (Figura 46), habrá momentos en el análisis en el que solo un nodo envíe datos de forma constante (Figura 45), esto estrecha la campana de gauss y limita la desviación estándar entre las muestras, además de producir unos valores muy altos en la distancia de Mahalanobis. En consecuencia, al analizar las muestras bajo las malas condiciones indicadas previamente, produce una respuesta poco fiable (Figura 44).

1075	1076	1077	1078	1079	1080	1081
514	357	354	374	518	515	367
17.3860	17.3860	17.3860	17.3860	17.3860	15.9300	16.9000
7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...
1	2	1	1	1	0	0
-27.0300	NaN	-41.1600	206.3200	-18.8300	0	0

Figura 43 Representación la matriz respuesta ante un funcionamiento normal.

576	577	578	579	580	581	582	583
170	5	170	5	170	5	170	5
16.9000	17.4646	16.8300	17.4606	17.0300	17.4548	16.9600	17.4488
7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...
0	1	0	1	0	1	0	1
0	11.3500	0	11.2900	0	11.4100	0	10.9000

Figura 44 Representación de la matriz respuesta ante una situación de pocos nodos por cluster.

479	480	481	482	483	484	485
5	5	5	5	5	5	5
16.2500	15.6700	17.1334	17.1334	17.1334	17.1334	17.1334
7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...	7.3754e...
0	0	1	1	1	1	1
0	0	15.0300	14.7000	14.3800	14.7700	14.5800

Figura 45 Representación de la matriz respuesta ante una predominancia de muestras de un sensor en el clúster.

En la Figura 45 se da una situación en la que hay una ráfaga de valores pertenecientes a un solo nodo. Esto llega a ser válido, y dichas muestras se detectan como valores correctos hasta que ese mismo nodo deja de enviar muestras por un periodo de tiempo. Al reanudarse el proceso, como es de esperar, la temperatura ha cambiado pero las muestras de la ventana que se utilizaba para la detección de outliers ahora tienen valores muy distintos a los recibidos para el análisis. En este caso, aun recibiendo valores del mismo nodo que a una primera vista parecen datos coherentes, se detectarán como outliers debido a un estrechamiento de la campana de gauss, es decir, valores muy pequeños de la desviación estándar y una gran MD de la muestra respecto a cada uno de los valores de la ventana.

Este caso se ha intentado resolver mediante el cálculo de la diferencia de tiempos de envío entre muestras consecutivas antes de su análisis para la detección de outliers. Es decir, se comprueba si en timestamp de la muestra a procesar tiene una diferencia mayor de una hora (marco temporal de la ventana en ese momento) respecto al último valor añadido a la ventana. Si se da el caso, el SDO actualiza la ventana con las N muestras consecutivas obtenidas a continuación del dataset. Por el contrario, si la diferencia de tiempo entre la muestra a procesar y la última muestra de la ventana es menor que el marco temporal límite establecido, el SDO continúa con el análisis.

Esta implementación soluciona el problema de la falta de muestras, pero también se han dado casos en los que los cambios bruscos de los valores en las muestras del dataset son debidos a comportamientos anómalos del sensor. Una forma de evitar este tipo de casos sería aumentar el número de nodos por clúster y así tener una mayor variabilidad de las muestras en la ventana.

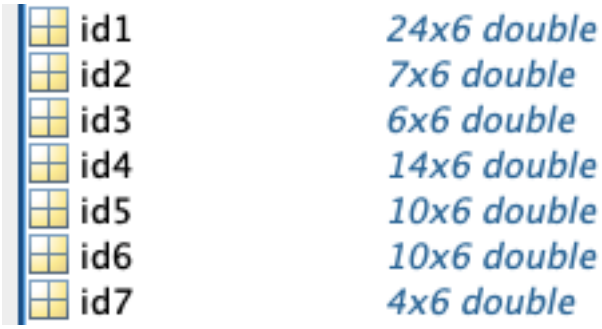


Figura 46 id-n matriz indicadora de la cantidad de nodos por clúster.

