

# Vehicle route optimization system with stochastic demands including predictive models



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

**Mención Internacional**

Autor

**Álvaro Lozano Murciego**

Directores

**Juan Francisco de Paz Santana**

**Javier Bajo Pérez**

Departamento de Informática y Automática

Universidad de Salamanca

*Tesis Doctoral*



## Declaración de autoría

D. Álvaro Lozano Murciego presenta el proyecto de tesis titulado *Vehicle route optimization system with stochastic demands including predictive models* para optar al Grado de Doctor en Ingeniería Informática por la Universidad de Salamanca, y declara que ha sido realizado bajo la dirección del Dr. Juan Francisco de Paz Santana, Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca, y el Dr. Javier Bajo Pérez, Profesor Titular en el Departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid.

### EL DOCTORANDO

### LOS DIRECTORES

D. Álvaro Lozano Murciego

Dr. Juan Francisco de Paz Santana

Dr. Javier Bajo Pérez

# Agradecimientos

Desde que comenzó el desarrollo de este trabajo de tesis han sido muchas las personas que me han ayudado y apoyado, ya sea de forma directa o indirecta, para que ésta llegase a buen término y a las que quiero mostrar mi más sincero agradecimiento:

A mis directores de tesis Dr. Juan Francisco de Paz Santana y Dr. Javier Bajo Pérez que con su experiencia y conocimiento han sabido guiarme y ayudarme en cada escollo en el camino logrando que mi trabajo llegase a buen puerto.

A Juan Manuel Corchado por darme la oportunidad de trabajar en el Grupo de Investigación BISITE donde, durante algo más de 6 años, he podido conocer a muy buenos compañeros y amigos que me han enseñado mucho. Tanto a los que ya no están, como a los que siguen, muchísimas gracias.

A Gabri, Barri y Dani, que me han apoyado y ayudado todos estos años y a los que nunca les estaré lo suficientemente agradecido. A Javier Caridad que me ha enseñado el apasionante mundo de la electrónica y a Diego con el que siempre pongo a prueba mis conocimientos de Machine Learning. A todos los compañeros y amigos del grupo ESALAB que siempre de manera desinteresada me han ayudado, mil gracias.

Al resto de personas que han estado ahí en las buenas y en las malas: mis amigos del GRUPO y mi familia, especialmente a mis padres que me enseñaron el valor del esfuerzo y la honestidad y con esto espero recompensar un poco de todo lo que me han dado. A mi hermana por su ayuda y apoyo incondicional y a Tania por su incommensurable cariño y paciencia.

A todos los que os sintáis parte de este trabajo, toda mi gratitud.



# Resumen

Durante los últimos años se ha producido un incremento en el número de ciudades que están incorporando nuevos sistemas basados en el concepto conocido como Internet de las Cosas o *Internet of Things* (IoT) para, por un lado, obtener nuevos datos acerca de la ciudad y, por otro, a partir de estos, ofrecer nuevos servicios y optimizar el consumo energético. Estas ciudades se mueven hacia un paradigma de ciudad inteligente o Smart City, cuyo principal objetivo es conseguir ciudades más sostenibles y que constituyan un mejor lugar donde vivir. Entre todas las aplicaciones que han surgido para *Smart Cities*, cabe destacar las destinadas a la logística inteligente, que buscan el ahorro y la eficiencia energética en el transporte realizado por las flotas de vehículos.

Unido a esto, nuevos sensores han emergido para llevar a cabo proyectos que incorporen el concepto de IoT y la conexión de cualquier objeto de la vida diaria a Internet. Estos sensores o nodos se agrupan en las conocidas como *Wireless Sensor Networks* (WSN) o redes inalámbricas de sensores. Dichas redes pueden desplegarse tanto en entornos rurales como en urbanos y suponen la infraestructura básica para la toma de datos en aplicaciones para *Smart Cities*. En la presente tesis se realiza una investigación sobre el uso indicado de los sensores junto con las WSN para la toma de datos en sistemas de recogida de desechos.

Asimismo, el procesamiento de los datos obtenidos por dichas redes y la extracción de patrones que permitan modelar su comportamiento ofrecen una valiosa información que puede emplearse más tarde para mejorar otros sistemas, como, por ejemplo, sistemas de optimización de rutas de vehículos. En este trabajo de tesis se aborda la utilización de modelos de predicción para predecir la demanda y que esta información pueda proporcionarse a otros sistemas para su posterior uso.

Por último, los datos obtenidos a través de una red de sensores y la información extraída gracias a los modelos de predicción habilitan la inclusión de nuevos métodos de optimización de rutas de vehículos en un sistema de recogida inteligente de desechos o *Smart Waste Collection System* en inglés. La optimización de las rutas de vehículos en este tipo de sistemas se formula en la literatura como un problema de rutas de vehículos o *Vehicle Routing Problem* (VRP), en el que generalmente sus parámetros son de naturaleza estocástica, pero con frecuencia son tratados de forma determinista. En esta tesis se aborda la resolución de problemas VRP que incluyan incertidumbre en la demanda de sus clientes. Para ello, se emplean nuevas metodologías propuestas en la literatura como Simheuristics y se proponen modificaciones en su utilización incluyendo modelos de predicción con el fin de obtener mejores resultados.



