

DESPLIEGUE Y GESTIÓN DE PLATAFORMAS BASADAS EN FOG COMPUTING EN EL SENO DE LA CIUDAD INTERCONECTADA

CHRISTIAN ÁLVAREZ SÁNCHEZ

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin Máster en Ingeniería Informática

Madrid, 22 de febrero de 2018

Directores:

José Luis Risco Martín
José Luis Ayala Rodrigo

Convocatoria de febrero de 2018

Calificación: 8 - Notable

Autorización de difusión

Christian Álvarez Sánchez

31 de Enero de 2018

El abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: Despliegue y gestión de plataformas basadas en fog computing en el seno de una ciudad interconectada, realizado durante el curso académico 2017-2018 bajo la dirección de José Luis Risco Martín y José Luis Ayala Rodrigo del Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Resumen en castellano

Despliegue y gestión de plataformas basadas en fog computing en el seno de una ciudad interconectada.

El presente proyecto aborda una tecnología relativamente reciente como es *fog computing*. Esta tecnología aprovecha toda la potencia de *Internet of Things* pero añadiendo un procesamiento de datos justo antes de enviarlos a la nube, también conocido como *edge computing*, con el fin de añadir al potencial de *IoT* una baja latencia en envío y recuperación de información. A través de la fabricación de diferentes tipos de dispositivos de bajo coste energético, económico y fácilmente replicables, y de su instalación en una *smart city* como es Ciudad Universitaria se logra la recuperación de información valiosa con el fin de estudiarla. Estos dispositivos están formados por sensores, tanto infrarrojos como de ultrasonidos, donde su tarea principal es la de recopilar datos sobre el tráfico rodado y el tránsito de personas. A continuación, los dispositivos envían los datos en bruto a través de ondas de baja frecuencia a una red en la que se redirigirán hacia nuestro servidor donde sufrirán un procesamiento y donde se añadirá más información del entorno de la *smart city*. Luego, estos datos serán utilizados para entrenar un modelo predictivo con el fin de conocer cuál será la afluencia de personas y de vehículos (públicos y privados) en un determinado día y con unas determinadas condiciones.

Palabras clave

Fog computing, smart city, cloud, internet of things, sensor, machine learning.

Abstract

Deployment and management of platforms based on fog computing within an interconnected city.

This project deals with a relatively recent technology such as fog computing. This technology takes full advantage of the Internet of Things but adds data processing just before sending them to the cloud, also known as edge computing, in order to add to the potential of IoT a low latency in sending and retrieving information. Through the manufacture of different types of devices with low energy cost, economic and easily replicable, and its installation in a smart city such as Ciudad Universitaria, the recovery of valuable information is achieved in order to study it. These devices are made up of sensors, both infrared and ultra-sound, where their main task is to collect data on road traffic and people transit. Next, the devices send the raw data through low frequency waves to a network where they will be redirected to our server, they will undergo processing and more information about the smart city environment will be added. Then, these data will be used to train a predictive model in order to know what will be the affluence of people and vehicles (public and private) on a given day and with certain conditions.

Keywords

Fog computing, smart city, cloud, internet of things, sensor, machine learning.

Índice general

Índice	I
List of Figures	III
List of Tables	IV
Agradecimientos	V
Dedicatoria	VI
1. Introducción	1
1.1. Introducción	1
1.2. Introduction	3
2. Estado del arte	7
2.1. Internet of things y fog computing	7
2.2. Aplicaciones	11
2.2.1. Agricultura	12
2.2.2. Fog computing en e-Health	13
2.2.3. Gestión energética en domótica	15
2.2.4. Fog computing en smart cities	16
3. Desarrollo del proyecto	18
3.1. Componentes	18
3.1.1. LaunchPads CC1310 y CC1350	18
3.1.2. Sigfox	21
3.1.3. Sensor ultrasonidos	22
3.1.4. Sensor infrarrojo	24
3.2. Desarrollo técnico	25
3.2.1. Desarrollo	25
3.3. Recopilación de datos	31
3.3.1. Machine learning	33
3.3.2. Algoritmos	36
3.3.3. Validación	40
3.4. Planificación del proyecto	40

4. Resultados	42
4.1. Tráfico rodado	42
4.1.1. Naive Bayes	43
4.1.2. Support vector machines	43
4.1.3. Redes Neuronales	43
4.1.4. Análisis discriminante	43
4.1.5. k vecinos más cercanos	44
4.1.6. Árbol de decisión	44
4.1.7. Árbol de decisión <i>Boosted</i>	44
4.1.8. Árbol de decisión <i>Bagged</i>	45
4.2. Tránsito de personas	45
4.2.1. Validación cruzada	45
4.3. Resultados	46
5. Conclusiones y trabajo futuro	47
5.1. Conclusiones	47
5.2. Conclusions	48
5.3. Trabajo futuro	49
Bibliografía	53

Índice de figuras

2.1. Internet of things	8
2.2. Arquitectura de Fog Computing	10
2.3. Arquitectura de Fog Computing	12
3.1. Frecuencias ISM	20
3.2. LaunchPad CC1310	21
3.3. Sigfox	22
3.4. LV-MaxSonar-EZ1	23
3.5. Rango del sensor de ultrasonidos	24
3.6. Sensor infrarrojo: Sharp GP2Y0A710K0F	25
3.7. Sensor infrarrojo: Panasonic AMG8853	25
3.8. Dispositivo ultrasonido para tráfico rodado	27
3.9. Capucha impresa en 3D	28
3.10. Dispositivo abierto	29
3.11. Dispositivo cerrado	30
3.12. Dispositivo ultrasonido para tráfico rodado	31
3.13. Colocación del dispositivo para personas	32
3.14. Diagrama Gantt: Planificación del proyecto	41

Índice de tablas

3.1.	Tabla Datos donde se guardan los datos del tráfico	34
3.2.	Tabla Meteo usada para entrenamiento de machine learning	35
3.3.	Tabla personas usada para entrenamiento de machine learning	36
4.1.	Resultados Naive Bayes	43
4.2.	Resultados SVN	43
4.3.	Resultados utilizando redes neuronales	43
4.4.	Resultados utilizando Análisis discriminante	44
4.5.	Resultados utilizando k vecinos más próximos	44
4.6.	Resultados utilizando árbol de decisión	44
4.7.	Resultados utilizando árbol de decisión <i>Boosted</i>	45
4.8.	Resultados utilizando árbol de decisión <i>Bagged</i>	45
4.9.	Resultados de los algoritmos aplicando validación cruzada.	46

Agradecimientos

Quería expresar mi agradecimiento y mi más profunda admiración a mis dos directores durante este proyecto, José Luis Risco Martín y a José Luis Ayala Rodrigo. Gracias por todo el apoyo y disponibilidad a lo largo del proyecto sabiendo guiarme y ayudarme en todas y cada una de mis dudas. Me habéis descubierto un nuevo camino en este universo llamado informática. Gracias a la Universidad Complutense de Madrid y en concreto a la Facultad de Informática, por haberme dado la posibilidad de estudiar aquello que me gusta y me apasiona. Ha sido un largo camino y parece que ha llegado a su fin aunque seguro que nuestros caminos se vuelven a unir en un futuro. Gracias a los compañeros con los que he compartido prácticas, tardes de estudio, cafés y cervezas y conversaciones, a menudo incoherentes, sobre la mejor manera de indentar código. Ahora empieza lo mejor. Muchas gracias a la persona que me saca de vez en cuando de este mundo tan absorbente que es la ingeniería. Ángela, te debo estar donde estoy. Y gracias a mis padres, Jesús y Nila, a mi hermano, Jorge y al resto de mi familia por el interés y la paciencia mostrada durante toda mi etapa como universitario. Como ya dije en alguna otra ocasión: "Gracias papá, gracias mamá, gracias por mi vida".

-Christian Álvarez Sánchez-

Dedicatoria

Dedicado con todo mi cariño a Leonardo, Isaac, Leonhard, Ada, Carl, Hipatia, Max, Michael, Marie, Carolina, James, Alan y al hijo que algún día tendré.

*Si he logrado ver más lejos, ha sido porque he
subido a hombros de gigantes.*

Isaac Newton

Capítulo 1

Introducción

En un agujero en el suelo, vivía un hobbit.

El Hobbit, J.R.R. Tolkien

1.1. Introducción

Hoy en día, estamos cada vez más conectados con nuestro entorno. La presencia en la sociedad de *smartphones*, *tablets* y otro tipo de *smart devices* es común es nuestra vida diaria para poner en contacto a personas como podría ser el uso de redes sociales como facebook o twitter. De esta concepto, surgió la posibilidad de no solo conectar personas, sino también establecer conexión con otros elementos como electrodomésticos, vehículos o paquetería con el fin de gestionar los mismos. En este punto, surge el internet de las cosas, también conocido como *internet of things* o por sus siglas IoT. Esta tecnología propone la interconexión de objetos cotidianos a través de internet, donde la conexión se realizaría, por ejemplo, mediante elementos de radiofrecuencia y que podrían ser gestionados por un procesador central situado en la nube.

Internet de las cosas, o *Internet of things (IoT)* por sus siglas en inglés, aún siendo una tecnología relativamente joven, goza de un desarrollo fructífero. Prueba de ello son los avances en la interacción con nuestras casas, donde podemos gestionar, a distancia desde nuestro teléfono móvil, la hora a la que se encenderá la calefacción o el horno. Otro ejemplo lo tenemos en los vehículos, donde no solo podremos conocer la situación del mismo, sino

su estado mecánico a través de aplicaciones móviles. Esta conexión no solo busca nuestra comodidad sino también nuestra seguridad. La interacción con nuestro entorno unida a nuestro estado de salud, medido a través de *smart devices*, ofrece una ingente cantidad de datos aprovechables que pueden ser utilizados para prevenir enfermedades o avisar de situaciones de emergencia.

Sin embargo, esta conectividad con lo que nos rodea conlleva varios desafíos: En primer lugar, la capacidad necesaria para recuperar los datos, que luego serán utilizados para crear información útil a la hora de la toma de decisiones o ejecutar acciones frente a los eventos que han producido esos datos. En segundo lugar, la necesidad de un procesamiento en la nube con una latencia aceptable que pueda hacer frente al volumen de datos enviados que normalmente será masivo y por último un consumo energético reducido (por parte de los sensores) para que la implantación de esta tecnología sea viable. Internet de las cosas puede hacer frente a estos desafíos mientras que los datos enviados a la nube no sobrepasen cierto umbral. Si se diera el caso de sobrepasarlo, la latencia en la respuesta por parte de la nube no sería aceptable y se perdería, por ejemplo, la respuesta en tiempo real.

Fog computing viene a dar solución al problema de la latencia. Es un nuevo modelo, dentro de internet de las cosas que previene el envío masivo de datos preprocesándolos en el borde de la red. El concepto es simple pero eficaz. Una red de sensores envían los datos a un nodo que trata los datos en primera instancia para eliminar aquellos erróneos, dar formato adecuado y ofrecer un pequeño procesamiento antes de enviarlos a la nube. Con estos pasos, el volumen de información destinada a la nube se reduce, disminuye en el tiempo de latencia consiguiendo una reducción en el tiempo total de operación y una forma más racional y eficiente de trabajar con los datos generados.

Una de las finalidades tanto de *IoT* como *fog computing* es el descubrimiento de modelos o patrones de comportamiento con el fin de poder mejorar la gestión en un ámbito definido. Por ejemplo, a través del estudio de la circulación de vehículos por determinadas calles, se pueden definir cuales son las horas de más afluencia de tráfico rodado y así establecer una

gestión de los semáforos más eficiente.

Aplicando los conceptos antes descritos, presentamos una aplicación basada en *fog computing*. Se trata de un proyecto que intenta demostrar las posibilidades que ofrece *fog computing* en la búsqueda de patrones en ciudades inteligentes o *smart cities*. En este caso, usaremos como escenario el campus de Ciudad Universitaria de la Universidad Complutense de Madrid. Este proyecto consta de dos partes bien diferenciadas: por un lado tendremos el apartado de volumen de tráfico rodado, en el cual utilizaremos sensores de ultrasonidos, donde la finalidad es conocer el flujo de vehículos, distinguiendo entre tráfico privado (automóviles) y tráfico público (autobuses). Por otro lado, la segunda parte del proyecto es el apartado de volumen de personas que acceden y salen de ciudad universitaria, donde utilizaremos sensores infrarrojos.

Cuando esté instalada toda la infraestructura, esta comenzará a tomar mediciones periódicas del flujo de personas y vehículos en puntos determinados con el objetivo de conocer cuál es la afluencia. La forma de tomar estas mediciones, será mediante ultrasonidos e infrarrojos que contarán el número de individuos y vehículos (distinguiendo entre coches y autobuses) que pasan por ellos. Los sensores y los dispositivos que los contienen serán distribuidos en las entradas y salidas de Ciudad Universitaria (metro y carreteras). Una vez recogida la información se enriquecerá tomando datos meteorológicos de las mediciones tales como lluvia, viento, temperatura, contaminación, etc., así como el día y la hora en la que se tomaron los datos. A continuación, la información se procesará utilizando el modelo de *fog computing* el borde de la red. Todo ello unido nos dará la posibilidad de poder establecer patrones de tráfico rodado y personas así como poder predecir que volumen de los mismos en un día determinado y con unas condiciones meteorológicas determinadas.