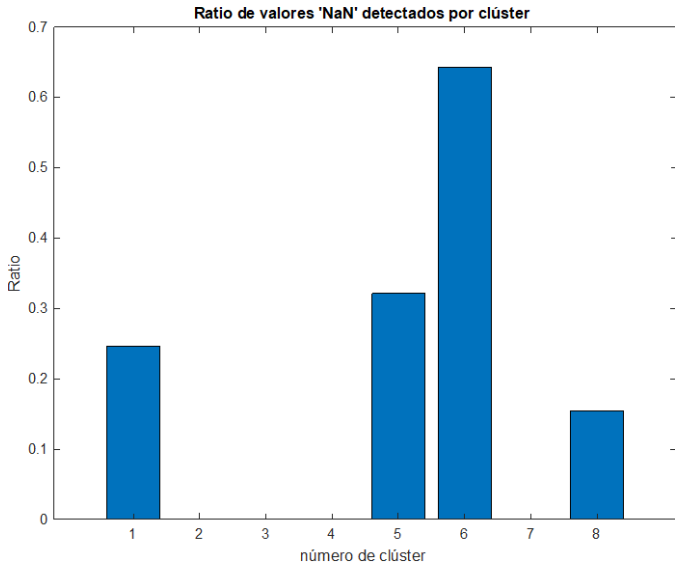


Figura 47 Ratio de valores 'NaN' detectados por clúster.

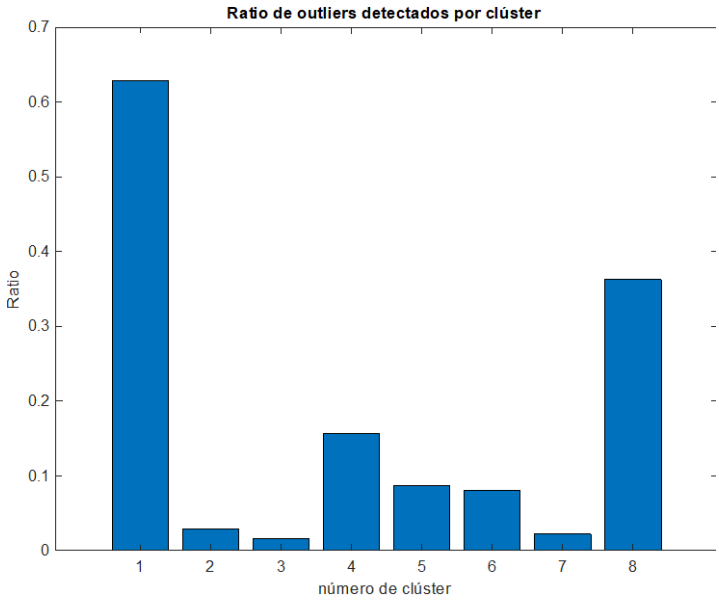


Figura 48 Ratio de outliers detectados por clúster.

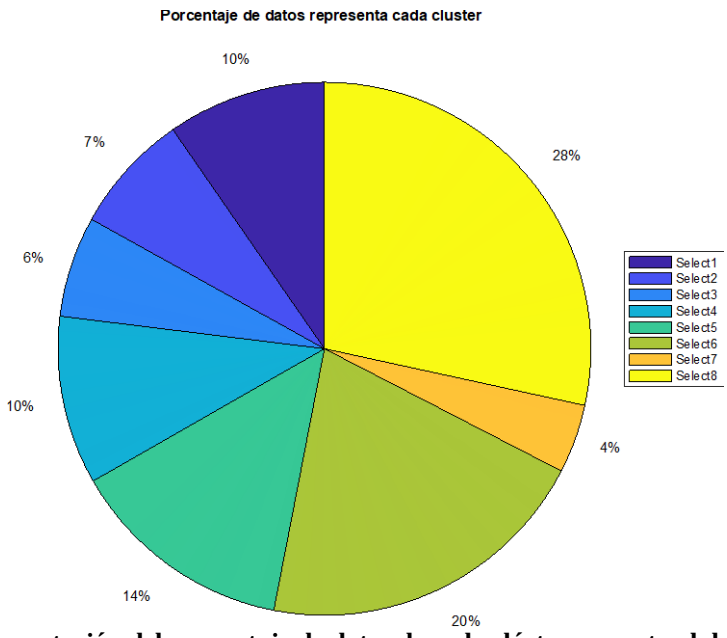


Figura 49 Representación del porcentaje de datos de cada clúster respecto al dataset completo.