# Abstract

In recent years there has been an increase in the number of cities that are incorporating new systems based on the concept known as the Internet of Things (IoT) to, on the one hand, obtain new data about the city and, on the other hand, from these, offer new services and optimize energy consumption. These cities are moving towards an intelligent city paradigm called Smart City, whose main objective is to achieve more intelligent and sustainable cities that constitute a better place to live. Among all the applications that have arisen for Smart Cities, it is worth highlighting those aimed at intelligent logistics, which seek savings and energy efficiency in transport performed by vehicle fleets.

Along with this, new sensors have emerged to carry out projects that incorporate the concept of IoT and the connection of an object of daily life to the Internet. These sensors or nodes are grouped into what are known as Wireless Sensor Networks (WSN). These networks can be deployed in both rural and urban environments and provide the essential infrastructure for data collection in applications for Smart Cities. In the present thesis, a research is conducted on the indicated use of the sensors together with the WSN for the data collection in waste collection systems.

Likewise, the processing of the data obtained by these networks and the extraction of patterns that allow modelling their behaviour offer valuable information which can be used later to improve other systems such as routing optimization systems. This thesis work deals with the use of prediction models to predict demand and then provide this information to other systems for later use.

Finally, the data obtained through a network of sensors and the information extracted through predictive models enable the inclusion of new methods of vehicle routing optimization in an intelligent waste collection system or Smart Waste Collection System. The optimization of vehicle routes in this type of system is formulated in the literature as a Vehicle Routing Problem (VRP), in which its parameters are generally stochastic in nature, but are treated deterministically. This thesis deals with the resolution of VRP problems that include uncertainty in the demand of their customers. To this end, new methodologies in the literature such as Simheuristics are used and, in this work, modifications are proposed including prediction models in their use in order to obtain better results.



# Contenido

Contenido.....	9
Lista de figuras.....	13
Lista de tablas .....	15
Capítulo 1 Introducción .....	19
1.1 Introducción .....	19
1.2 Hipótesis y objetivos .....	21
1.3 Motivación .....	23
1.4 Metodología .....	24
1.5 Estructura de la memoria.....	26
Capítulo 2 Estado del arte.....	31
2.1 Introducción .....	31
2.2 Obtención de datos en redes inalámbricas de sensores .....	31
2.2.1 Redes inalámbricas de sensores .....	32
2.2.2 Redes de área amplia y bajo consumo LPWAN.....	34
2.2.3 Aplicaciones de redes inalámbricas de sensores LPWAN .....	42
2.2.4 Toma de datos en sistemas de recogida de residuos .....	43
2.3 Modelado predictivo .....	45
2.3.1 Random Forest Regressors .....	45
2.3.2 ExtraTree Regressors .....	49
2.3.3 Gradient Boosting Regressor .....	49
2.3.4 Support Vector Regressors .....	51
2.3.5 Redes neuronales artificiales.....	52
2.4 Optimización de rutas de vehículos .....	54
2.4.1 Definición de VRP .....	54
2.4.2 Variantes del problema VRP .....	56
2.4.3 Métodos de resolución .....	59
2.4.3.1 Métodos exactos .....	60
2.4.3.2 Métodos aproximados.....	62
2.4.3.3 Heurísticas .....	62

2.4.3.4	Metaheurísticas .....	64
2.4.4	La incertidumbre en el CVRP .....	67
2.4.5	Simheuristic .....	70
Capítulo 3	Sistema propuesto .....	73
3.1	Arquitectura general del sistema .....	73
3.2	Subsistema de obtención de datos .....	75
3.2.1	Red inalámbrica de sensores LoRaWAN y sensor de bajo consumo .....	77
3.3	Subsistema de predicción de la demanda .....	81
3.4	Subsistema de optimización de rutas bajo incertidumbre .....	83
Capítulo 4	Casos de estudio .....	89
4.1	Obtención de datos mediante sensores LoRaWAN .....	89
4.1.1	Resultados obtenidos del nodo desarrollado .....	91
4.1.2	Estudio de la cobertura de la red inalámbrica de sensores .....	93
4.1.3	Resultados del sistema de optimización de rutas .....	97
4.1.4	Conclusiones .....	101
4.2	Predicción de la demanda en sistemas de Bike-Sharing .....	103
4.2.1	Sistema de alquiler de bicicletas de Salamanca: SalenBici .....	104
4.2.2	MAS de predicción de la demanda y visualización de rutas en BSS .....	104
4.2.3	Proceso de obtención de datos .....	107
4.2.4	Selección de modelos de predicción .....	110
4.2.5	Discusión de resultados obtenidos .....	111
4.2.6	Herramienta de visualización y predicción .....	116
4.2.7	Conclusiones .....	118
4.3	Optimización de rutas de recogida con demandas estocásticas combinando metodología Simheuristics y modelos de predicción .....	119
4.3.1	Combinación de Simheuristics con modelos de predicción .....	119
4.3.2	Caso de estudio .....	122
4.3.2.1	Simulación de los escenarios .....	123
4.3.2.2	Generación de los problemas deterministas CVRP .....	125
4.3.2.3	Cálculo de soluciones de problemas CVRP y evaluación frente a escenarios	127
4.3.3	Conclusiones y líneas futuras .....	130
Capítulo 5	Conclusiones .....	135
5.1	Contribuciones del trabajo de tesis .....	135