1.2. Introduction

Nowadays, we are increasingly connected with our environment. The presence in society of smartphones, tablets and other smart devices is common in our daily life to connect

people such as the use of social networks such as facebook or twitter. From this concept, the possibility arose of not only connecting people, but also establishing a connection with other elements such as electrical appliances, vehicles or parcels in order to manage them. At this point, comes the internet of things, also known as Internet of Things or by its acronym IoT. This technology proposes the interconnection of everyday objects through the internet, where the connection would be made, for example, by means of radio frequency elements and that could be managed by a central processor located in the cloud.

Internet of things, or Internet of things (IoT) by its abbreviations in English, although being a relatively young technology, enjoys a fruitful development. Proof of this are the advances in the interaction with our houses, where we can manage, at a distance from our mobile phone, the time at which the heating or the oven will turn on. Another example is in vehicles, where we can not only know the situation of the vehicle, but also its mechanical status through mobile applications. This connection not only seeks our comfort but also our security. The interaction with our environment linked to our state of health, measured through smart devices, offers a huge amount of usable data that can be used to prevent diseases or warn of emergency situations.

However, this connectivity with what surrounds us entails several challenges: First, the necessary capacity to recover the data, which will then be used to create useful information when making decisions or executing actions in the face of events that they have produced that data. Secondly, the need for a cloud processing with an acceptable latency that can cope with the volume of data sent that will normally be massive and finally a reduced energy consumption (by the sensors) so that the implementation of this technology be viable The Internet of Things can face these challenges as long as the data sent to the cloud does not exceed a certain threshold. If the case were to surpass it, the latency in the response by the cloud would not be acceptable and, for example, the response in real time would be lost.

Fog computing comes to solve the problem of latency. It is a new model, within the internet of things that prevents the massive sending of data by preprocessing them at the

edge of the network. The concept is simple but effective. A network of sensors sends the data to a node that treats the data in the first instance to eliminate the erroneous ones, to format properly and to offer a small processing before sending them to the cloud. With these steps, the volume of information destined to the cloud is reduced, it decreases in the latency time, achieving a reduction in the total time of operation and a more rational and efficient way of working with the generated data.

One of the purposes of both IoT and fog computing is the discovery of models or patterns of behavior in order to improve management in a defined area. For example, through the study of the circulation of vehicles through certain streets, it is possible to define which hours are the most traffic flows and thus establish a more efficient traffic management.

Applying the concepts described above, we present an application based on fog computing. It is a project that tries to demonstrate the possibilities offered by fog computing in the search for patterns in smart cities or smart cities. In this case, we will use as a scenario the campus of the Ciudad Universitaria of the Complutense University of Madrid. This project consists of two well differentiated parts: on the one hand we will have the volume of road traffic section, in which we will use ultrasonic sensors, where the purpose is to know the flow of vehicles, distinguishing between private traffic (cars) and traffic public (buses). On the other hand, the second part of the project is the volume section of people who access and leave the Ciudad Universitaria, where we will use infrared sensors.

When all the infrastructure is installed, it will begin to take periodic measurements of the flow of people and vehicles at certain points in order to know what the influx is. The way to take these measurements will be through ultrasound and infrared that will count the number of individuals and vehicles (distinguishing between cars and buses) that pass through them. The sensors and the devices that contain them will be distributed at the entrances and exits of the Ciudad Universitaria (metro and roads). Once the information is collected, it will be enriched by taking meteorological data from measurements such as rain, wind, temperature, pollution, etc., as well as the day and time at which the data was

taken. Next, the information will be processed using the fog computing model at the edge of the network. All this together will give us the possibility of being able to establish traffic patterns and people as well as being able to predict their volume on a given day and with certain meteorological conditions.

Capítulo 2

Estado del arte

No he perdido contra la dificultad, si no contra el tiempo.

Leonardo di ser Piero da Vinci

2.1. Internet of things y fog computing

Cuando aludimos a *Internet of Things* o *IoT*, hacemos referencia al conjunto de dispositivos físicos y electrónicos que tienen integrados sensores, actuadores, software y sistemas de conectividad con el fin de recopilar e intercambiar información.

El término fue acuñado en 1999 por Kevin Ashton³ en el Auto-ID Center del MIT. Kevin Ashton trabajaba con identificación por radio frecuencia o RFID y vio en esta tecnología el requisito necesario para que *IoT* diera un paso adelante. Él pensaba que, si todos los objetos que nos rodean (electrodomésticos, automóviles, paquetería, etc) en nuestra vida diaria fueran equipados con RFID los ordenadores podrían controlar tanto el inventario como el estado actual de los mismos. Además, esta tecnología no estaría solo supeditada a RFID si no que podría utilizar otro tipo de comunicación como códigos de barras, códigos QR o conexión a internet mediante *wifi*. Sin embargo, y aunque el término proviene de 1999, no fue hasta 2008-2009⁶ cuando *IoT* surgió realmente con el avance de la tecnología y acceso mundial a internet.

También tenemos que tener en cuenta que el número de dispositivos conectados entre sí aumenta día a día. Concretamente, en 2011, el número de dispositivos conectados superó

al número de personas que habitan la tierra⁷. Este incremento en el número de dispositivos lleva consigo el aumento, al mismo ritmo, de las identificaciones de dispositivos y aunque no todos estén conectados directamente a internet, muchos de ellos sí. Esto ha provocado que las direcciones IPv4 sean insuficientes y se tenga que pasar a tener direcciones IPv6 mucho más numerosas. El número de dispositivos sigue aumentando y se estima que para 2020 la cantidad ascienda a 50 mil millones de dispositivos. *IoT* tiene efectos notables tanto en el



Figura 2.1: Internet of Things

trabajo como en el hogar, donde puede desempeñar un papel importante en un futuro próximo (vida asistida, domótica, e-health, transporte inteligente, etc.). También trae consigo importantes avances en el mundo empresarial como podrían ser: logística, automatización industrial, transporte de bienes, seguridad, etc).

Por otro lado, *IoT* va ligada a la tecnología de *cloud computing* ya que la unión entre ellas permite el almacenamiento, el procesamiento y la recuperación de información con cierta velocidad y accesibilidad siempre que haya conexión, es decir, bajo demanda. Igualmente, nos

permite configurar recursos tales como el almacenamiento, capacidad de cómputo, número de servidores, aplicaciones, etc., a precios asequibles. Por lo tanto, *cloud computing* nos ofrece un almacenamiento y una capacidad de computación prácticamente ilimitados a muy bajo coste lo que ha producido un avance en las demás tecnologías, en este caso IoT. Sin embargo, el aumento de dispositivos interconectados formando una red de redes y, consecuentemente, el aumento de sensores y actuadores puede llegar a aumentar los tiempos de respuesta ante el procesamiento, envío y recuperación de la información. En este punto es donde *fog computing* puede solucionar esta cuestión.

Fog Computing es una infraestructura distribuida en la que ciertos procesos o servicios de aplicaciones se gestionan en el borde (también conocido como *edge*) de la red mediante un dispositivo inteligente, pero otros se gestionan aún en la nube. Es esencialmente una capa intermedia entre la nube y el hardware para permitir un procesamiento, análisis y almacenamiento de datos más eficientes, lo que se logra reduciendo la cantidad de datos que hay que transportar a la nube.

Si bien la nube es un ingrediente perfecto para el Internet de las cosas, no todos los escenarios de IoT pueden aprovecharla. Las soluciones IoT industriales exigen baja latencia y un procesamiento inmediato de los datos. Las organizaciones no pueden permitirse el retraso causado por la ida y venida entre la capa de dispositivos y las plataformas IoT basadas en la nube. La solución requiere un procesamiento instantáneo de los flujos de datos con una rápida respuesta. Cisco ofreció uno de los primeros motores en el mercado destinado a *fog computing*. La compañía se acredita haber acuñado el término incluso antes de que IoT se convirtiera en una palabra de moda. Cisco colocó Fog como la capa para reducir la latencia en los escenarios de nubes híbridas. El resultado del interés de Cisco, lo llevó a fundar un consorcio, el 19 de noviembre de 2015, Cisco Systems, ARM Holdings, Dell, Intel, Microsoft y la Universidad de Princeton, fundaron el Consorcio OpenFog, para promover los intereses y el desarrollo en *fog computing*⁸.

The Internet of Thing Architecture and Fog Computing

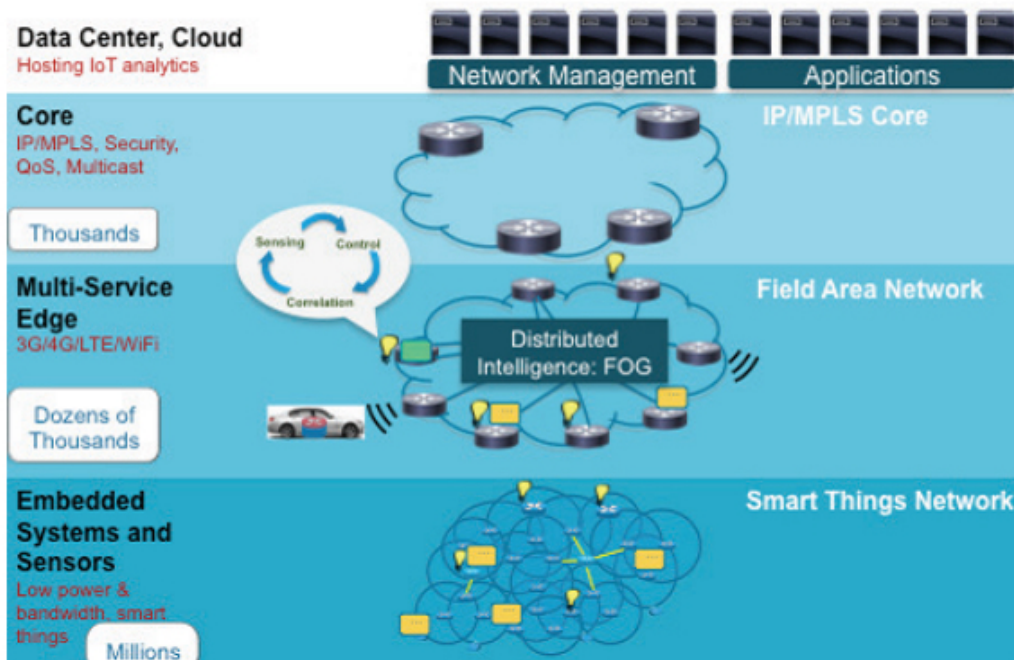


Figura 2.2: Fog Computing Fuente: Cisco

La arquitectura de *fog computing*¹⁰ consta de diferentes capas como se puede ver en la figura 2.3

- Rendimiento: la baja latencia es uno de los motivos principales para adoptar este tipo de arquitectura.
- Seguridad: la seguridad extremo a extremo es fundamental para el éxito de la implementación. Por lo tanto, todas las capas de la arquitectura (software y hardware) deben cumplir los requisitos de seguridad.
- Capacidad de gestión.
- Análisis y control de datos: la capacidad de los nodos para ser autónomos requieren tanto un análisis de datos localizados como un control de los mismos que deben ocurrir en el nivel de la arquitectura adecuado.

- Infraestructura hardware: deben proporcionar un soporte robusto y protección para sus componentes externos ya que en muchos despliegues deben sobrevivir a condiciones adversas
- Seguridad en el nodo: se debe tener la información necesaria para determinar el examen físico apropiado en cuanto a medidas de seguridad, el método óptimo para establecer y mantener la confianza, y qué tipo de políticas poner en marcha para que el nodo actúe con seguridad
- Gestión del nodo: Se refiere a los sistemas de administración que no están ejecutándose en el sistema operativo. También son llamados a veces *Hardware Platform Management (HPM)*
- Virtualización de Hardware y contenedores: estos mecanismos basados en hardware están disponibles en todos los procesadores que se usan para este tipo de plataformas. Juegan un papel importante en la seguridad, en la entrada/salida y cálculo.
- Gestión del nodo: funcionamiento general del nodo y sus comunicaciones con otros nodos y sistemas.
- Soporte de aplicaciones: son arquitecturas software que no tienen un uso determinado pero que ayudan y apoyan a otras aplicaciones.
- Aplicaciones de servicio: es la capa que ofrece los requisitos y casos de uso finales y resuelve las necesidades específicas del dominio.