4.3 Uso de SVM para la reducción de la dimensionalidad

El funcionamiento de los sensores en la infraestructura de SmartSantander se basa en el envío continuo de datos a los gateways, que a su vez hacen llegar estas medidas a los servidores centrales en la nube. El problema que tiene esta estrategia de envío y retrasmisión de datos es que al aumentar el número de dispositivos conectados, el ancho de banda que se utiliza en la red aumenta en proporción. Esto es debido a que cada muestra que se genera en los sensores es reenviada a la nube para su procesado. Es una de las causas que limita la cantidad de dispositivos en la infraestructura. Para dar solución a este problema de dimensionalidad, se pretende utilizar técnicas de predicción de datos como estrategia de gestión de la información retransmitida. Consiste en el procesado de los datos para la estimación de medidas en cada uno de los gateways y la retransmisión de resultados a la nube solo en los casos en que la estimación no resulte adecuada. Es una logística que hará uso del paradigma informático distribuido denominado Fog Computing [26].

Fog Computing es una forma de computación distribuida, donde cada uno de los dispositivos (en este caso los gateways) conectados a la red puede procesar los datos y solo transmitir un resumen al nivel superior, o hacerlo solo en determinados casos de alarmas o comportamientos poco frecuentes.

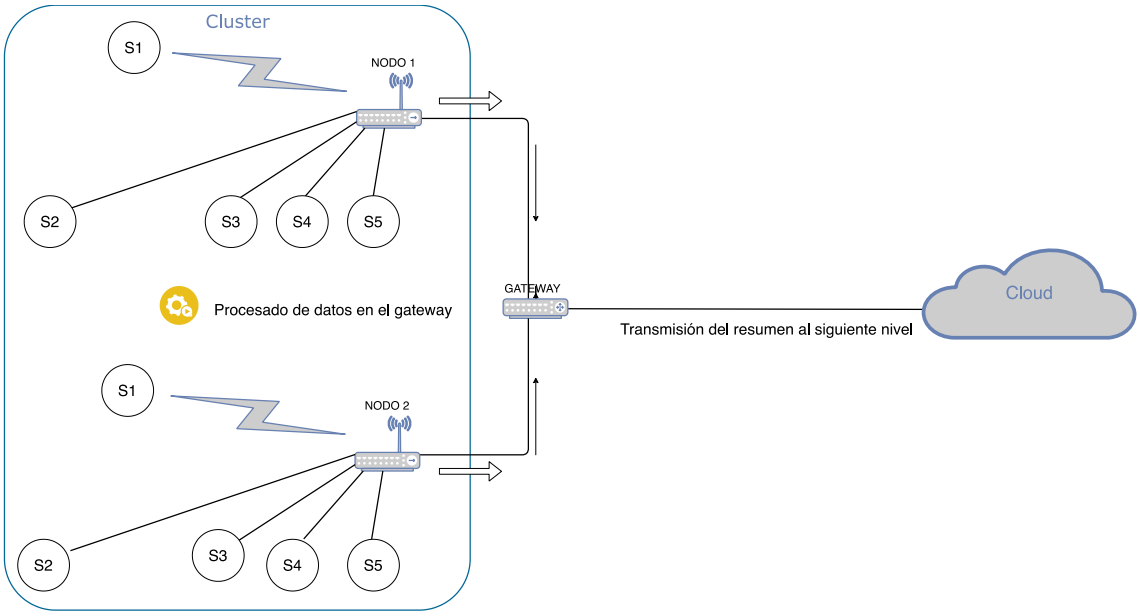


Figura 50 Esquema de la reducción de dimensión en la plataforma.

En la Figura 50 se muestra un ejemplo de esquema del proceso de funcionamiento para la reducción de dimensionalidad. Cada uno de los gateways recibe información de cinco sensores (el número de sensores que tiene cada nodo es una simple suposición). El procesado en el consistirá en analizar las medidas de los sensores para, a continuación hacer una eliminación de outliers y enviar un resumen del proceso al cloud. El resumen que enviará el nodo comprenderá un modelo regresivo con la información necesaria para una predicción de valores. Dicha predicción brinda la posibilidad de saber el valor de las muestras sin tener la necesidad de que el gateway haga un envío del dataset completo. En consecuencia ahorraría recursos de la red.

Un modelo regresivo es una técnica estadística de aplicación frecuente que sirve de base para el estudio y caracterización de un sistema de interés, mediante la formulación de un modelo matemático de la relación entre una variable de respuesta, ‘y’ y un conjunto de variables explicativas ‘q’ (‘x1, x2, ... xq’). La elección de la forma explícita del modelo puede basarse en el conocimiento previo del sistema o en consideraciones como la "suavidad" y la continuidad de ‘y’ en función de las variables ‘x’. En términos muy generales, todos estos modelos pueden considerarse de forma:

$$y = f(x1,...xq) + e$$

donde la función ‘f’ refleja la relación verdadera pero desconocida entre 'y' y las variables explicativas. El error aditivo aleatorio 'e', se supone que tiene de media 0 y la varianza sigma_e^2, refleja la dependencia de 'y' de cantidades distintas de x1,....,xq. El objetivo es formular una función fhat(x1,x2,...,xp) que sea una aproximación razonable de 'f'.