5.2	Líneas de trabajo futuras .....	138
Capítulo 6	Research overview .....	141
6.1	Introduction .....	141
6.1.1	Hypothesis and objectives.....	142
6.1.2	Motivation.....	144
6.1.3	Methodology.....	145
6.2	Background .....	147
6.2.1	Data collection WSNs.....	147
6.2.1.1	Wireless Sensor Networks.....	147
6.2.1.2	Low Power Wide Area Networks.....	148
6.2.1.3	Application of wireless sensor networks (LPWAN) .....	154
6.2.1.4	Data collection in Smart Waste Collection Systems .....	155
6.2.2	Demand Prediction.....	157
6.2.2.1	Random Forest Regressors .....	157
6.2.2.2	Extra Tree Regressors.....	157
6.2.2.3	Gradient Boosting Regressor.....	158
6.2.2.4	Support Vector Regressors .....	159
6.2.2.5	Artificial Neural Networks .....	160
6.2.3	CVRP, uncertainty and Simheuristics.....	162
6.2.3.1	VRP formal definition .....	162
6.2.3.2	VRP variants.....	164
6.2.3.3	Solving methods .....	166
6.2.3.4	Uncertainty in CVRP.....	167
6.2.3.5	Simheuristic methodology.....	169
6.3	Proposed system.....	171
6.3.1	System architecture .....	171
6.3.2	Collection data subsystem.....	173
6.3.2.1	Wireless Sensor Network and LoRaWAN sensor proposed .....	174
6.3.3	Demand prediction subsystem .....	178
6.3.4	Optimization collection routes under uncertainty subsystem .....	180
6.4	Case studies .....	183
6.4.1	Data collection by using LoRaWAN WSN .....	183
6.4.1.1	Measurement results from the developed sensor.....	185

6.4.1.2	Network coverage study .....	187
6.4.1.3	Route optimization results .....	190
6.4.1.4	Conclusions .....	192
6.4.2	Demand prediction in Bike Sharing Systems .....	194
6.4.2.1	Bike Sharing System of Salamanca: SalenBici .....	194
6.4.2.2	MAS for demand prediction and route visualization in BSS .....	195
6.4.2.3	Data collection process .....	198
6.4.2.4	Predictive model selection.....	201
6.4.2.5	Obtained results discussion .....	202
6.4.2.6	Visualization and demand prediction web tool .....	206
6.4.2.7	Conclusions .....	208
6.4.3	Simheuristics combined with demand predictive models.....	210
6.4.3.1	Combination of Simheuristics with prediction models .....	210
6.4.3.2	Case study.....	213
6.4.3.3	Solving and evaluating against scenarios CVRP instances .....	219
6.4.3.4	Conclusions .....	221
6.5	Conclusions and future lines of work.....	223
6.5.1	Research contributions.....	223
6.5.2	Future lines of work .....	226
	Bibliografía .....	227
	Glosario.....	241
	Siglas y acrónimos .....	243
	ANEXO I: Publicaciones y trabajos relacionados .....	249



## Lista de figuras

<b>Figura 1</b> Esquema básico de comunicación de una red inalámbrica de sensores .....	32
<b>Figura 2</b> Pila de protocolos de las redes inalámbricas de sensores [Wang and Balasingham 2010] .....	33
<b>Figura 3</b> Gráfica de tecnologías inalámbricas en función de la transferencia de datos y el rango .....	35
<b>Figura 4</b> Diagrama básico de la arquitectura LoRaWAN .....	39
<b>Figura 5</b> Pila del protocolo LoRaWAN .....	40
<b>Figura 6</b> Toma de datos en sistemas de recogida de desechos .....	44
<b>Figura 7</b> (Parte superior) Salida de un Árbol de regresión (Parte inferior) Árbol de regresión .....	46
<b>Figura 8</b> Aplicación del algoritmo SVR a un conjunto de datos empleando distintos $\varepsilon$ .....	52
<b>Figura 9</b> Diagrama simple de la arquitectura de un perceptrón multicapa (MLP) .....	53
<b>Figura 10</b> Representación gráfica de los parámetros de entrada de un problema CVRP clásico y una posible solución.....	55
<b>Figura 11</b> Diagrama general de los métodos de resolución para problemas VRP.....	59
<b>Figura 12</b> Esquema completo de la metodología <i>Simheuristic</i> propuesta por Juan et al. [Juan et al. 2015b] .....	71
<b>Figura 13</b> Diagrama general del sistema propuesto.....	74
<b>Figura 14</b> Diagrama general del sistema de obtención de datos.....	76
<b>Figura 15</b> Diagrama de comunicación general del sistema propuesto.....	78
<b>Figura 16</b> Localización de los módulos funcionales del nodo y las placas de desarrollo empleadas para el prototipo. ....	79
<b>Figura 17</b> Diagrama de una célula de carga y el circuito eléctrico asociado.....	80
<b>Figura 18</b> Predicción de la demanda a partir de datos históricos de recogida .....	82
<b>Figura 19</b> Diagrama de la metodología para el entrenamiento de modelos, selección de hiperparámetros y selección de los modelos de predicción .....	82
<b>Figura 20</b> Diagrama de la metodología <i>Simheuristics</i> incluyendo modelos de predicción....	84
<b>Figura 21</b> Rutas de recogida generadas para el caso de estudio en la provincia de Salamanca .....	90
<b>Figura 22</b> Municipios del caso de estudio clasificados por su altitud .....	94
<b>Figura 23</b> Mapa de cobertura de la red propuesta realizado con la herramienta Radio Mobile .....	95
<b>Figura 24</b> Configuración de las estaciones base en la herramienta Radio Mobile .....	96
<b>Figura 25</b> Configuración de los nodos en la herramienta Radio Mobile .....	96
<b>Figura 26</b> Simulación de la generación de residuos, selección de nodos, instancias CVRP y sus soluciones.....	98
<b>Figura 27</b> Resultados de cada día de recolección frente al coste total fijo de rutas estáticas .....	100
<b>Figura 28</b> Ahorros para cada día de recolección.....	100