2.2. Aplicaciones

En este punto, daremos un vistazo a algunas aplicaciones o estudios que demuestran y representan la aplicación de *fog computing*. Aunque sea esta una tecnología relativamente nueva, no impide que ya se esté aplicando para probar sus posibilidades y, como veremos

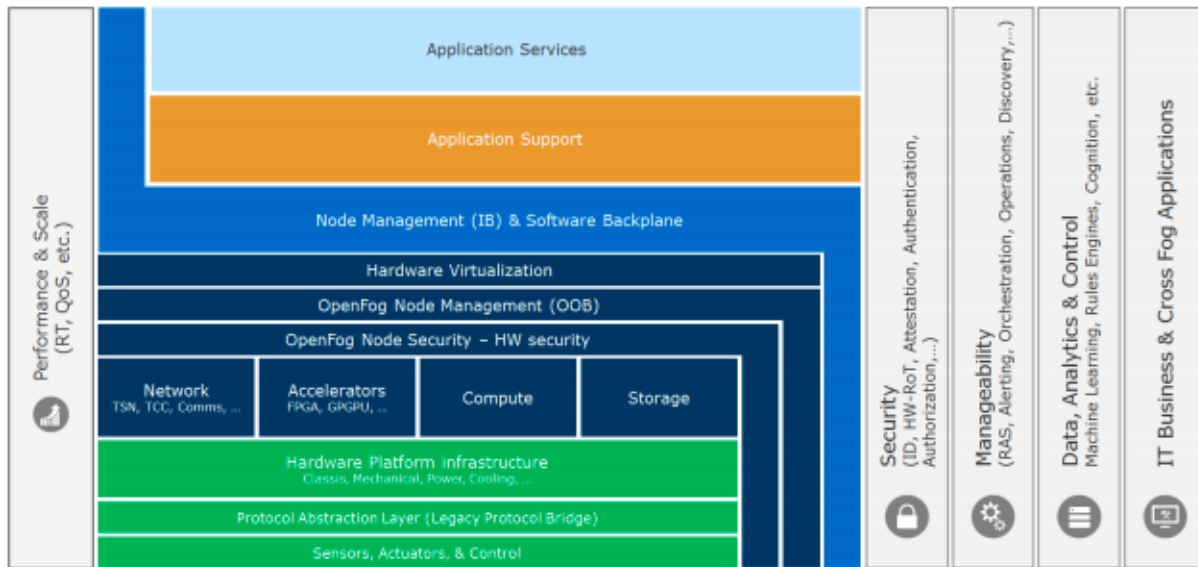


Figura 2.3: Arquitectura de Fog Computing

a continuación, su utilización tiene cabida tanto en agricultura, como en bienestar y salud, pasando por la gestión, control y estudio de modelos de una ciudad.

2.2.1. Agricultura

Dentro de las *smart cities*, uno de los dominios a tener en cuenta es la alimentación de la población. El crecimiento de los núcleos urbanos obliga a un mayor control y predicción de las necesidades de los ciudadanos con el fin de asegurar su manutención. Por esta razón, se debe aumentar el rendimiento y la eficiencia de los procesos que generan alimento.

Uno de los ejemplos más claros de *IoT* utilizando *fog computing*¹³ es Phenonet¹⁴. Este proyecto, desarrollado en Australia por la Commonwealth Scientific and Industrial Research Organisation (CSIRO) y OpenIoT⁴ tenía como objetivo aumentar la eficiencia y la eficacia de los cultivos controlando su crecimiento, el punto exacto de madurez y la predicción de heladas o sequías.

El sistema estaba formado por una red extensa de varios tipos de sensores encargados de recopilar información del entorno y de los cultivos. El primer tipo de sensor se encargaba de monitorizar el estado de las plantas desde la propia plantación tales como niveles de

nutrientes del suelo, incidencia del sol, etc. El segundo sensor, iba acompañado de un globo que tomaba mediciones atmosféricas, tales como temperatura o humedad. Y por último, un tercer sensor colocado en maquinaria agrícola que controlaría el crecimiento de las plantas. Todos ellos se encargan de recopilar el volumen de datos necesario, normalmente masivo, para poder representar la realidad correctamente.

En este punto es donde se pone de manifiesto el fog computing. En lugar de enviar toda la información a la nube para ser almacenada y procesada, el tratamiento de la información ocurre en los mismos sensores o en dispositivos *edge* que funcionarían como una pasarela o *gateway*. Esto conlleva las ventajas de que se puede desechar información errónea (algún sensor da lecturas incorrectas) y no depender de la nube para todo el procesamiento de la información con lo que disminuimos la latencia.

La aplicación de *fog computing* a la agricultura da como resultado un mayor control sobre el cultivo. Se podrían predecir heladas u olas de calor, conocer el punto óptimo de recogida (por el día o por la noche) o modificar los fertilizantes e insecticidas para una mayor calidad del producto.

2.2.2. Fog computing en e-Health

IoT está tomando una gran relevancia los últimos años con el avance de las tecnologías. Los *smartphones* se han convertido en objetos de uso cotidiano y, al igual, las pulseras inteligentes para medir nuestras constantes vitales tanto en nuestra vida diaria como a la hora de realizar ejercicio.

Tomemos, por ejemplo, el ámbito del entrenamiento físico. Una persona puede llevar consigo una pulsera o dispositivo que mida sus constantes vitales en todo momento. En este punto, puede no interesar que los datos recuperados en reposo sean procesados y enviados a la nube ya que carecen de interés. Por el contrario, si se encuentra en un gimnasio si es de utilidad la información recogida. El dispositivo, en este caso, tomará las mediciones adecuadas y las enviará, por ejemplo, al *smartphone* que funcionaría como instrumento de

fog computing. El teléfono móvil procesará y almacenará temporalmente los datos y, cuando tenga acceso a internet, serán enviados a la nube donde se realizarán informes o resúmenes para una recuperación posterior.

En el campo de la salud, *IoT*, está siendo profundamente aplicado. Varios estudios muestran que la aplicación de esta tecnología mejora la fiabilidad y la latencia. Por ejemplo, el proyecto de *fog computing* E-HAMC¹ llevado a cabo en 2015 muestra como *fog computing* puede dar solución a los problemas enunciados anteriormente de fiabilidad y latencia. Los sistemas tradicionales de actuación ante emergencias sanitarias normalmente se ejecutaban de manera activa, es decir, el individuo tenía que realizar una llamada al servicio de emergencia adecuado o a familiares provocando en primer lugar un retardo en la posible respuesta y en segundo la posibilidad de no realizar correctamente el aviso, ya sea a estos servicios o a familiares. Aplicando *fog computing* se consigue que el dispositivo de salud recoja, analice, filtre y se suban a la nube de manera automática, mejorando el tiempo del proceso. E-HAMC, concretamente, contenía una lista de familiares a los que avisar en caso de emergencia y, además, avisar al servicio adecuado, por ejemplo, urgencias en cardiología. Otra de las ventajas, es la de conocer la situación exacta mediante GPS. Este dispositivo *fog* contiene una interfaz básica donde da la opción de elegir el tipo de emergencia al que se está haciendo frente: accidente, asesinato, incendio, derrumbe de edificio, terrorismo, robo y emergencia sanitaria.

Como se mencionó anteriormente, E-HAMC como dispositivo *edge*, realiza el procesamiento y actuación necesarias dependiendo de la emergencia. Una vez se ha realizado este proceso, los datos se pueden enviar a la nube para que otras entidades puedan usar esos datos para coordinarse, planificar o realizar informes. Por ejemplo, si una zona de una determinada ciudad es propensa a los robos, se pueden utilizar los datos de E-HAMC para colocar más policía y mejorar la seguridad. Del mismo modo, si hay algún punto donde los accidentes son más propensos a ocurrir se puede mejorar la iluminación, señalización y, además, poner un punto de acceso rápido a recursos sanitarios como podrían ser ambulancias.

Por último, E-HAMC, puede funcionar de manera autónoma. Pensemos ahora en una persona con problemas cardíacos. Puede darse la posibilidad de que no sea capaz por sí misma de activar un aviso de emergencia. En este punto, la aplicación actuaría de manera autónoma, enviando los correspondientes avisos.

Como vemos por estos ejemplos, *fog computing*, permite una mejora en el ámbito de la salud muy notable que puede servir tanto para ayudar a personas en situaciones de emergencia como para aplicarlo en las *smart cities* con el fin de mejorar su funcionamiento.

2.2.3. Gestión energética en domótica

IoT también está presente en los hogares. La reducción de costes de fabricación de sensores, actuadores, adaptadores de red, etc., unido al aumento de rendimiento de los mismos los han hecho muy atractivos a la hora de incorporarlos a nuestros hogares. Además, la cantidad de energía necesaria para poner en funcionamiento el entorno domótico se ha reducido notablemente, lo cual hace más atractivos estos sistemas. Esta red estará formada por sensores situados estratégicamente que pueden medir, temperatura, intensidad lumínica, humedad, presencia humana, detección de gases, etc., que dan una visión global sobre el estado y situación de la casa. Estos datos serán enviados a la nube donde se almacenarán y procesarán. Asimismo, la red, contendrá una serie de actuadores que responderán después de haber recuperado los datos desde la nube.

Este sistema es válido y eficiente si tenemos una red domótica con pocos sensores y actuadores. Sin embargo, la escalabilidad resulta un problema cuando pasamos a tener muchos sensores y actuadores con el resultado de obtener unas latencias muy altas que infringen los requisitos de tiempo⁵. Aquí es donde la tecnología de *fog computing* puede solucionar este tipo de problemas.

Como se ha comentado antes, *fog computing*, acerca el procesamiento al límite de la red (*edge*), por lo que libera a la nube de un gran volumen de procesamiento. Como se demuestra en el artículo *Energy Management-as-a-Service Over Fog Computing Platform*²,

la implementación de un sistema domótico o *Home Energy Management (HEM)* sobre *fog* es perfectamente viable. En su caso, han monitorizado y medido el consumo energético de cada dispositivo (desde el frigorífico hasta el coche eléctrico) y gestionado el consumo de los mismos dando como resultado un sistema de sistemas. Siguiendo la filosofía de *IoT* han elaborado un sistema adaptativo y escalable utilizando dispositivos con diferentes tipos de conexión como bluetooth o ethernet. Lo mismo ocurre con los actuadores y sensores, no se han utilizado todos del mismo fabricante y se han tenido que adaptar a su sistema. Además, el sistema contiene un dispositivo de computación, llamado *HEM control panel*. El principal objetivo es descubrir y monitorizar los dispositivos de la plataforma dinámicamente, envío, recepción y procesamiento de datos y activar los dispositivos de acuerdo con estos datos.

Por lo tanto, agrupando todos estos sistemas se ha demostrado que *fog computing* puede ser utilizado en domótica aportando todos los beneficios de *cloud computing* e *IoT* y además solucionando los problemas de latencia y sobrecarga de información poco útil o errónea.

2.2.4. Fog computing en smart cities

Por último, veremos como se puede aplicar *IoT* y *fog computing* en ciudades. Un estudio llevado a cabo en Barcelona: “A new era for cities with fog computing” ¹⁵ muestra detenidamente cuales son las posibilidades de IoT dentro de una urbe. En primer lugar, se instalaron cabinas a lo largo de la ciudad que gestionaran su energía automáticamente, y a su vez se encargase de los problemas que pudieran surgir como la interrupción de la conexión con la nube, en definitiva, mantener los servicios operativos. Estas cabinas monitorizaban, almacenaban, procesaban y enviaban datos a la nube según unos determinados eventos. Así pues, las cámaras que grababan el movimiento por la ciudad enviaban los datos a estas cabinas que procesaban las imágenes. En el caso en el que no ocurriera ningún evento reseñable, los datos no se subían a la nube. Sin embargo, si sucedía el evento (ruido, velocidad, personas caminando), se almacenaban, procesaban y enviaban las imágenes justamente anteriores, durante y posteriores al suceso. De esta manera, no se sobrecargaba ni trataba información

poco interesante la nube. A su vez, los sensores dispuestos por las calles pueden monitorizar el tráfico y recuperar, por ejemplo, el número de las matrículas, su velocidad, tamaño, tamaño de los atascos, etc. En este caso, se podrían utilizar los elementos de *fog computing* para analizar en tiempo real el volumen de tráfico rodado y gestionar de manera adecuada los intervalos de los semáforos con el fin de agilizar y descongestionar las calles.

Este ejemplo, podría ser tratado desde un punto de vista fuera de *fog computing* y simplemente aplicar IoT sobre *cloud computing*. Sin embargo, pensemos en una ciudad con cientos de calles, miles de semáforos, donde ocurren muchos eventos a diario. Podría darse el caso que la cantidad de información enviada y procesada por la nube fuera tal que los tiempos de latencia fueran inaceptables, lo que convierte a *fog computing* en una opción real y factible para solucionar estos problemas.

Este proyecto se enmarca en la descripción dada para *smart cities* en la sección 2.2.4 porque pretende monitorizar, almacenar y procesar los datos relativos a Ciudad Universitaria y establecer un modelo de predicción mediante ellos y la utilización de *fog computing*.

Capítulo 3

Desarrollo del proyecto

Pones tu pie en el camino y si no cuidas tus pasos, nunca sabes a donde te pueden llevar.

John Ronald Reuel Tolkien, El Señor de los Anillos

En este capítulo se expondrá el desarrollo del proyecto, mostrando los componentes utilizados, describiendo la arquitectura *IoT*, los algoritmos estudiados así como los desafíos encontrados durante el proceso.