Para la implementación y pruebas del funcionamiento del proceso explicado anteriormente se hará uso de una máquina de vectores de soporte (SVM). SVM es un algoritmo de aprendizaje supervisado. El uso de dicho algoritmo está basado en la clasificación binaria o regresión de los datos. En este caso se utilizará la creación de un modelo regresivo que más tarde será empleado para la predicción de valores de las muestras. Para el uso de SVM se necesita principalmente una base de datos ‘ground truth’.

En el aprendizaje automático, el término " ground truth " se refiere a la precisión de la clasificación del conjunto de entrenamiento para las técnicas de aprendizaje supervisado. Esto se utiliza en modelos estadísticos para probar o refutar hipótesis de investigación. El término "ground truthing" se refiere al proceso de recopilar los datos objetivos (demostrables) apropiados para esta prueba. Por ejemplo, supongamos que estamos probando un sistema de visión estereoscópica para ver qué tan bien puede estimar las posiciones en 3D. El término ‘ground truth’ se podría definir como las posiciones dadas por un telémetro láser que se sabe que es mucho más preciso que el sistema de cámara.

En esta implementación, el ‘ground truth’ será nuestra matriz respuesta que se obtendrá después de aplicar el proceso de detección de outliers del apartado anterior. Dicha matriz se utilizará para crear un modelo regresivo mediante el uso de SVM. Como se ha comentado con anterioridad, la respuesta de la máquina de vectores de soporte es un modelo entrenado que se puede emplear para la predicción de dichos datos.

El proceso que se seguirá una vez obtenida la matriz respuesta en cada gateway será el de crear un modelo regresivo con SVM. Para ello se ordenará (si hace falta) la matriz respuesta cronológicamente en función del timestamp. A continuación se seleccionarán las filas de los valores corregidos y timestamp de la matriz respuesta. Esto nos proporcionara una matriz de dos columnas representando cada característica respectivamente.

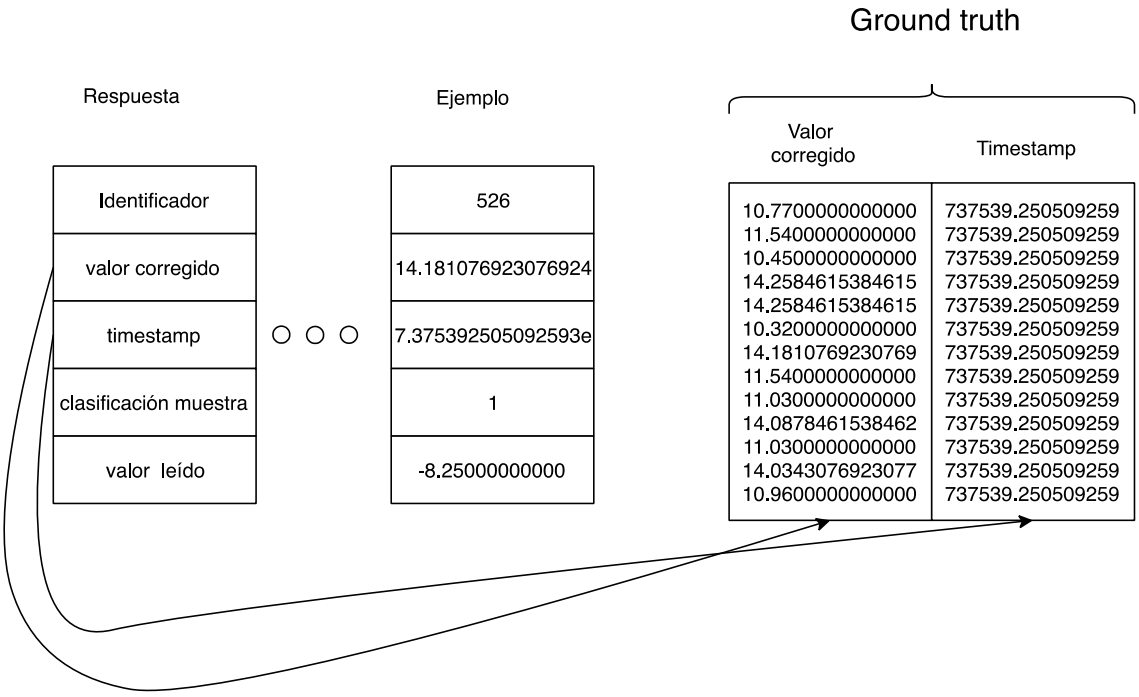


Figura 51 Representación de la creación de una matriz 'ground truth' para la generación de un modelo SVM.