<b>Figura 29</b> (a) Uno de los camiones de balanceo de bicicletas de SalenBici. (b) Mapa situación de las estaciones del sistema facilitado por el ayuntamiento de Salamanca .....	104
<b>Figura 30</b> Diagrama general del sistema Multi-Agente propuesto .....	105
<b>Figura 31</b> Diagrama de secuencia que representa la carga de datos <i>trip data</i> en el sistema multi-agente .....	108
<b>Figura 32</b> Número total de eventos (salidas o llegadas) realizados en BSS SalenBici desde enero de 2013 hasta marzo de 2017 .....	109
<b>Figura 33</b> Periodos de tiempo de funcionamiento de cada estación en el BSS SalenBici....	109
<b>Figura 34</b> Esquema de selección de hiperparámetros y entrenamiento aplicado al caso de estudio de BSS .....	110
<b>Figura 35</b> RMSLE obtenido para cada algoritmo de cada estación en los datos de llegadas .....	112
<b>Figura 36</b> RMSLE obtenido para cada algoritmo de cada estación en los datos de salidas .	112
<b>Figura 37</b> Coeficiente de determinación de los modelos en las llegadas de cada estación ..	113
<b>Figura 38</b> Coeficiente de determinación de los modelos en las salidas de cada estación....	113
<b>Figura 39</b> Test U de Mann-Whitney ( $H_0$ igualdad) para (a) llegadas (b) salidas.....	114
<b>Figura 40</b> Test U de Mann-Whitney ( $H_0$ menor que) para (a) llegadas (b) salidas. ....	114
<b>Figura 41</b> Diagrama de secuencia del proceso de generación de predicciones de la demanda. ....	115
<b>Figura 42</b> Aplicación Web servida por el WebAPI Agent. Sección visualización de datos históricos .....	116
<b>Figura 43</b> Sección de visualización. Selección de estaciones individuales y periodo de tiempo y visualización de cantidad de viajes a cada estación.....	117
<b>Figura 44</b> Sección de predicción de la demanda. Demanda para cada estación. ....	117
<b>Figura 45</b> Proceso de simplificación seguido en la metodología Simheuristics.....	120
<b>Figura 46</b> Proceso de simplificación propuesto incluyendo modelos de predicción para la metodología Simheuristics.....	121
<b>Figura 47</b> Esquema principal del diseño del caso de estudio .....	122
<b>Figura 48</b> Ejemplo de los datos generados para 5 recogidas cada 3 días. ....	124
<b>Figura 49</b> Datos empleados de cada contenedor para obtener la media para el proceso de simplificación.....	125
<b>Figura 50</b> Ejemplo de resultados de los regresores para un contenedor del problema A-n32-k5.vrp .....	127
<b>Figura 51</b> Coste de simulación medio (CSM) para las soluciones de cada día de recolección (arriba) y la factibilidad de las soluciones en la simulación (abajo).....	129
<b>Figura 52</b> Boxplots del Coste Medio de Simulación para cada uno de los métodos e instancias CVRP estudiadas.....	130

## Lista de tablas

<b>Tabla 1</b>	Comparación de tecnologías para WSN .....	37
<b>Tabla 2</b>	Opciones disponibles para un servidor de red LoRaWAN.....	42
<b>Tabla 3</b>	Envíos y medidas de datos de sensores por día y sus consumos teórico y empírico.	93
<b>Tabla 4</b>	Parámetros para la generación de los escenarios para instancia A-n32-k5.vrp .....	123
<b>Tabla 5</b>	Características de entrada para los regresores .....	126
<b>Tabla 6</b>	Parámetros para la generación de los escenarios .....	126
<b>Tabla 7</b>	Parámetros empleados en el solver .....	128



---

# CAPÍTULO 1

---



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

# INTRODUCCIÓN

---



# Capítulo 1 Introducción

Este primer capítulo introduce la temática en la que se encuentra englobado el trabajo de tesis y plantea la hipótesis, los objetivos y la metodología que se empleará a lo largo de su desarrollo.