3.1. Componentes

3.1.1. LaunchPads CC1310 y CC1350

La elección del dispositivo *Sub-1Ghz LaunchPad CC1310 y CC1350* no fue al azar. El espectro de frecuencia que se puede utilizar para las comunicaciones por radio está controlado por la Conferencia de Administraciones de Correos y Telecomunicaciones (CEPT) en Europa. Esta organización asigna una serie de bandas de frecuencia para un uso específico así como estándares y certificaciones. Esto significa que el espectro útil en la mayoría de las regiones está supeditado a una licencia que se debe comprar para poder operar, por ejemplo, un transmisor de radio. La Unión Internacional de Telecomunicaciones (UIT-R), coordina la utilización global del uso compartido del espectro de radio y tiene reservado varias bandas de frecuencia para uso industrial, aplicaciones científicas y médicas (ISM). Estas bandas no

necesitan de una licencia para poder operar sobre ellas y varían ligeramente de un país a otro. Las bandas más populares para ISM son 433MHz, 868MHz, 915MHz, 2.4GHz y 5GHz. Como regla general, las bandas de mayor frecuencia ofrecen más canales y más ancho de banda y por lo tanto puede servir a redes más grandes e impulsar el rendimiento en cuanto a datos se refiere. Por otro lado, las ondas de radio de más baja frecuencia se propagan mejor y por lo tanto pueden lograr un mejor rango, especialmente dentro de edificios.

El *Sub-1Ghz LaunchPad CC1310* es un dispositivo formado por un microcontrolador inalámbrico con una frecuencia de radio Sub-1 GHz, lo cual nos ofrece una conectividad de largo alcance y que, además, está combinado con un procesador ARM-Cortex M3 de 32 bits que funciona a 48 MHz. El CC1310 ha sido diseñado para aplicaciones de baja potencia y largo alcance, lo que lo convierte en un candidato perfecto para aplicaciones *IoT*. Por último, este dispositivo contiene unas características periféricas que incluyen un controlador de sensores externos de ultrabaja potencia que resulta ideal para recopilar datos analógicos y digitales de forma autónoma, mientras que el resto del sistema está en modo suspensión. La alimentación del launchpad ha sido diseñada para recibir energía desde una fuente compatible con USB como podría ser una *powerbank* o conectándolo al ordenador directamente.

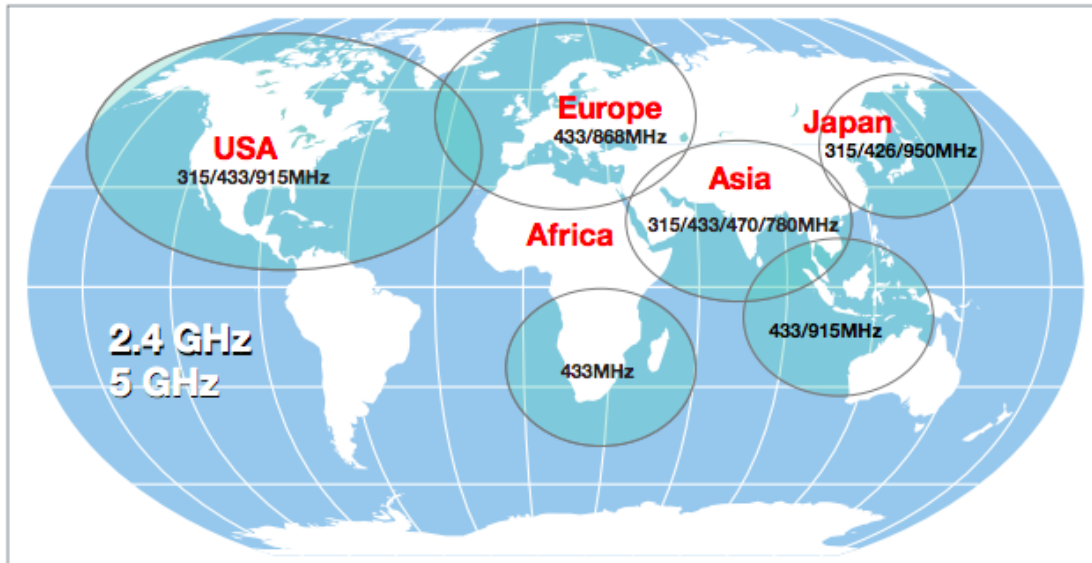


Figura 3.1: Frecuencias ISM

El *CC1310*, al ofrecer conectividad inalámbrica, nos brinda la posibilidad enviar los datos recogidos a Sigfox, una red *wireless* construida para conectar dispositivos de baja energía donde la comunicación se realiza mediante comandos AT. Sigfox esta descrito en profundidad en la sección 3.1.2.

Como resumen, se eligió el *CC1310* porque ofrece:

- Largo alcance: Puede llegar hasta los 20 kilómetros con la potencia de una pila de botón. Si lo ponemos en contexto, con este nivel de energía el *WiFi* alcanzaría alrededor de 100 metros, el *bluetooth*, unos 400 metros, mientras que el *CC1310* llega a más 20 kilómetros en condiciones normales.
- La potencia necesaria es ultrabaja: Como se comentó anteriormente, basta con una pila de botón para conseguir un alcance muy alto en comparación y además, al ser de consumo muy reducido, puede durar más de diez años.
- Conexión robusta: En entornos urbanos donde la densidad de edificios puede llegar a representar un problema, este dispositivo ofrece una conexión robusta y confiable

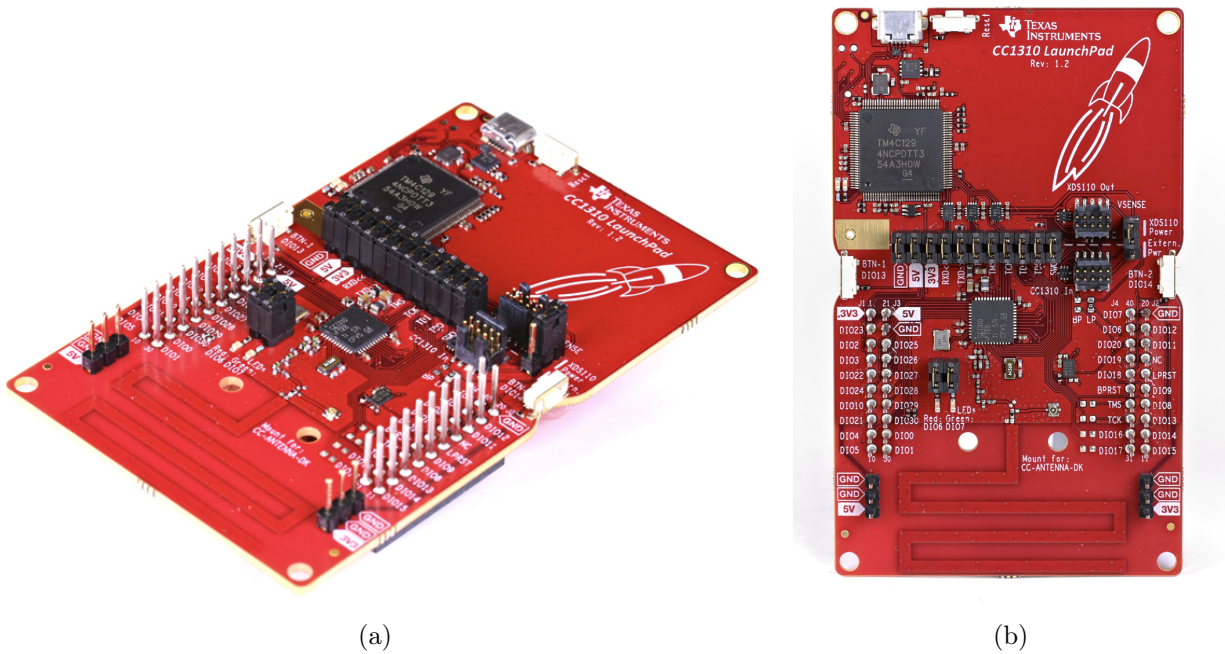


Figura 3.2: LaunchPad CC1310

con un cifrado AES¹ incorporado. Además, ofrece una conectividad que atraviesa con mayor facilidad muros y evita esquinas.

3.1.2. Sigfox

Sigfox es un proveedor de red, conceptualmente similar a un operador de telefonía móvil como Movistar o Vodafone. La red Sigfox está dirigida principalmente a aplicaciones *IoT* y si lo comparamos con un móvil, ofrece un coste y duración de la batería significativamente menor, lo cual son ventajas para las aplicaciones *IoT*. Sigfox utiliza una banda de transmisión Sub-1Ghz, concretamente 868 MHz en Europa que proporciona conectividad de largo alcance (más de 20 kilómetros) entre la estación base y el nodo (topología de estrella). Para lograr un alcance tan amplio, Sigfox debe transmitir a velocidades de datos muy bajas (hasta 100 bps, usando paquetes de 12 bytes, y no más de 140 mensajes por nodo y día) y aunque esta red es bidireccional, la carga útil para transmisiones desde la estación base a los nodos

¹Advanced Encryption Standard

es muy baja, lo que limita el rango de sistemas utilizables. Sin embargo, este enfoque es suficiente para la mayoría de las aplicaciones *IoT* como un detector de humo o un medidor de nivel de agua.



Figura 3.3: Sigfox

En nuestro proyecto hemos usado Sigfox para la recolección de datos desde el dispositivo CC1310 por medio de sus sensores. La información se envía a la nube donde puede ser vista y descargada desde una interfaz gráfica proporcionada por el propio Sigfox. El envío de mensajes desde el dispositivo con Sigfox se realiza mediante comandos AT. En el desarrollo técnico del proyecto se especificará más en profundidad [3.2](#).

3.1.3. Sensor ultrasonidos

El sensor de ultrasonidos es el encargado de tomar las mediciones para el tráfico rodado. A la hora de elegir un sensor adecuado a este objetivo se valoró el utilizar tanto infrarrojo como ultrasonido. Los sensores tenidos en cuenta para nuestro proyecto han sido:

- **LV-MaxSonar-EZ1**⁹: Este sensor de ultrasonidos, concretamente la versión 1010, que podemos ver en la figura [3.4](#) ofrece unas características interesantes para el propósito del proyecto, concretamente las mediciones del tráfico rodado. En primer lugar es un sensor de bajo coste, lo que nos permitía poder escalar el proyecto sin disparar su costo. Además, ofrece un rango fiable y estable en cuanto a datos se refiere y su montaje es simple ya que viene preparado para su uso y con los orificios de montaje definidos.

Otra de las características es que la velocidad de medición es lo suficientemente rápida como para cumplir nuestras necesidades y, por último, su gasto de energía es mínimo, lo cual es perfecto para *IoT*.



Figura 3.4: LV-MaxSonar-EZ1

El rango de este sensor se adaptaba muy bien a nuestras necesidades. Empieza a funcionar adecuadamente a partir de las 6 pulgadas (15,2 cm) y nos da datos fiables hasta 256 pulgadas (6,45 m). Como el sensor va a ir colocado en un semáforo o en una farola y apuntando a los vehículos, este rango es más que suficiente. En la figura 3.5 podemos verlo de forma más clara además de como los voltajes influyen en el rango. Por último, señalar que las lecturas ocurren cada 50 milisegundos (20Hz), lo que nuevamente es más que suficiente para tomar varios datos del mismo vehículo y así aumentar la precisión.

- **Sharp GP2Y0A710K0F¹¹**: Este sensor de infrarrojos, que podemos ver en la figura 3.6 se valoró también en un principio para tomar mediciones de tráfico. Al igual que el de ultrasonidos su rango de mediciones, aunque menor (entre 100cm y 550cm), era suficiente. El consumo también es bastante reducido para este tipo de aplicaciones y también el número de medidas por segundo nos permitía tomar varios valores del mismo vehículo para afinar nuestros datos. Sin embargo este infrarrojo tenía un problema. Al estar situado en exteriores la luz ambiental provocaba que no tomara medidas adecuadas por lo que tuvimos que descartarlo.

MB1010

LV-MaxSonar®-EZ1™ Beam Pattern

Sample results for measured beam pattern are shown on a 30-cm grid. The detection pattern is shown for dowels of varying diameters that are placed in front of the sensor

A 6.1-mm (0.25-inch) diameter dowel

B 2.54-cm (1-inch) diameter dowel

C 8.89-cm (3.5-inch) diameter dowel

D 11-inch wide board moved left to right with the board parallel to the front sensor face.

This shows the sensor's range capability.

Note: For people detection the pattern typically falls between charts A and B.