La matriz de dos columnas obtenida se empleará para el entrenamiento del modelo regresivo empleando SVM. Dicho modelo será enviado al cloud donde se podrá usar para la predicción de los datos.

1x1 RegressionSVM

Property	Value
BoxConstraints	*39937x1 double*
CacheInfo	*1x1 struct*
ConvergenceInfo	*1x1 struct*
Epsilon	0.1000
Gradient	*79874x1 double*
IsSupportVector	*39937x1 logical*
NumIterations	7640
OutlierFraction	0
ShrinkagePeriod	0
Solver	'SMO'
Y	*39937x1 double*
X	*39937x2 double*
RowsUsed	*[ ]*
W	*39937x1 double*
ModelParameters	*1x1 SVMParams*
NumObservations	39937
BinEdges	*0x0 cell*
HyperparameterOptimizationResults	*[ ]*
PredictorNames	*1x2 cell*
CategoricalPredictors	*[ ]*
ResponseName	'Y'
ExpandedPredictorNames	*1x2 cell*
ResponseTransform	'none'
Alpha	*2854x1 double*
Beta	*[ ]*
Bias	16.7949
KernelParameters	*1x1 struct*
Mu	[12.4708,7.3755e+05]
Sigma	[2.2713,3.3956]
SupportVectors	*2854x2 double*

Figura 52 Datos que contiene un modelo SVM una vez creado.

Para el proceso predictivo se necesitará proporcionar el modelo entrenado y una secuencia de timestamps. La secuencia de timestamps necesaria no tiene que coincidir con la que se ha utilizado para el entrenamiento del modelo pero sí que tiene que ser una secuencia posterior a la primera muestra utilizada en la matriz 'ground truth'. Como resultado se obtendrán los valores de temperatura para cada timestamp proporcionado con un factor de error del proceso que se muestra a continuación.

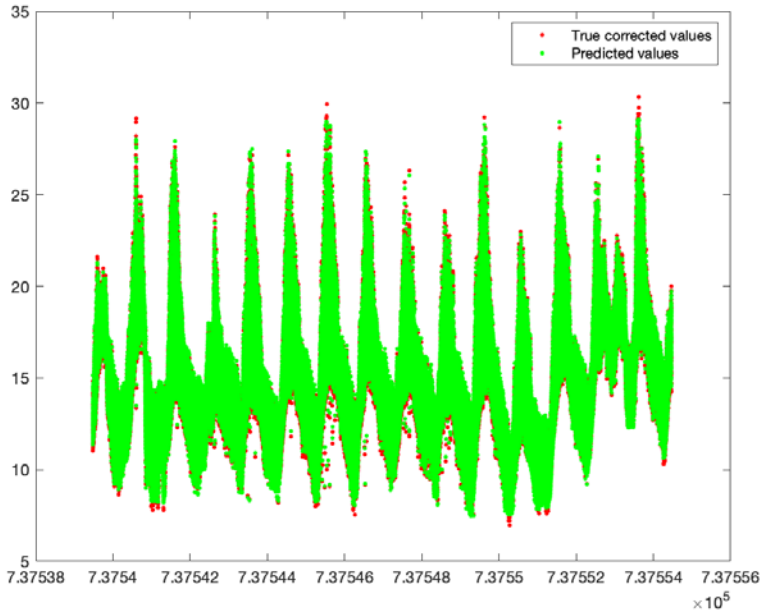


Figura 53 Representación gráfica de los valores reales de un dataset y su predicción.