## 1.1 Introducción

Durante los últimos años se ha producido un incremento en el número de ciudades que están incorporando nuevos sistemas basados en el conocido como Internet de las Cosas o *Internet of Things* (IoT) para, por un lado, obtener nuevos datos acerca de la ciudad y, por otro, ofrecer nuevos servicios y optimizar el consumo energético a partir de dichos datos. Estas ciudades se mueven hacia un paradigma de ciudad inteligente o *Smart City* [Silva et al. 2018] cuyo principal objetivo es conseguir ciudades más sostenibles y que constituyan un mejor lugar donde vivir.

Entre todas las aplicaciones que han surgido para *Smart Cities*, se encuentran las destinadas al control de flujo de personas en las ciudades [Alvarez-Campana et al. 2017]; al control de aparcamiento [Bagula et al. 2015]; a la obtención de información sobre lugares accesibles [Mora et al. 2017]; al control de iluminación en domótica [Vinagre et al. 2016] y de alumbrado público [De Paz et al. 2016] junto al uso de *Smart Grids*; a la gestión del agua, de los residuos y de los servicios de salud; a la logística inteligente y un largo etcétera.

Pueden ser utilizadas tanto en grandes ciudades como en zonas rurales, donde cada vez es más común encontrar aplicaciones relacionadas con la agricultura de precisión, el *Smart Farming* [Villarrubia et al. 2017b] y el control de ganado inteligente [Musat et al. 2018]. Al mismo tiempo, nuevos sensores han emergido para poder hacer posible este tipo de proyectos junto con el concepto de IoT y la conexión de cualquier objeto de la vida diaria a Internet. Estos sensores o nodos se agrupan en las conocidas como *Wireless Sensor Networks* (WSN) o redes inalámbricas de sensores en castellano [Buratti et al. 2009]. Estas redes inalámbricas pueden ser desplegadas en entornos rurales al igual que en urbanos y componen la infraestructura básica para la toma de datos en aplicaciones para *Smart Cities*. La investigación en el uso indicado de este conjunto de sensores junto a WSN para la toma de datos y la eficiencia en su consumo energético son un punto crítico hoy en día. El principal reto para muchas aplicaciones empleadas en *Smart Cities* se centra en que estos sensores posean una eficiencia energética que permita una mayor autonomía de la batería, llegando a ser incluso de varios años. Entre todas las aplicaciones relacionadas con las *Smart Cities*, destacan las relacionadas con la logística

inteligente: control de reparto de bienes, sistemas de gestión de recogida de desechos [Gutierrez et al. 2015], sistemas de alquiler de bicicletas (*Bike Sharing Systems*) [Pal and Zhang 2017][Lin and Chou 2012], etc. En estas aplicaciones se busca optimizar al máximo los recursos disponibles a la vez que se provee a los ciudadanos de un servicio adecuado. En el caso de la recogida inteligente de desechos o *Smart Waste Collection* en inglés, al igual que en el resto de las aplicaciones de logística, existen tres partes fundamentales para dotar de inteligencia y eficiencia a estos sistemas: eficiencia en el sistema de toma de datos, modelado y predicción de la demanda a partir de los datos obtenidos y sistema de optimización de rutas.

En primer lugar, es necesario dotar de sensores eficientes al sistema para permitir que el mantenimiento de estos no suponga una traba a la hora de su implantación. Asimismo, se requiere el uso de WSN adecuadas para que la creación de sensores con un bajo consumo sea posible. En segundo lugar, es preciso modelar la demanda existente en estos sistemas. La demanda hace referencia a la cantidad de bienes que será necesario recoger o depositar dependiendo del sistema de logística concreto. Estos sistemas suelen modelarse formalmente como problemas de rutas de vehículos, en inglés *Vehicle Routing Problems* (VRP) [Eksioglu et al. 2009]. En ellos, una flota de vehículos debe dar respuesta a la demanda de una serie de clientes, sujeta a unas restricciones y tratando de emplear el mínimo número de recursos posibles. Conocer y modelar la demanda de estos problemas resulta un punto crítico para la posterior optimización de las rutas. Por último, los métodos empleados habitualmente en la literatura y en el software comercial disponible a menudo no contemplan la introducción de la incertidumbre en los métodos de resolución de dichos problemas. Existe una creciente línea de investigación en este apartado con cada vez más trabajos publicados que pretende reflejar la incertidumbre presente en las variables empleadas en los VRP. Entre los distintos métodos de resolución que contemplan la incorporación de incertidumbre al problema, se encuentran los métodos de optimización estocástica [Erkin and Koole 2017], optimización robusta [Solano-Charris 2015] y nuevas metodologías [Juan et al. 2015a] que pretenden dar un enfoque alternativo y más flexible frente a las dos primeras.