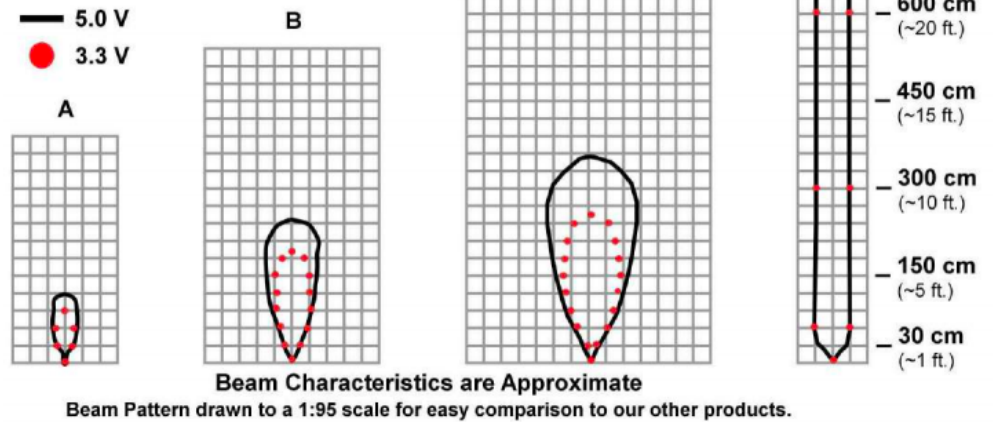


Figura 3.5: Rango del sensor de ultrasonidos

Valorando las cualidades y defectos de ambos sensores se optó por utilizar el sensor de ultrasonidos ya que en exteriores funciona correctamente además de disponer de un rango más amplio si se diera el caso de tener que cambiar la localización del mismo.

3.1.4. Sensor infrarrojo

El sensor elegido para la medición del tráfico de personas fue el *Panasonic AMG8853*¹². Es un sensor de detección con una matriz térmica 8x8 que funciona tanto a 3.3V como a 5V y comunicación I2C. Ha sido diseñado específicamente para detectar la temperatura de una superficie y el movimiento de personas u objetos.

Las características del sensor son las siguientes: Consta de 64 sensores térmicos individuales que pueden formar una imagen de acuerdo con el calor que detectan con un rango de medición entre -20°C y +80°C. Además la distancia de detección es de 5 metros con un ángulo de visión de 60 grados. Por último, al estar formado por una matriz, nos permitía



Figura 3.6: Sensor infrarrojo: Sharp GP2Y0A710K0F

conocer la dirección del desplazamiento de las personas con lo que simplificábamos el despliegue del componente y abaratabamos costes con el consiguiente aumento de escalabilidad. Todas estas cualidades hacían del *Panasonic AMG8853* un sensor muy cercano a nuestras necesidades.



Figura 3.7: Sensor infrarrojo: Panasonic AMG8853

3.2. Desarrollo técnico

3.2.1. Desarrollo

Una vez estudiados que componentes eran los más adecuados para la medición de los dos tipos de tráfico, llegó el momento de comenzar con la construcción de los sensores.

Dispositivo para el tráfico rodado

Comenzamos con el dispositivo para el tráfico rodado, dado que había sido iniciado por otro alumno (Adrián Asensio). La construcción del dispositivo no fue excesivamente compleja ya que solo necesitaríamos la tarjeta CC1310, el ultrasonido, el sensor inductivo y la fuente de alimentación. En primer lugar, el sensor de ultrasonidos LV-MaxSonar-EZ1 [3.1.3](#) conectaría tres de sus pines a la tarjeta CC1310. Concretamente, los pines de voltaje y GND se conectarían respectivamente a 5V y a GND y el pin analógico (indicado en el sensor como AN) debería estar conectado al pin DIO23 del *launchpad*. A continuación, el sensor inductivo conectaría uno de sus pines a GND de la tarjeta y el otro al DIO24. Por último, deberemos conectar la *powerbank* a la tarjeta para proporcionar alimentación. En la figura [3.12](#) podemos ver como son las conexiones de forma gráfica.

Una vez construida la circuitería se añadió una capucha impresa en 3D para controlar y dirigir el haz de ultrasonidos en la dirección adecuada y concentrarlo enfrente del dispositivo. En la figura [3.9](#) podemos ver el modelo 3D.

El siguiente paso fue el de contener todo dentro de una caja que hiciera funciones de protección frente a los elementos atmosféricos. El dispositivo entero lo podemos ver en las figuras [3.10](#) y [3.11](#) tanto abierto como cerrado.

Este dispositivo tuvo que realizarse multiplicado por cinco debido a que tenía que medir las principales salidas y entradas de Ciudad Universitaria. Los dispositivos deberían estar colocados a lo largo de la Avenida Complutense donde esta se comunica con carreteras de entrada/salida a Ciudad Universitaria.

Una vez realizadas las pruebas en interior para comprobar que el dispositivo funcionaba correctamente llegó el momento de desarrollar el software necesario para interpretar si estamos midiendo un vehículo particular o uno público. En primera instancia, estos dispositivos estaban pensados para ser colocados en lo alto de farolas o semáforos apuntando hacia el suelo. Así pues una vez colocado, se activaría el sensor inductivo pasando un imán cerca de la carcasa protectora y como consecuencia comenzaría la lectura, recogida y envío de datos

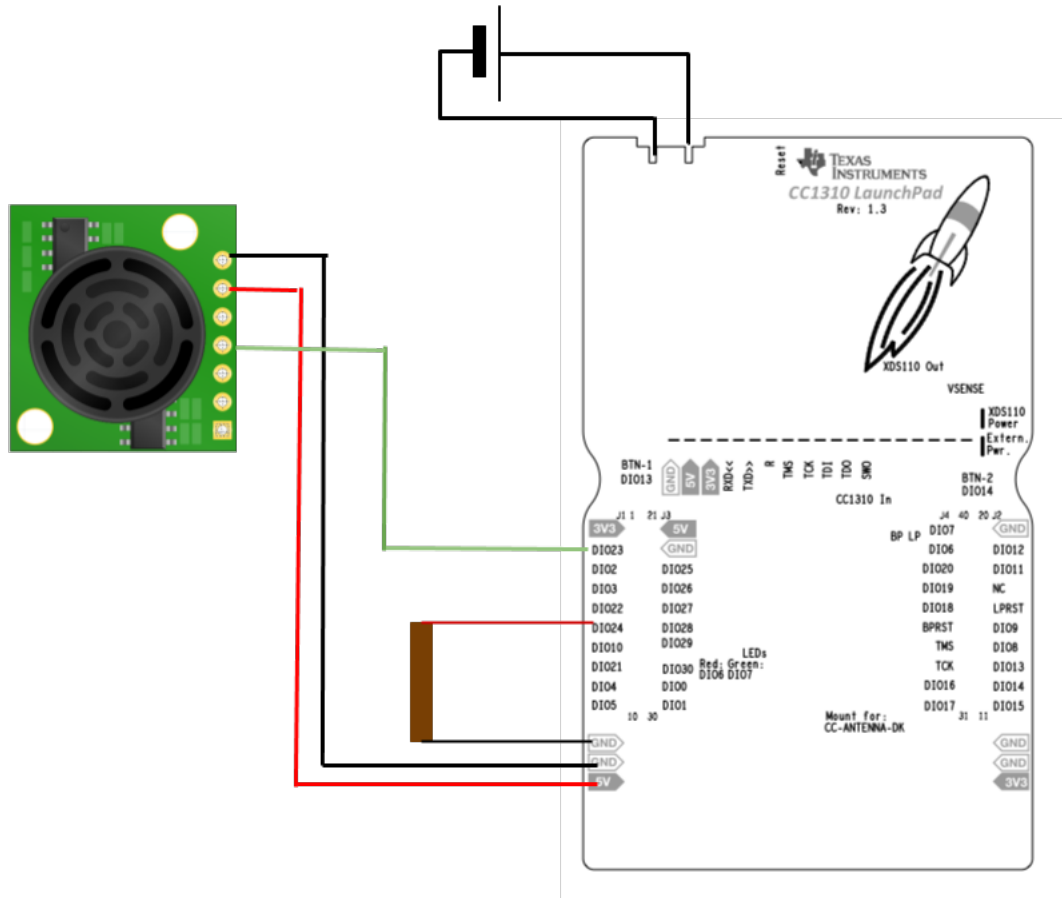


Figura 3.8: Dispositivo ultrasonido para tráfico rodado

a *Sigfox*. El algoritmo tiene una primera tarea de calibración donde no debe pasar ningún vehículo para tener la mayor precisión posible. Este proceso se hace tomando unos 20 valores y hallando la media de la distancia al suelo. La calibración tiene en cuenta los posibles errores en las mediciones y actúa en consecuencia variando la media. A continuación se establecen, a partir de la distancia al suelo, los valores máximo y mínimo tanto para coches como para autobuses. Al pasar uno de estos dos tipos de vehículos el sensor toma una serie de medidas consecutivamente para reconocer el tipo de vehículos y saber si ha terminado de atravesar el haz de ultrasonidos o si por el contrario sigue siendo el mismo vehículo.

Sin embargo, por situaciones ajenas al trabajo de fin de máster no se pudieron instalar en

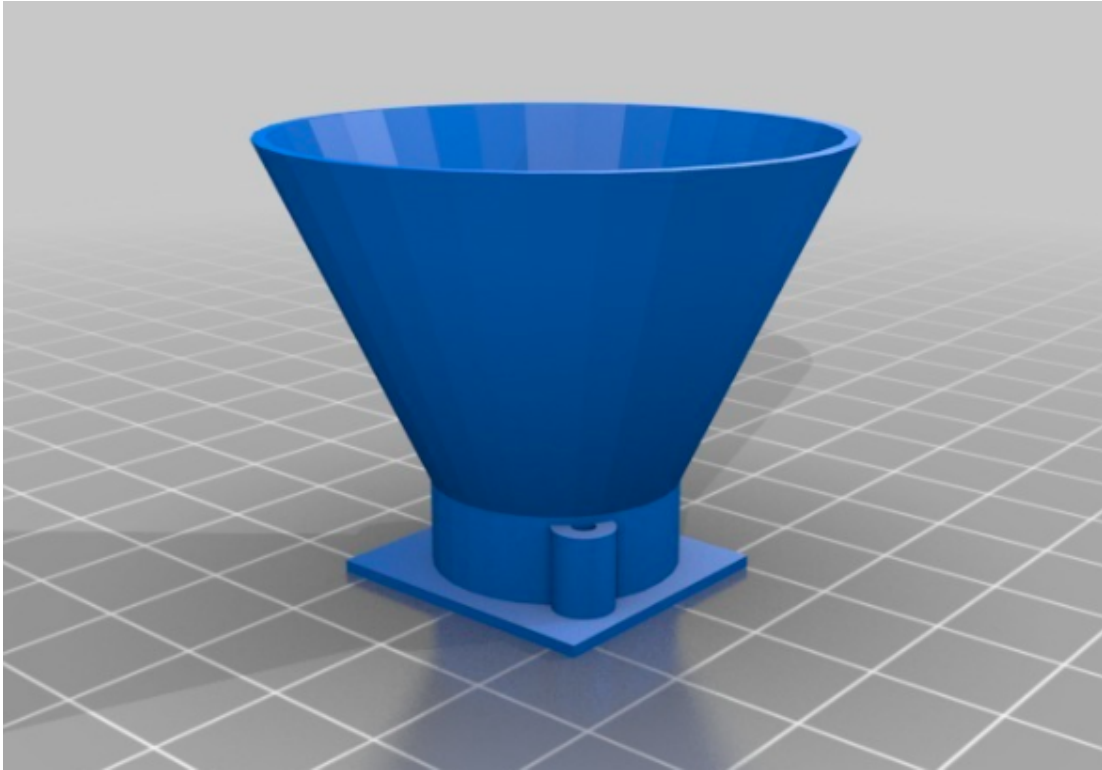


Figura 3.9: Capucha impresa en 3D

farolas ni semáforos a tiempo. Por lo que se optó por variar el algoritmo para que en lugar de tomar medidas verticales las tomara horizontales. Como plan de contingencia, el dispositivo se colocó a un metro de altura del suelo y se contaban el número de medidas por unidad de tiempo para conocer si era un vehículo particular, donde las medidas serían menores al ser más corto y un vehículo público donde las medidas serían mayores al tener más longitud que un coche. Al tener que modificar el algoritmo perdimos un poco de precisión debido a que se tuvo que establecer una media de velocidad y si, por ejemplo, un particular circulaba con lentitud podía ser reconocido como un autobús. Sin embargo, durante las pruebas de campo se observó que la cantidad de errores eran muy pocos en comparación con los aciertos así que se optó por seguir por este camino.