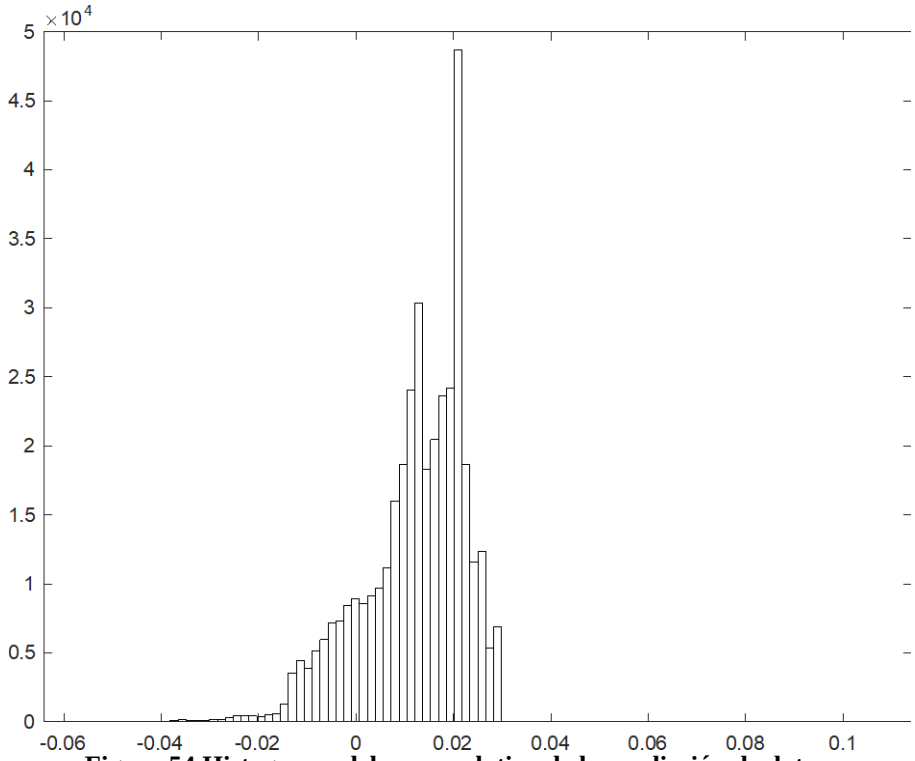


Figura 54 Histograma del error relativo de la predicción de datos.

La media de error en los cálculos de 16,35% para la predicción de los valores en el dataset completo. En base a estos resultados podemos concluir que en el caso de haber implementado este sistema en los gateways de SmartSantander, podríamos haber evitado el envío de los 415 642 valores ya que ejecutando la función modelada en los servidores centrales directamente se tendrían los mismos valores que los reales, salvo por un error que en la media es poco más del 16%.

Capítulo 5: Conclusiones y Líneas Futuras

En este apartado se analiza la consecución de los objetivos que se plantearon al comenzar el TFG. Además, se comenta la posible continuación de este trabajo para una mejora de rendimiento y resultados.

5.1 Conclusiones

La implementación de técnicas de análisis de datos en una infraestructura IoT supone una necesidad ante el futuro desarrollo de nuevas aplicaciones. Como principal conclusión es importante destacar que se ha cumplido con los objetivos que se plantearon al inicio de este TFG.

El SGD que se ha implementado, cumple con los objetivos acordados que son los de proporcionar información acerca de los sensores que están activos en la red. Este sistema es una fuente de información importante a la hora de monitorizar el funcionamiento de los dispositivos en una infraestructura IoT.

En cuanto al sistema de gestión de outliers, se puede afirmar que tiene con diferencia mejores resultados el análisis del dataset completo en comparación con la aplicación del algoritmo K-means. Esto es debido a que el SDO todavía no tiene una solución de cara al control de las muestras, es decir, no se controla la posibilidad de falta de muestras en el dataset. Al no tener una garantía de que hay un flujo constante de medidas recibido para un análisis posterior, en una ventana de clúster puede haber falta de muestras durante horas, lo que conlleva saltos de temperatura notables. Esta diferencia brusca entre valores puede detectar las muestras como outliers en una simple situación ante una diferencia temporal enorme. Por otro lado, es necesario que cada cluster esté clasificado con un número de nodos bastante amplio para asegurar un flujo suave ante el cambio de valores en las muestras.

En la situación del análisis del dataset completo con el SDO, al tener todas las muestras en un mismo dataset, tenemos la fluidez necesaria en cuanto al valor de las medidas. Lo que nos proporciona una respuesta de salida más fiable. Para la decisión de la utilidad de este algoritmo en el SDO, habría que hacer más pruebas con un número elevado de nodos en la red (para así aumentar el número de nodos por clúster), sea mediante la variación de la constante K o la implantación de nuevos dispositivos.

Respecto a la reducción de dimensionalidad, cabe destacar que muestra unos resultados buenos con el uso de SVM. La predicción de las muestras tiene un error del 16% respecto a los valores reales lo que supone un error poco relevante ante futuros cálculos en el cloud.