Por lo tanto, es necesario investigar (1) nuevas formas de toma de datos con sensores que permitan una mayor eficiencia y duración de sus baterías, (2) sistemas de predicción de la demanda que permitan modelarla y, finalmente, (3) sistemas de optimización que utilicen la información procesada y que tengan en cuenta la incertidumbre que está presente en los problemas de recogida reales



## 1.2 Hipótesis y objetivos

En la actualidad, es posible encontrar sistemas de recogida de desechos que emplean sensores para medir su nivel de llenado, pero es necesario que también incluyan una gran eficiencia en su batería y que puedan tomar medidas relativas al peso del contenedor ya que podrían ser útiles en la toma de datos de estos sistemas. Además, es necesario investigar sobre modelos de predicción que puedan predecir de forma certera la demanda generada por los clientes en los problemas de enrutado de vehículos. Por último, una gran mayoría de los trabajos presentes en la literatura no tienen en cuenta los aspectos relativos a la incertidumbre en los problemas de rutas de vehículos VRP, por lo que en este trabajo se pretende incorporar estos aspectos a los métodos de resolución de dichos problemas y avanzar en estas líneas de investigación.

Por tanto, se plantea la siguiente hipótesis como base de esta tesis doctoral:

*“El diseño de sistemas basados en redes inalámbricas de sensores de bajo coste y bajo consumo energético, que incluya sistemas de modelado y predicción de la demanda y que incorpore la incertidumbre en sus métodos de resolución produce ahorros en energía, costes, personal y tiempo frente a los sistemas de logística convencional.”*

Para poder validar esta hipótesis se deberá abordar una serie de objetivos generales que se describen a continuación:

- Analizar la toma de datos para sistemas de logística inteligente estudiando las redes inalámbricas de sensores adecuadas para largas distancias y sensores con una alta eficiencia energética. Al mismo tiempo, investigar en el diseño de sensores que tomen datos de forma eficiente y con un consumo energético reducido.
- Estudiar el comportamiento de la demanda en los problemas de logística relacionados con VRP y diseñar modelos que, a partir de los datos tomados y otros adicionales relativos al clima o datos temporales, sean capaces de predecir de forma precisa demandas futuras.
- Incorporar la incertidumbre en los parámetros de los problemas de rutas de vehículos VRP e incluir métodos de resolución de estos problemas que provean de soluciones robustas, teniendo en cuenta la estocasticidad de sus variables en el mundo real.
- Diseñar un sistema que aúne lo anterior y que ofrezca ahorros en costes, energía y mano de obra frente a un enfoque convencional. El desarrollo de un caso de estudio con el que validar la hipótesis del presente trabajo.

Estos objetivos generales se llevarán a cabo a través de una serie de objetivos específicos que permitan concretar las tareas que se deben desarrollar a lo largo de la presente tesis. Se detallan a continuación dichos objetivos específicos:

- Estudio del estado del arte de redes inalámbricas de sensores para la toma de datos en sistemas de logística inteligente.
- Estudio del arte de redes inalámbricas de sensores de bajo consumo (LPWAN) y su aplicación en sistemas de logística inteligente.
- Estudio de algoritmos de modelado predictivo de la demanda que puedan ser empleados en sistemas de logística inteligente.
- Estudio del estado del arte de los problemas VRP y sus diferentes variantes.
- Estudio del estado del arte de los problemas VRP con incertidumbre.
- Diseño de un sistema que incorpore sensores de bajo coste y consumo energético para la toma de datos
- Diseño de un sistema que incorpore modelos de predicción que permitan modelar la demanda en sistemas de logística inteligente.
- Diseño de un sistema de optimización de rutas de vehículos que incorpore la incertidumbre en sus soluciones.
- Diseño y desarrollo de un caso de estudio que emplee sensores de bajo consumo energético y redes inalámbricas de sensores de largo alcance.
- Diseño y desarrollo de un caso de estudio de un sistema de predicción de la demanda.
- Diseño y desarrollo de un caso de estudio que incorpore los sistemas previos, como son la toma de datos, los sistemas de predicción y la incertidumbre, a los métodos de resolución de los problemas de rutas de vehículos.

La consecución exitosa de estos objetivos permitirá obtener y evaluar los resultados producidos de cada uno de los casos de estudio y contrastar la hipótesis planteada en este trabajo de tesis doctoral.

### 1.3 Motivación

El auge del IoT ha propiciado la incorporación de sensores a multitud de ámbitos en las *Smart Cities*. Esto ha supuesto la creación de WSN en las que los nodos juegan un papel crucial en la toma de datos y en la actuación, si de un actuador se tratase. Una limitación que ha surgido de manera inminente en el despliegue de estos sensores ha sido la duración de la batería: en casos de uso como la implantación de sensores en un contenedor, en estaciones meteorológicas aisladas o en agricultura de precisión no es posible el acceso a suministro eléctrico y es necesaria una batería con la máxima duración posible para reducir el coste de mantenimiento de estos sistemas. Debido a este hecho, nuevas redes LPWAN [Sinha et al. 2017] han surgido recientemente para dar respuesta a esta limitación y permitir la inclusión de sensores con baterías con autonomía de, incluso, varios años. El desarrollo de sensores que utilicen este tipo de redes es vital para la creación de sistemas que posteriormente los empleen en su funcionamiento, como son los sistemas de recogida inteligente de residuos.

Directamente relacionado con esto, se encuentra la información que aportan los datos recogidos por estos sensores y sus posibles usos en la mejora de los sistemas de recogida. Disponer de datos históricos de la producción de desechos de cada contenedor (denominado demanda en problemas VRP) unido a otras variables, como pueden ser el tiempo climatológico o el calendario anual, permiten la creación de modelos que puedan llegar a predecir dicha producción con suficiente certeza. Estos modelos permiten anticipar posibles llenados en los contenedores y realizar una recogida más eficiente e inteligente teniendo en cuenta dicha información.