Las mediciones tomadas por los dispositivos se enviaban cada 15 minutos tal y como se indica en la tarea *sigfoxTask*. Son recogidas por *Sigfox* y automáticamente enviadas mediante un *callback* a nuestro servidor:

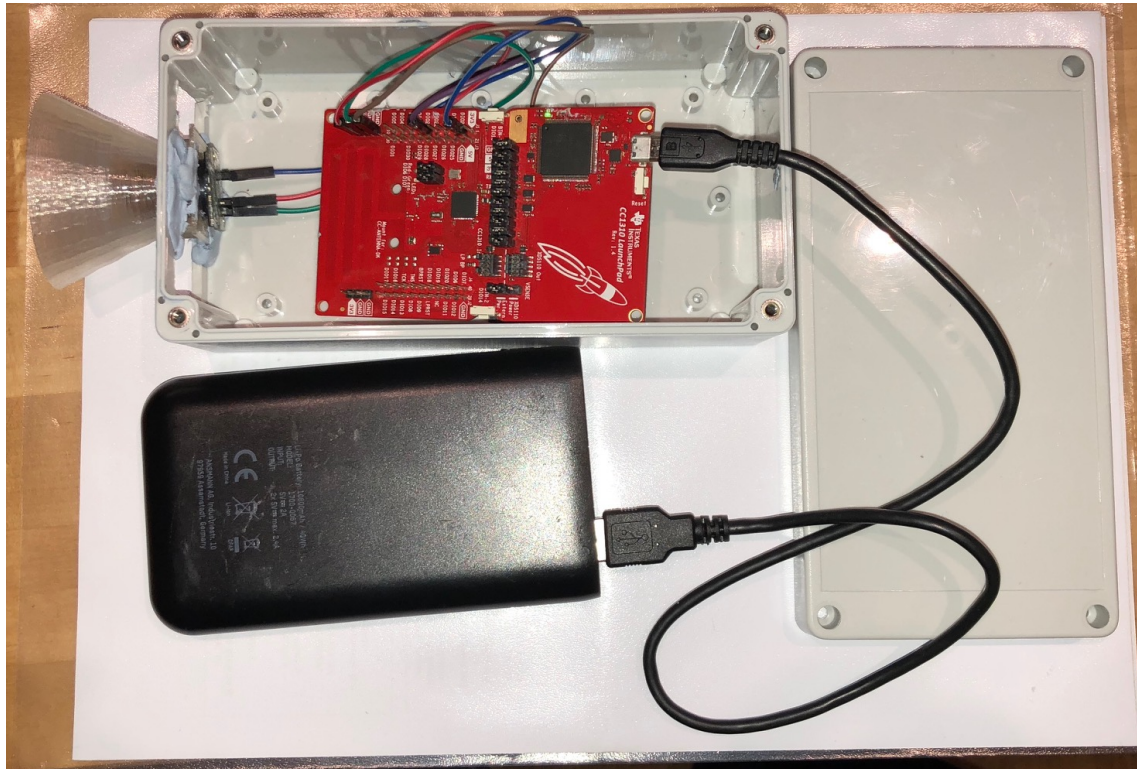


Figura 3.10: Dispositivo abierto

```
http://xxx.xx.xx.xx/cargarJson.php?id={device}&data={data}&time={time}&lng={lng}&lat={lat}&station={station}&avgSnr={avgSnr}
```

que utilizando *scripts PHP* recogerán la información, la parsearán y la incluirán en una base de datos *MySQL* para su posterior procesamiento. Durante este proceso, los datos se enriquecen con información meteorológica proveniente de dos fuentes: Aemet (Agencia estatal de meteorología)² donde se ha utilizado su herramienta de OpenData para conseguir los datos de temperatura, lluvia y viento. La otra fuente ha sido el portal de datos de la comunidad ³ de donde recuperamos la contaminación del monóxido de carbono. Una vez los datos están almacenados y parseados, se pudo pasar a la siguiente fase que fue la de probar algoritmos de *machine learning* para comprobar cuál era el que más confianza nos ofrecía.

²url<http://www.aemet.es/>

³url<http://datos.madrid.es/>



Figura 3.11: Dispositivo cerrado

Se probaron nueve algoritmos descritos brevemente en la sección 3.3.1.

Dispositivo para el tráfico de personas

El dispositivo para medir el tránsito de personas fue el siguiente en construirse. En este caso, se utilizó la tarjeta CC1350 y al igual que el primer dispositivo su construcción no resultó compleja. Utilizamos, como se ha dicho la tarjeta CC1350 para establecer la comunicación con *Sigfox*, el sensor infrarrojo AMG8853 descrito en la sección 3.1.4, de nuevo un sensor inductivo, la fuente de alimentación y el cableado necesario para realizar el circuito. Siguiendo las indicaciones del *datasheet*, se unió el pin SCL (Serial CLOCK) del sensor AMG8853 al pin DIO4 del launchpad y el pin SDA (Serial DATA) al pin DIO5 que son los pines para transmitir y recibir mediante I2C (utilizado para la comunicación entre diferentes partes de un circuito, en este caso, el sensor y el launchpad). Al igual que en el dispositivo de tráfico rodado se instaló un sensor inductivo para activar el dispositivo, y por último, a diferencia del sensor de ultrasonidos, este funciona a 3.3 voltios con lo que se adaptó el circuito para cumplir con estas características y, al igual que el primer dispositivo, este sería protegido por una caja del mismo tipo. En la figura 3.12 se puede ver el esquema

del montaje.

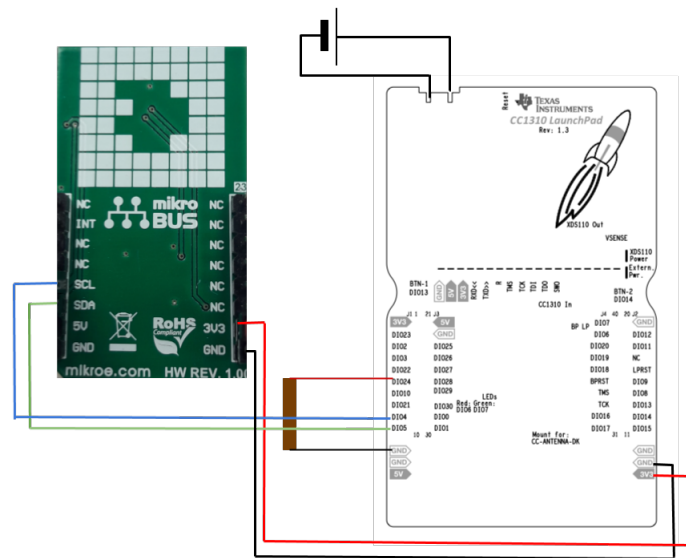


Figura 3.12: Dispositivo ultrasonido para tráfico rodado

A continuación, se tuvo que decidir donde colocar el dispositivo para que las mediciones fueran lo más correctas posibles y, a su vez, no sufriera vandalismo, ya que debido a la naturaleza del sensor, este debía estar colocado a una altura accesible (las mediciones serían los cuerpos de las personas). Se decidió colocarlo sobre la entrada al Metro de Ciudad Universitaria apuntando a las escaleras. Este lugar parecía el más indicado porque el ángulo y la distancia a las personas permitía recoger información de toda la entrada y a su vez, el dispositivo, parecía formar parte de la estructura. En la imagen 3.13 se puede ver el lugar donde se pretendía instalar.

3.3. Recopilación de datos

Durante la realización del proyecto, una de las tareas más complejas fue la recopilación de datos. Al contar solo con un cuatrimestre para su realización, esta tarea era crítica así que se tuvo que contar con un plan de contingencia si llegado el momento no se habían conseguido datos suficientes para entrenar los algoritmos.



Figura 3.13: Colocación del dispositivo para personas

Primero se optó por desarrollar los dispositivos para así instalarlos en semáforos o farolas y que ellos, automáticamente, surtieran de datos al resto de procesos (parseo de información, scripts php, base de datos, etc.). Sin embargo, no pudimos colocarlos debido a problemas, tanto burocráticos como de tiempo, con el Ayuntamiento de Madrid y la Universidad Complutense de Madrid. Al no poder instalarlos y por ende, tener que modificar el algoritmo, se cambió el enfoque para que la toma de datos fuera horizontal (desde el suelo) en lugar de vertical. Este cambio añadió un "daño colateral" que significaba que siempre debía estar presente una persona por la seguridad del dispositivo. En este punto se acudió durante una semana, durante diferentes días y horas, para tomar medidas del tráfico rodado mientras se vigilaba el dispositivo por lo que la cantidad de datos es solo perteneciente a una semana, aunque como se comprobó durante estas pruebas de campo, el dispositivo y el sistema funcionaba correctamente (recopilación, envío, procesamiento y persistencia). Además, y como nuevo plan de contingencia, se pidió al Departamento de Tecnologías del Tráfico los datos del tránsito de sus mediciones en los puntos que nos interesaban, pero lamentablemente no llegaron a tiempo para poder incluirlos en el entrenamiento.

En el caso del dispositivo para personas, se pidió a Metro Madrid que proporcionara la

información de sus tornos. Este plan de contingencia se estableció porque, al ser el segundo dispositivo desarrollado, el tiempo era demasiado ajustado para conseguir los permisos de instalación, hacer las pruebas, recopilar datos suficientes, etc. Por lo que llegado el momento se optó por utilizar este plan y poder tener tener algunos datos para pruebas. Metro Madrid nos ofreció los datos de Octubre 2017 y de Noviembre 2017 para trabajar con ellos.

Ante estas vicisitudes, tuvimos que adaptarnos y aunque los datos no han sido suficientes como para entrenar los modelos de la forma más adecuada, se ha intentado probar con todos los algoritmos descritos en la sección 3.3.1, realizando con éxito la prueba de concepto.

3.3.1. Machine learning

Una vez que obtenido un importante volumen de datos, el siguiente paso era utilizar un algoritmo de *machine learning* para entrenar nuestro sistema. El tipo de algoritmos elegidos para realizar pruebas fueron los algoritmos de clasificación supervisados y enumerados en la sección 3.3.2. Un algoritmo de clasificación supervisado nos permite hacer predicciones basadas en un conjunto de mediciones, sobre diferentes características y clases, que en nuestro caso son ofrecidas por los sensores. Es decir, las clases en nuestro caso, serían la afluencia del número de vehículos o personas. Esta clasificación la podemos desglosar de una forma binomial y clasificar cada ejemplo en una afluencia de pocas/os o muchas/os personas/tráfico. Nuestras tablas de entrenamiento para los diferentes algoritmos tenían el siguiente aspecto: Por un lado:

- **id**: Clave única. Sin valor real aplicable.
- **device**: Identificador de la tarjeta CC13X0 que envía los datos.
- **data**: Dato enviado por la tarjeta CC13X0 sin procesar.
- **coches**: Número de coches de esa medición. Extraído del campo data.
- **autobuses**: Número de coches de esa medición. Extraído del campo data.

- **longitud:** Longitud de la tarjeta CC13X0.
- **latitud:** Latitud de la tarjeta CC13X0.
- **fecha:** Fecha de la medición.
- **hora:** Hora de la medición.
- **afluencia:** La clase binomial para entrenar el algoritmo. Será poca o mucha y se calcula a partir de la media de todas las mediciones tomadas. En virtud de los datos recogidos, si la suma de coches y autobuses sobrepasa la cantidad de 120 entonces esta clase será mucha, en caso contrario, será poca.

id	device	data	coches	autobuses	longitud	latitud	fecha	hora	afluencia
41	20C70F	c0127a10	127	10	-4	40	2017-11-20	14:30:00	mucha

Cuadro 3.1: Tabla Datos donde se guardan los datos del tráfico

Por otro lado:

- **id:** Clave única. Sin valor real aplicable.
- **día:** Día de toma de la medida.
- **mes:** Mes de toma de la medida.
- **año:** Año de toma de la medida.
- **hora:** Hora de toma de la medida.
- **temperatura:** Temperatura a la hora de la medición.
- **lluvia:** Lluvia a la hora de la medición.
- **viento:** Viento a la hora de la medición.

- **contaminacion:** Contaminación a la hora de la medición.
- **afluencia:** La clase binomial para entrenar el algoritmo. Será poca o mucha y se calcula a partir de la media de todas las mediciones tomadas. Si la suma de coches y autobuses sobrepasa la cantidad de 120 entonces esta clase será mucha, en caso contrario, será poca.

id	día	mes	año	hora	temperatura	lluvia	viento	contaminacion	afluencia
10	28	11	2017	19	6.3	0.2	0	0.3	mucha

Cuadro 3.2: Tabla Meteo usada para entrenamiento de machine learning

Para establecer si la afluencia de las personas era mucha o poca se halló la media diaria de las personas que salen del metro de Ciudad Universitaria durante en un mes completo, teniendo en cuenta solo aquellos días lectivos. Se calculó que la media de afluencia era 28476 personas diarias que se redondeó a 28000 para establecer un umbral donde si la afluencia superaba esa línea, se consideraba una alta afluencia, es decir, mucha. Si por el contrario, el número se hallaba por debajo la afluencia se determinaba como poca.

- **id:** Clave única. Sin valor real aplicable.
- **día:** Día de toma de la medida.
- **mes:** Mes de toma de la medida.
- **año:** Año de toma de la medida.
- **temperatura:** Temperatura a la hora de la medición.
- **lluvia:** Lluvia a la hora de la medición.
- **viento:** Viento a la hora de la medición.
- **contaminacion:** Contaminación a la hora de la medición.

- **afluencia:** La clase binomial para entrenar el algoritmo. Será poca o mucha y se calcula a partir de la media de todas las mediciones tomadas. Si la media sobrepasa la cantidad de 28000 personas diarias entonces esta clase será mucha, en caso contrario, será poca.

id	dia	mes	anyo	temperatura	lluvia	viento	contaminacion	afluencia
10	28	11	2017	6.3	0.2	0	0.3	mucha

Cuadro 3.3: Tabla personas usada para entrenamiento de machine learning

Una vez tenemos tanto la tabla 3.2 como la 3.3 estas serán usadas para entrenar los algoritmos.