5.2 Líneas futuras

Como posibilidad de extender el trabajo respecto al SGD es la proyección en el mapa de cada uno de los nodos con las etiquetas respectivas presentando la información de disponibilidad de cada sensor. Esto proporcionaría una forma visual rápida de gestión y mantenimiento de cualquier sensor en tiempo real. Además, se podría añadir un listado de todos los nodos y los sensores conectados a la red, para así tener un control total del funcionamiento de cada uno de ellos.

Para el SDO y la reducción de dimensionalidad, como futuro trabajo, se podría implementar un control de perdida de muestras y mecanismos de gestión ante una situación de falta de valores debido a un espacio temporal relativamente grande. Además, habría que estudiar mejor la importancia de K-means en el SDO. Es decir, aumentar la cantidad de nodos por cluster para tener más medidas respectivamente. Asimismo comprobar la relevancia de K-means para una futura implementación del sistema en la infraestructura.

Respecto a la reducción de dimensionalidad, antes de su implementación en una infraestructura, es necesario asegurarse de utilizar el mejor algoritmo predictivo que implique el menor coste de procesado. A su vez, se podría estudiar la mejor combinación de características de las muestras para una predicción más eficiente y precisa. Para ello haría falta el uso de un sistema de aprendizaje automático declarativo para el ciclo de vida completo [38].

Bibliografía

[1] J. Manyika, M. Chui y J. Bughin, «McKinsey Global Institute,» Mayo 2013. [En línea]. Available: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/disruptive-technologies.

[2] J. Manyika, M. Chui, J. Woeztel, J. Bughin, P. Bisson, R. Dobbs y D. Aharon, «Unlocking the potential of the Internet of Things,» McKinsey Global Institute, 2015. [En línea]. Available: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world. [Último acceso: 26 Agosto 2019].

[3] T. D. 2. g. challenge, «The ACM Digital Library,» 26 May 2014. [En línea]. Available: https://dl.acm.org/citation.cfm?doid=2611286.2611333. [Último acceso: 26 Agosto 2019].

[4] O. o. N. Statistics, «2011 Census: Population and household estimates for the United Kingdom, March 2011,» 21 Marzo 2013. [En línea]. Available: https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/bulletins/populationandhouseholdestimatesfortheunitedkingdom/2011-03-21. [Último acceso: 27 Agosto 2019].

[5] Paradigm4, «Data Scientist Survey Infographic,» 1 Julio 2014. [En línea]. Available: https://www.paradigm4.com/infographic2014/.

[6] A. f. t. I. o. T. A. Survey, «ResearchGate,» Enero 2018. [En línea]. Available: https://www.researchgate.net/publication/326171965_Analytics_for_the_Internet_of_Things_A_Survey. [Último acceso: 29 Agosto 2019].

[7] B. Dykas, «Telit,» 5 Junio 2018. [En línea]. Available: https://www.telit.com/blog/three-most-common-iot-problems/.

[8] Oxford English Dictionary, 2017.

[9] J. W Tukey, «The Future of Data Analysis,» 1 Julio 1961. [En línea]. Available: https://projecteuclid.org/download/pdf_1/euclid.aoms/1177704711. [Último acceso: 29 Agosto 2019].

[10] U. Fayyad, «From Data Mining to Knowledge Discovery in Databases,» 15 Marzo 1996. [En línea]. Available: https://doi.org/10.1609/aimag.v17i3.1230.

[11] C. o. Analytics y T. Davenport, «Harvard Business Review,» 2016. [En línea]. Available: https://hbr.org/2006/01/competing-on-analytics.

[12] H. t. W. c. managers y H. Varian, «McKinsey,» Enero 2009. [En línea]. Available: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/hal-varian-on-how-the-web-challenges-managers.

[13] A. 3.0 y T. Davenport, «Harvard Business Review,» Diciembre 2013. [En línea]. Available: https://hbr.org/2013/12/analytics-30.

[14] B. D. A. D. V. P. V. Prescriptive y J. Bertolucci, «InformationWeek,» 31 Diciembre 2013. [En línea]. Available: https://www.informationweek.com/big-data/big-data-analytics/big-data-analytics-descriptive-vs-predictive-vs-prescriptive/d/d-id/1113279.

[15] G. B. A. Framework, N. Chandler, B. Hostmann, N. Rayner y G. Herschel, «Gartner,» 22 Septiembre 2011. [En línea]. Available: https://www.gartner.com/imagesrv/summits/docs/na/business-intelligence/gartners_business_analytics__219420.pdf.