Por último, la información de la que se dispone en un problema VRP es, en general, estocástica, aunque, de manera habitual, este tipo de problemas se abordan en la literatura desde la asunción de que la información es determinista. En los últimos años, se ha incrementado el trabajo en líneas de investigación [Berhan et al.] que pretenden incorporar la estocasticidad a las variables de este problema: la demanda (la cantidad de desechos producida en este caso), la matriz de costes (el coste de desplazamiento de un nodo a otro, por ejemplo, teniendo en cuenta el tráfico), los clientes (la necesidad de visitar un determinado contenedor o no). En problemas de recogida de residuos una de las principales variables estocásticas es la demanda: la cantidad de desechos que se va a encontrar un camión cuando llegue al contenedor. Es evidente que en un sistema que posea todos sus contenedores equipados con sensores, la incertidumbre de la demanda será menor ya que se dispone de información reciente y la variabilidad puede deberse a otros motivos como errores de medida o desechos fuera del contenedor. Sin embargo, en la gran mayoría de los casos, los sistemas de recogida poseen contenedores convencionales, de los que solo se conocen datos históricos, por lo que en estos es posible predecir su demanda con los modelos creados y tener en cuenta la incertidumbre en el cálculo de las rutas.

Entre los distintos métodos de resolución que se encuentran en la literatura para abordar problemas VRP estocásticos podemos encontrar principalmente dos: la optimización estocástica [Powell and Topaloglu 2003] y la optimización robusta [Bertsimas and Sim 2004]. El principal problema de estos métodos es que o bien adoptan una solución factible pero demasiado conservadora (optimización robusta) o sus métodos de cálculo de soluciones suponen un gran incremento en coste computacional y no garantizan una solución que satisfaga todas las restricciones (optimización estocástica): el trabajo de Maggioni et al. [Maggioni et al.] ofrece una buena comparativa entre los dos enfoques. Como alternativa a estos dos métodos de resolución, se pueden encontrar otras metodologías en la literatura [Juan et al. 2011] [Juan et al. 2015a] que abordan este problema desde otra perspectiva distinta. Estos emplean la simulación para determinar la factibilidad de una solución creada a priori con datos y algoritmos deterministas y su comportamiento en un gran número de escenarios generados a partir de las distribuciones de probabilidad de cada uno de clientes a visitar. Esto permite utilizar métodos de solución con heurísticas ampliamente estudiadas en la literatura y, asimismo, evaluar el comportamiento de las soluciones bajo incertidumbre.

Este enfoque permite dotar al *decision maker* de herramientas para evaluar distintas soluciones y cómo se comportan, incorporando incertidumbre en las demandas del problema VRP.

## 1.4 Metodología

A lo largo del desarrollo del presente trabajo de tesis se ha empleado como metodología la conocida como *Action Research* [Reason and Bradbury 2008], que ha sido ampliamente utilizada en sistemas de información y está enfocada a la acción y al cambio. En ella, se plantea e identifica el problema a partir de una hipótesis, partiendo de unos conceptos definidos dentro de un modelo cuantitativo de la realidad. Tras esto, se lleva a cabo una recopilación y organización de la información de los conceptos tratados en la hipótesis para después realizar un análisis y diseño de una propuesta destinada a solventar el problema o verificar la hipótesis planteada. Posteriormente, se emplean casos de estudio específicos que permitan poner en práctica la solución propuesta. Por último, se describen las conclusiones obtenidas a partir de los resultados del trabajo de investigación.

Uno de los puntos clave de esta metodología es establecer un conjunto de actividades que tengan como fin llevar a buen término los objetivos inicialmente planteados y la confirmación o refutación de la hipótesis planteada. Estas actividades comprenden:

- Definición de la problemática existente: descripción del problema y sus variables para plantear los objetivos y la hipótesis de trabajo de la investigación.

- Revisión del estado del arte: en este apartado es necesario considerar trabajos previos y conceptos fundamentales en torno a la hipótesis de trabajo para tener una base sólida como punto de partida para la investigación.
- Análisis y diseño de la propuesta de solución.
- Diseño de casos de estudio para validar cada componente de la solución propuesta.
- Ejecución y obtención de resultados de cada uno de los casos de estudio propuestos.
- Estudio, validación y discusión de los resultados obtenidos en los casos de estudio planteados.
- Extracción de conclusiones obtenidas a partir de los resultados de los casos de estudio planteados.
- Diseminación y publicación de los resultados en congresos y revistas científicas.

Dado que estas actividades se ejecutan de manera iterativa a lo largo del proceso de investigación, se puede considerar este proceso como un proceso iterativo e incremental al igual que muchas metodologías de ingeniería del software.

Al aplicar esta metodología de investigación se consigue un doble beneficio ya que se obtienen resultados tras aplicar la solución propuesta a entornos reales que son directamente transferibles al mercado para su uso. Por otro lado, se genera conocimiento relevante que es posible trasladar a otros ámbitos diferentes al considerado en este trabajo.