Las variables de decisión pueden ser:

- La fecha (día, mes, año), variables ambientales y contaminación.
- Variables ambientales y de contaminación.
- Día lectivo de la semana, variables ambientales y contaminación.
- Etc.

En nuestro caso hemos optado por las variables ambientales y de contaminación en el caso del tráfico rodado porque debido a la poca cantidad de datos disponibles, la fecha no variaba lo suficiente y no se podía utilizar como variable de decisión. En el caso del tráfico de personas se ha utilizado, además de las variables ambientales y de contaminación, el día lectivo.

3.3.2. Algoritmos

Los siguientes algoritmos de clasificación fueron aplicados al tráfico de vehículos y se aplicó sobre ellos la matriz de confusión para comprobar evaluar cual de ellos era más apropiado para nuestros propósitos.

A continuación se describen los algoritmos usados para realizar la clasificación, tanto para tráfico rodado como para tránsito de personas. En el caso de los vehículos se han estudiado todos y en el caso de las personas se han estudiado Naive Bayes, árboles de decisión, máquinas de vector soporte y análisis discriminante.

Naives bayes

Es un algoritmo clasificador que utiliza el teorema de Bayes. El algoritmo acepta que la presencia o ausencia de alguna de las características no afecta a que estén o no presentes otras características. Una ventaja del clasificador de Naives Bayes es que se puede entrenar con un pequeño volumen de datos para poder hacer predicciones.

Sea $\{A_1, A_2, \dots, A_i, \dots, A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero (0). Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$. Entonces, la probabilidad $P(A_i|B)$ viene dada por la expresión:

$$P(A_i | B) = \frac{P(B | A_i) P(A_i)}{P(B)}$$

K-nearest Neighbours

Es un método de clasificación supervisada que sirve para estimar la función de densidad $F(x/C_j)$ de las características x en cada clase C_j . Es un método que clasifica los ejemplos en relación a la cercanía de las características que tienen entre ellos. Para ello almacena los vectores de las características y las clases y calcula la distancia euclidiana entre los vectores que se han almacenado y el nuevo vector y se seleccionan los k ejemplos más cercanos. El nuevo vector se clasifica con la clase que más se repite en los vectores seleccionados.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ri} - x_{rj})^2}$$

Regresión logística

Este algoritmo de *machine learning* es utilizado para predecir el resultado de una categoría con un número limitado de estados, en nuestro caso la afluencia (mucho o poco), en función de otras variables. Es un algoritmo que utiliza teoría probabilística y se basa en ella para predecir, dependiendo de unas variables, la clase resultante.

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Hay que tener en cuenta que $F(x)$ es la probabilidad de que la variable independiente (afluencia) se iguale a mucho o poco, β_0 es el punto donde se corta con los ejes de la ecuación de la regresión lineal (cuando algún valor del criterio es igual a 0) y β_1 es el coeficiente de regresión multiplicado por algún otro valor.

Máquinas de vector soporte

Este método se basa en encontrar el hiperplano que mejor divida un conjunto de datos divididos en dos clases. Consta de unos vectores de apoyo que son los puntos más cercanos al hiperplano y que si los elimináramos modificarían a este. Por lo tanto podemos pensar en este hiperplano como una barrera que separe los datos que hemos introducido en las dos clases. Cuanto más lejos estén los datos del hiperplano, más seguros podemos estar de que ese dato pertenece a la clase en concreto y, por supuesto, en los datos de entrenamiento el hiperplano tiene que delimitar correctamente los datos en cada clase. Hay que tener en cuenta que los datos pueden proceder de más de dos variables y estar "mezclado", con lo que el hiperplano ya no nos serviría. En este caso se debe aplicar las funciones de Kernel para aumentar la dimensionalidad.

Red neuronal

Dentro de las redes neuronales podemos encontrar varios tipos de algoritmos. Para entrenar una red neuronal, debe especificar la arquitectura de la misma, así como las opciones

del algoritmo de entrenamiento. Seleccionar y ajustar estos parámetros puede ser difícil y costoso tiempo. La optimización Bayesiana es un algoritmo muy adecuado para optimizar los parámetros internos de los modelos de clasificación.

Análisis discriminante

Es una técnica supervisada de extracción de características utilizada para encontrar una combinación lineal de varias funciones que separan las clases. Su objetivo principal es el de reducir el tamaño de los datos y con ello disminuir el tiempo de computación. Una vez aplicada esta técnica a los datos, la dispersión de los datos de la misma clase disminuye y se maximiza entre clases diferentes. Esta técnica supone que las funciones de densidad de las probabilidades condicionales $p(\vec{x}|y = 0)$ y $p(\vec{x}|y = 1)$ siguen una distribución normal con parámetro de media y covarianza $(\vec{\mu}_0, \Sigma_0)$, $(\vec{\mu}_1, \Sigma_1)$. Bajo esta suposición, la solución óptima de Bayes es predecir puntos como en la segunda clase si el registro de las proporciones de probabilidad está debajo de algún umbral T , de tal modo que:

$$(\vec{x} - \vec{\mu}_0)^T \sum_0^{-1} (\vec{x} - \vec{\mu}_0) + \ln|\Sigma_0| - (\vec{x} - \vec{\mu}_1)^T \sum_1^{-1} (\vec{x} - \vec{\mu}_1) + \ln|\Sigma_1| > T$$

El análisis discriminante lineal es muy utilizado en predicciones de bancarrota, reconocimiento de caras, marketing, estudios biométricos, ciencias de la tierra, etc.

Árboles de decisión

En los árboles de decisión enfocados al *machine learning* estos son usados después de un entrenamiento para clasificar un nuevo ejemplo. Este se envía por las diferentes ramas del árbol y, dependiendo de sus características, toma una decisión (en cada nodo) u otra hasta que alcanza un nodo "hoja" que determinará la clase de nuestro ejemplo predictivo.

Árboles de decisión empaquetados

Es un metaalgoritmo de *machine learning* que permite aumentar y estabilizar otros algoritmos, en nuestro caso árboles de decisión. Esta técnica produce un nuevo conjunto de datos donde se reduce la varianza y evita el sobreajuste.

Árboles de decisión potenciados

Al igual que el anterior, es un método para crear nuevos conjuntos de datos donde estos sufren una corrección para poder predecir el ejemplo que le pasemos con más probabilidad. Esta técnica se puede usar varias veces sobre los árboles sucesivos del primer conjunto de datos para ir afinando nuestra probabilidad.

3.3.3. Validación

Validación cruzada

La validación cruzada o cross-validation es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica. En este caso, esta técnica de entrenamiento y validación se utilizó solamente para el tránsito de personas y concretamente con los algoritmos de clasificación de Naive Bayes, árboles de decisión, máquinas de vector soporte y análisis discriminante, descritos más arriba.

3.4. Planificación del proyecto

En esta sección se muestra la planificación que se ha llevado durante la realización del proyecto. La duración del proyecto ha sido de un cuatrimestre. Se comenzó el 2 de Octubre de 2017 y se termino el 31 de Enero de 2018. Se puede ver en la figura [3.14](#).

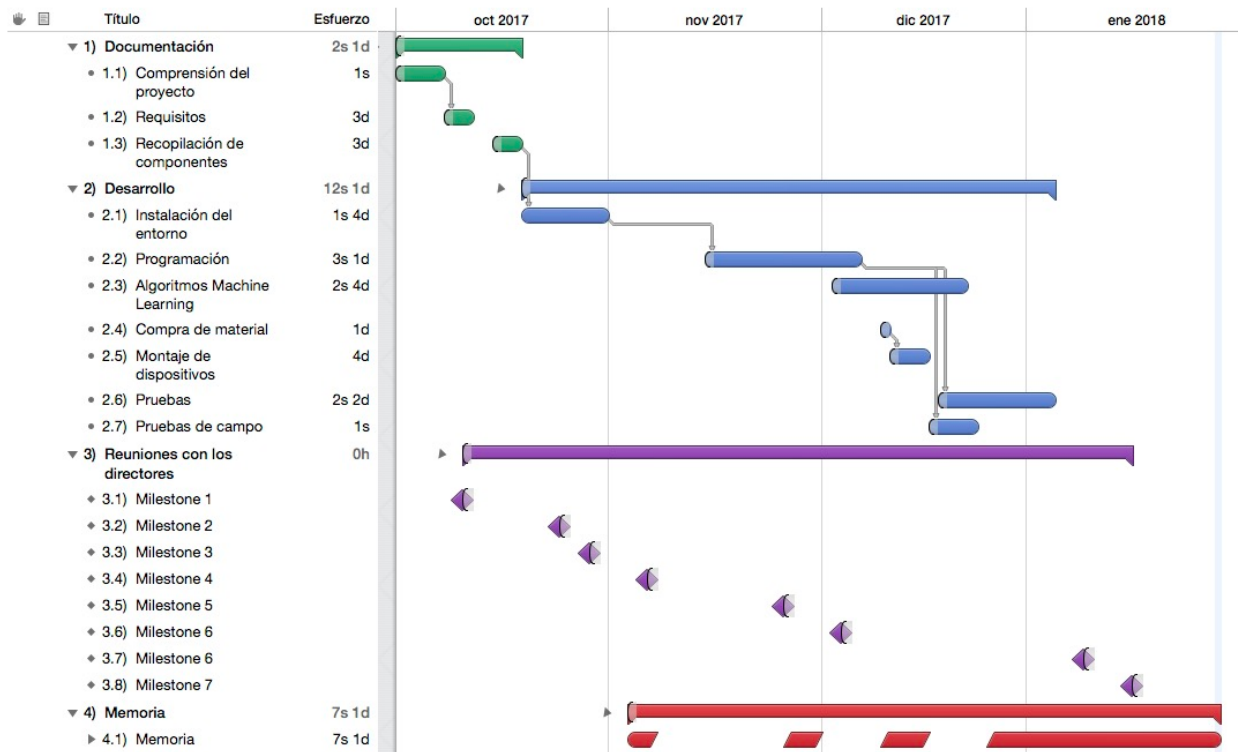


Figura 3.14: Diagrama Gantt: Planificación del proyecto

Capítulo 4

Resultados

Podemos saber poco del futuro, pero lo suficiente para darnos cuenta de que hay mucho que hacer.

Alan Turing

En esta sección se mostrarán los resultados obtenidos al utilizar los datos provenientes tanto de la recolección de los dispositivos como los que hemos obtenido como resultado de aplicar los planes de contingencia (ofrecidos por Metro Madrid). Como se demuestra a continuación el algoritmo que ha resultado más beneficioso para entrenar un modelo de predicción, aún con los pocos datos que teníamos, es el árbol de decisión para el tráfico rodado y para el tránsito de personas el de validación cruzada.

4.1. Tráfico rodado

En esta primera sección se muestran los resultados de los algoritmos de *machine learning* que se han utilizado para entrenar los modelos para el tráfico rodado.

Para ver establecer que algoritmo nos es más adecuado se han utilizado matrices de confusión. Estas son herramientas que permiten la visualización del desempeño de un algoritmo empleado en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real.

Los datos en verde indican las predicciones correctas y los rojos las incorrectas.

4.1.1. Naive Bayes

Los resultados para el algoritmo de Naive Bayes

	Matriz de confusión: Naive Bayes		
	Mucha	Poca	Total
Mucha	55	23	70,5 % 29,1 %
Poca	40	32	44,4 % 55,5 %
Total	57,8 % 42,2 %	58,2 % 41,8 %	58 % 42 %

Cuadro 4.1: Resultados Naive Bayes

4.1.2. Support vector machines

Los resultados para el algoritmo de support vector machines:

	Matriz de confusión: SVN		
	Mucha	Poca	Total
Mucha	55	23	70,5 % 29,1 %
Poca	42	30	41,6 % 58,3 %
Total	56,7 % 43,3 %	56,6 % 43,3 %	56,7 % 43,3 %

Cuadro 4.2: Resultados SVN

4.1.3. Redes Neuronales

Los resultados para el algoritmo que utiliza redes neuronales:

	Matriz de confusión: Redes neuronales		
	Mucha	Poca	Total
Mucha	32	32	50 % 50 %
Poca	38	47	54,5 % 45,5 %
Total	54,5 % 45,5 %	50 % 50 %	52,2 % 47,8 %

Cuadro 4.3: Resultados utilizando redes neuronales

4.1.4. Análisis discriminante

Los resultados para el algoritmo de análisis discriminante:

	Matriz de confusión: Análisis discriminante		
	Mucha	Poca	Total
Mucha	56	22	71,8 % 28,2 %
Poca	45	27	45,2 % 54,8 %
Total	55,4 % 44,6 %	55,1 % 44,9 %	55,3 % 44,7 %

Cuadro 4.4: Resultados utilizando Análisis discriminante

4.1.5. k vecinos más cercanos

Los resultados para el algoritmo de análisis discriminante:

	Matriz de confusión: k vecinos más próximos		
	Mucha	Poca	Total
Mucha	58	20	65,9 % 34,1 %
Poca	15	77	79,1 % 21,9 %
Total	79,5 % 20,5 %	74 % 26 %	76,7 % 23,3 %

Cuadro 4.5: Resultados utilizando k vecinos más próximos

4.1.6. Árbol de decisión

Los resultados para el algoritmo de Árbol de decisión:

	Matriz de confusión: Árbol de decisión		
	Mucha	Poca	Total
Mucha	66	12	84,6 % 15,4 %
Poca	6	66	91,6 % 8,4 %
Total	91,6 % 8,4 %	84,6 % 15,4 %	88,1 % 11,9 %

Cuadro 4.6: Resultados utilizando árbol de decisión

4.1.7. Árbol de decisión *Boosted*

Los resultados para el algoritmo de Árbol de decisión *Boosted*:

	Matriz de confusión: Árbol de decisión <i>Boosted</i>		
	Mucha	Poca	Total
Mucha	58	20	74,3 % 25,7 %
Poca	18	74	75 % 25 %
Total	76,3 % 23,7 %	72,9 % 27,1 %	74,6 % 25,4 %

Cuadro 4.7: Resultados utilizando árbol de decisión *Boosted*

4.1.8. Árbol de decisión *Bagged*

Los resultados para el algoritmo de Árbol de decisión *Bagged*:

	Matriz de confusión: Árbol de decisión <i>Bagged</i>		
	Mucha	Poca	Total
Mucha	41	37	52,5 % 47,5 %
Poca	18	74	48,6 % 51,4 %
Total	52,5 % 47,5 %	48,6 % 51,4 %	50,6 % 49,4 %

Cuadro 4.8: Resultados utilizando árbol de decisión *Bagged*

4.2. Tránsito de personas

En esta segunda sección se muestran los resultados de los algoritmos de que se ha utilizado para entrenar el modelo para el tránsito de personas. En este caso se ha utilizado la validación cruzada o cross-validation con Naives bayes, árboles de decisión, máquinas de vector soporte y análisis discriminante. Se ha utilizado validación cruzada que es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica.