[16] T. F. T. O. Analytics y M. Corcoran, «Docplayer,» 2012. [En línea]. Available: https://docplayer.net/985643-The-five-types-of-analytics-michael-corcoran-sr-vice-president-cmo.html.

[17] W. HALL, «Arxiv,» 3 Julio 2018. [En línea]. Available: https://arxiv.org/pdf/1807.00971.pdf.

[18] T. a. I. o. T. b. a. f. f. defence y P. P. Ray, «Ieeexplore,» IEEE, 23 Mayo 2016. [En línea]. Available: https://ieeexplore.ieee.org/document/7475314/authors#authors.

[19] A. i. t. briefing, «Researchgate,» Enero 2014. [En línea]. Available: https://www.researchgate.net/publication/263326113_Apple_iBeacon_technology_briefing.

[20] Wikipedia, «Beacons,» [En línea]. Available: https://en.wikipedia.org/wiki/Types_of_beacons.

[21] J. M. Huidobro, «La tecnología RFID,» [En línea]. Available: https://www.acta.es/medios/articulos/ciencias_y_tecnologia/058037.pdf.

[22] M. Wolf, «ieeexplore,» 2017. [En línea]. Available: https://ieeexplore.ieee.org/abstract/document/7748567/authors#authors.

[23] [En línea]. Available: https://ubeam.com.

[24] [En línea]. Available: https://www.kickstarter.com/projects/1071086547/ampy-power-your-devices-from-your-motion?lang=es.

[25] M. Schott, «K-Nearest Neighbors (KNN) Algorithm for Machine Learning,» 22 Abril 2019. [En línea]. Available: https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26.

[26] A. k.-n. n. r. i. s. p. recognition y D. Coommans, «Academia,» 1982. [En línea]. Available: https://www.academia.edu/4616250/Alternative_k-nearest_neighbour_rules_in_supervised_pattern_recognition_Part_2._Probabilistic_classification_on_the_basis_of_the_kNN_method_modified_for_direct_density_estimation?auto=download.

[27] T. SRIVASTAVA, «Analytics Vidhya,» 26 Marzo 2018. [En línea]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.

[28] «Metodos de Remuestreo,» [En línea]. Available: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/Boots/tema1BooPres.pdf.

[29] J. R. Berrendero, «Clasificación y regresión logística,» [En línea]. Available: http://verso.mat.uam.es/~joser.berrendero/cursos/Matematicas-e2/e2-tema4-16.pdf.

[30] T. s. M. d. V. d. S. (SVM) y E. J. S. Carmona, «Universidad Nacional de Educación a Distancia,» 11 Julio 2014. [En línea]. Available: http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona]%20SVM.pdf.

[31] T. o. S. V. M. (SVM) y V. Jakkula, «Semantic Scholar,» 2011. [En línea]. Available: https://pdfs.semanticscholar.org/7cc8/3e98367721bfb908a8f703ef5379042c4bd9.pdf.

[32] M. L. y. S. V. M. p. e. t. e. dinero y J. Álvarez, «Analitícaweb,» 22 Diciembre 2016. [En línea]. Available: https://www.analiticaweb.es/machine-learning-y-support-vector-machines-porque-el-tiempo-es-dinero-2/.

[33] S. Glen, «Statistics How To,» 21 Noviembre 2017. [En línea]. Available: https://www.statisticshowto.datasciencecentral.com/mahalanobis-distance/.

[34] M. Riquelme, «Web y Empresas,» 24 Julio 2018. [En línea]. Available: https://www.webyempresas.com/desviacion-estandar-o-tipica/.

[35] J. G. Jiménez, «Jesús García Jiménez,» 22 Enero 2010. [En línea]. Available: https://jesusgarciaj.com/2010/01/22/la-curva-de-distribucion-normal/.

[36] P. SHARMA, «Analytics Vidhya,» 19 Agosto 2019. [En línea]. Available: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/.

[37] C. CABELLO, «Nobbot,» 13 Julio 2016. [En línea]. Available: https://www.nobbot.com/redes/fog-computing/.

[38] M. Boehm, I. Antonov, M. Dokter, R. Ginthoer, K. Innerebner, F. Klezin, S. Lindstaedt, A. Phani y B. Rath, «Arxiv,» 6 September 2019. [En línea]. Available: https://arxiv.org/abs/1909.02976.