4.2.1. Validación cruzada

Los resultados para los algoritmos utilizando validación cruzada con un K=5 (días lectivos de la semana):

Validación cruzada	
Algoritmo	Resultado
Naives Bayes	0,3538
Support Vector Machines	0,1846
Análisis discriminante	0,1538
Árbol de decisión	0,1846

Cuadro 4.9: Resultados de los algoritmos aplicando validación cruzada.

4.3. Resultados

Como se ha comprobado después del estudio de varios modelos de entrenamiento el que más se adecua a nuestras necesidades son el árbol de decisión y el análisis discriminante con validación cruzada. En el primero, que corresponde al tráfico rodado, se tenían muchos más datos y por esa razón los resultados son más concluyentes. Se ve, después de aplicar la matriz de confusión, que utilizando árboles de decisión conseguimos un acierto en la predicción de 88,1 % frente al 11,9 % de error. Por lo tanto creemos que este es el algoritmo a aplicar para el modelo de predicción referente al tráfico de vehículos. En el segundo, correspondiente al tránsito de personas, en el que hemos utilizado la validación cruzada sobre diferentes algoritmos con un $K=5$ que hace referencia a los días lectivos de cada semana. Al tener solo los datos referentes a Octubre y Noviembre, el entrenamiento no ha sido muy profundo y por lo tanto los los resultados finales del algoritmo no son del todo concluyentes pero, aún así, el análisis discriminante se evidencia como un candidato firme a ser utilizado.

Capítulo 5

Conclusiones y trabajo futuro

Per aspera ad astra.

Séneca, el joven.

5.1. Conclusiones

La tecnología de *fog computing* es relativamente nueva pero después de haber realizado un estudio sobre qué herramientas o dispositivos eran los más adecuados para este proyecto comprobamos que era viable. Los desafíos a los que nos enfrentábamos eran el ofrecer una escalabilidad viable, una latencia adecuada así como un coste y una eficiencia energética que posibilite alcanzar éxito final con vistas a aplicarlo en entornos reales.

Durante el trabajo, se estableció una red de dispositivos formados por sensores de infrarrojos y ultrasonidos, que aunque no se pudieron colocar en Ciudad Universitaria, si que se comprobó que su funcionamiento era correcto mediante las pruebas de campo. Se constató, también, como la comunicación entre estos dispositivos y la red Sigfox era correcta así como la comunicación con el nodo que condensaba esta información, en nuestro caso un servidor cedido por la facultad que se encargó del procesamiento.

Después de haber acreditado el buen funcionamiento de los dispositivos y algoritmos desarrollados (incluso con los cambios), el procesamiento de datos y probado diferentes modelos de *machine learning* en busca del más adecuado para establecer un modelo predictivo,

vemos que esta tecnología puede ser un buen camino a tomar dentro de *IoT*.

Aunque durante el proyecto nos hemos encontrado con dificultades que escapaban a nuestro control, véase que aunque no se han podido instalar los dispositivos adecuadamente en su entorno definitivo y en consecuencia no se ha obtenido un volumen de datos adecuado, si que se ha demostrado la viabilidad del proyecto. Se ha probado que aplicar *fog computing* en una *smart city*, como es la Ciudad Universitaria, puede ser viable desde el punto de vista económico, escalable, ya que solo habría que replicar cada dispositivo y añadirlo a nuestra red Sigfox y tolerante a fallos, puesto que al utilizar grandes volúmenes de datos la gran cantidad de datos correctos opaca a los incorrectos. Sin embargo, una de las facetas de *fog computing* que no hemos podido comprobar es la disminución de la latencia respecto a tener que subir los datos a la nube. Al no tener un gran tráfico de datos, no se ha podido estresar el servidor y tomar medidas de latencia y esto es algo que en un futuro se debería validar.

En definitiva, este proyecto demuestra que *fog computing* es un camino perfectamente posible para establecer modelos de predicción en *smart cities*, siendo escalable, con un coste reducido tanto económicamente como energéticamente y con el fin de gestionar mejor tanto el tráfico rodado como el tránsito de personas.

5.2. Conclusions

The technology of fog computing is relatively new but after having carried out a study on which tools or devices were the most suitable for this project we verified that it was viable. The challenges we faced were to offer viable scalability, adequate latency as well as a cost and energy efficiency that enables us to achieve final success in order to apply it in real environments.

During the work, a network of devices formed by infrared and ultrasound sensors was established, although they could not be placed in Ciudad Universitaria, if it was verified that their operation was correct through field tests. It was also found that the communication between these devices and the Sigfox network was correct, as well as the communication

with the node that condensed this information, in our case a server assigned by the faculty that was responsible for the processing.

After having accredited the good functioning of the devices and algorithms developed (even with the changes), the data processing and tested different models of machine learning in search of the most suitable to establish a predictive model, we see that this technology can be a good path to take within IoT.

Although during the project we have encountered difficulties that were beyond our control, see how not being able to install the devices in their final environment or the lack of access to an adequate volume of data has been corroborated the viability of the project. It has been proven to apply fog computing in a smart city, as is the Ciudad Universitaria, it can be viable from the economic point of view, scalable, since you only have to replicate each device and add it to our network Sigfox and tolerant to failures, since when using large volumes of data the large amount of correct data dulls the wrong. However, one of the facets of fog computing that we have not been able to verify is the decrease in latency with respect to having to upload the data to the cloud. By not having a large data traffic, it has not been possible to stress the server and take measures of latency and this is something that in the future should be validated.

In short, this project shows that fog computing is a perfectly possible way to establish prediction models in smart cities, being scalable, with a reduced cost both economically and energetically and in order to better manage both the traffic and the traffic of people.

5.3. Trabajo futuro

Las posibilidades de la tecnología de *fog computing* son inmensas. Siguiendo la línea de este trabajo, enfocado a las *smart cities*, a continuación se describen algunas posibilidades que su continuación nos ofrece.

En primer lugar, una de las posibilidades de continuación es la de añadir más información del entorno. Es decir, añadir a las mediciones de lluvia, temperatura, viento y medición

del monóxido de carbono otras muchas métricas. Por ejemplo, se podría completar la información atmosférica midiendo también el metano, dióxido de azufre, monóxido de nitrógeno, dióxido de nitrógeno, ozono, etc. que tienen un impacto considerable en la contaminación. Por otro lado, y tratándose de Ciudad Universitaria situada dentro de la ciudad de Madrid, esta está obligada a cumplir la normativa de tráfico de vehículos. Esta situación puede influir en forma de avisos y circunstancias excepcionales como es el caso de la limitación de circular a los vehículos privados con matrícula par/impar, que los mismos no puedan circular por el centro de Madrid o la limitación de velocidad en vía urbana. Así pues esta información también podría ser valiosa a la hora de realizar un modelo veraz.

Otra de las posibilidades es añadir a las tarjetas CC1310/CC1350 nuevos sensores que nos permitan evitar depender de mediciones externas. Durante el presente proyecto nos hemos apoyado en plataformas externas como AEMET (Agencia Estatal de METeorología) y OpenData Madrid para recoger y enriquecer los datos. En un futuro, sería deseable no depender de estas plataformas locales si no que los dispositivos tuvieran sensores que ofrecieran la recuperación de estos datos. Con este avance se aumentaría el nivel de escalabilidad del proyecto y en principio solo habría que colocar los dispositivos directamente y esperar la recogida de datos para empezar con el modelado de la ciudad, lo cual sería un gran avance en la usabilidad y escalabilidad a costa de aumentar ligeramente el precio de cada dispositivo.

Por último, una tercera línea de trabajo sería el de incorporar al proyecto, no solo la afluencia entrante y saliente de personas de Ciudad Universitaria, sino la capacidad de saber cómo se mueven dentro de ella. Esto se podría conseguir colocando dispositivos infrarrojos de medición de personas en las entradas a las distintas facultades y edificios. También se podrían situar, dentro de las facultades, en lugares estratégicos como las cafeterías, bibliotecas, laboratorios, salas de grados, etc. Con esto conseguiríamos, en primer lugar, validar la correspondencia entre las personas que entran y salen del metro con aquellas que asisten a la universidad diferenciando así a alumnos, profesores, etc. de personas que no tienen relación con la universidad; y, en segundo lugar conocer cuál es la distribución de las personas

dentro de Ciudad Universitaria y dentro de sus facultades. Sería una forma interesante de conocer, además del flujo externo, también el interno y pasar de una *smart city* vista como un sistema de caja negra a un sistema de caja blanca.

Bibliografía

- [1] Mohammad Aazam and Eui-nam Huh. E-hamc: Leveraging fog computing for emergency alert service. pages 518–523, 03 2015.
- [2] Mohammad Abdullah Al Faruque and Korosh Vatanparvar. Energy management-as-a-service over fog computing platform. 3:1–1, 04 2015.
- [3] Kevin Ashton. That 'internet of things' thing. <http://www.rfidjournal.com/articles/view?4986>, 2009.
- [4] OpenIoT Consortium. Openiot. <http://www.openiot.eu/>, 2012.
- [5] A. Copie, T. F. Forti?, and V. I. Munteanu. Benchmarking cloud databases for the requirements of the internet of things. pages 77–82, June 2013. ISSN 1334-2762. doi: 10.2498/iti.2013.0546.
- [6] Dave Evans. The internet of things how the next evolution of the internet is changing everything, 2011.
- [7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions, 2013.
- [8] MSV. Janakiram. is fog computing the next big thing in the internet of things. <https://www.forbes.com/sites/janakirammsv/2016/04/18/is-fog-computing-the-next-big-thing-in-internet-of-things>, 2016.
- [9] Maxbotix. Lv-maxsonar-ez1 datasheet. https://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf, 2017.

- [10] OpenFog Consortium. OPFRA001.020817. Openfog reference architecture for fog computing. <http://www.openfogconsortium.org/ra/>, 2017.
- [11] Panasonic. Gp2y0a710k0f datasheet. http://www.socle-tech.com/doc/IC%20Channel%20Product/Sensors/Distance%20Measuring%20Sensor/Analog%20Output/gp2y0a710k_e.pdf, 2017.
- [12] Panasonic. Amg8853 datasheet. https://media.digikey.com/pdf/Data%20Sheets/Panasonic%20Sensors%20PDFs/Grid-EYE_AMG88.pdf, 2018.
- [13] Charith Perera, Yongrui Qin, Julio C. Estrella, Stephan Reiff-Marganiec, and Athanasios V. Vasilakos. Fog computing for sustainable smart cities: A survey. *ACM Computer Survey*, 50(3):32:1–32:43, June 2017. ISSN 0360-0300. doi: 10.1145/3057266. URL <https://arxiv.org/pdf/1703.07079.pdf>.
- [14] Commonwealth Scientific and Industrial Research Organisation (CSIRO). Phenonet. <http://www.plantphenomics.org.au/>, 2011.
- [15] Marcelo Yannuzzi, Frank van Lingen, Anuj Jain, Oriol Lluch Parellada, Manel Mendoza Flores, David Carrera, Juan Luis Perez, Diego Montero, Pablo Chacin, Angelo Corsaro, and Albert Olive. A new era for cities with fog computing. *IEEE Internet Computing*, 21(2):54–67, March 2017. ISSN 1089-7801. doi: 10.1109/MIC.2017.25. URL <https://doi.org/10.1109/MIC.2017.25>.