Simultáneamente al desarrollo de esta tesis, se han elaborado diferentes artículos científicos que incluyen distintos casos de estudio llevados a cabo en el periodo de investigación. Estos artículos han sido publicados en revistas científicas con factor de impacto.

Asimismo, se han desarrollado las tareas planificadas a lo largo del tiempo y se han establecido hitos con el fin de hacer un seguimiento del progreso de dichas tareas y de aplicar medidas correctoras si fuese necesario.

Adicionalmente, se han recibido cursos de formación relacionados con la temática de la tesis, se ha asistido a seminarios y conferencias y se han establecido contactos con investigadores nacionales y extranjeros que han aportado ideas y nuevos enfoques, enriqueciendo la investigación realizada. Asimismo, se ha realizado una estancia de investigación en un centro extranjero con el fin de ampliar los horizontes y disponer de una mayor relación con otros

investigadores que se encuentran actualmente trabajando en áreas de investigación similares y que con sus aportaciones han enriquecido este trabajo de tesis doctoral.

## 1.5 Estructura de la memoria

Con el fin de validar la hipótesis de partida del trabajo y llevar a buen término los objetivos establecidos previamente, se ha estructurado la presente memoria de trabajo de tesis en seis capítulos, siendo este el primero de ellos.

Seguidamente, el capítulo 2 describe el estado del arte de los diferentes conceptos relacionados con la hipótesis de partida, comenzando por la obtención de datos en redes inalámbricas de sensores (WSN) y, concretamente, las LPWAN, seguido por el estudio de modelos empleados para la predicción de la demanda y, a continuación, una revisión del estado del arte de problemas de rutas de vehículos VRP, sus variantes, algoritmos y heurísticas empleadas para solventarlos. Por último, una revisión del estado del arte de las metodologías que proponen incorporar incertidumbre en el planteamiento de estos problemas, identificando fortalezas y debilidades de cada una e introduciendo otras alternativas que se emplearán y modificarán en dicho trabajo.

El capítulo 3 describe el sistema propuesto de logística inteligente para la recolección de desechos incluyendo incertidumbre. Se plantea el sistema desde la toma de datos en los contenedores, el despliegue de redes LPWAN que permitan una alta autonomía de estos sensores, el procesamiento de estos datos y la incorporación de modelos de predicción a partir de ellos y, finalmente, la optimización de las rutas de recogida incluyendo incertidumbre en su cálculo.

En el capítulo 4 se presentan los casos de estudio planteados para validar el sistema propuesto: en ellos se evalúan las diferentes partes del sistema final propuesto en casos de estudio reales. El primer caso de estudio está enfocado a la toma de datos y al desarrollo de un sensor que sea capaz de tomar medidas del nivel de llenado de los contenedores y que posea una gran eficiencia energética. El segundo caso de estudio está destinado a la evaluación de sistemas de predicción de la demanda y el tercer caso de estudio se centra en la incorporación de la incertidumbre junto con los modelos de predicción de la demanda en la optimización de las rutas de vehículos.

En el capítulo 5 de este trabajo se presentan los resultados obtenidos tras los casos de estudio realizados en el capítulo anterior y las conclusiones a las que se ha llegado tras la discusión de dichos resultados.

Finalmente, en el capítulo 6 se presenta un resumen en inglés de la investigación realizada a lo largo de este trabajo de tesis, incluyendo un resumen del estado del arte, el sistema propuesto, los casos de estudio y las conclusiones.





---

## CAPÍTULO 2

---



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

## ESTADO DEL ARTE

---



## Capítulo 2 Estado del arte

Este capítulo describe el estado del arte de las tecnologías y conceptos en los que se apoya este trabajo. Entre ellos podemos encontrar la obtención de datos con redes LPWAN, el modelado predictivo y la optimización de rutas de vehículos

### 2.1 Introducción

Este capítulo recoge el estado del arte de las tecnologías que rodean las tres partes fundamentales del sistema propuesto para un sistema de logística de recogida de residuos inteligente: la obtención de datos en redes de sensores inalámbricas, el modelado predictivo a partir de los datos recogidos y de otras fuentes de información y la optimización de rutas de vehículos.

En la primera sección de este capítulo se estudian los diferentes tipos de redes inalámbricas de sensores existentes, así como sus principales características y aplicaciones en la literatura. Se realiza un análisis de la red empleada según el consumo de batería que se plantea que presente el nodo final. Tras esto, se realiza una revisión de las principales LPWAN empleadas en la actualidad y se estudia la adecuación de cada una de ellas para ser empleadas en el sistema propuesto.

Más adelante, se lleva a cabo una revisión del estado del arte de distintos modelos predictivos de *Machine Learning* para aplicar a la predicción de la demanda en este tipo de problemas. Se revisan distintos algoritmos de predicción ampliamente utilizados en el estado del arte actual que han demostrado obtener buenos resultados.

Por último, se define formalmente el concepto de VRP, se estudia su taxonomía y se profundiza en las vertientes de este problema que incorporan incertidumbre en su planteamiento. Se plantean las distintas alternativas existentes en la literatura para solventar el problema de optimización con incertidumbre y se selecciona la que se incluirá como parte de la propuesta de este trabajo de tesis.

### 2.2 Obtención de datos en redes inalámbricas de sensores

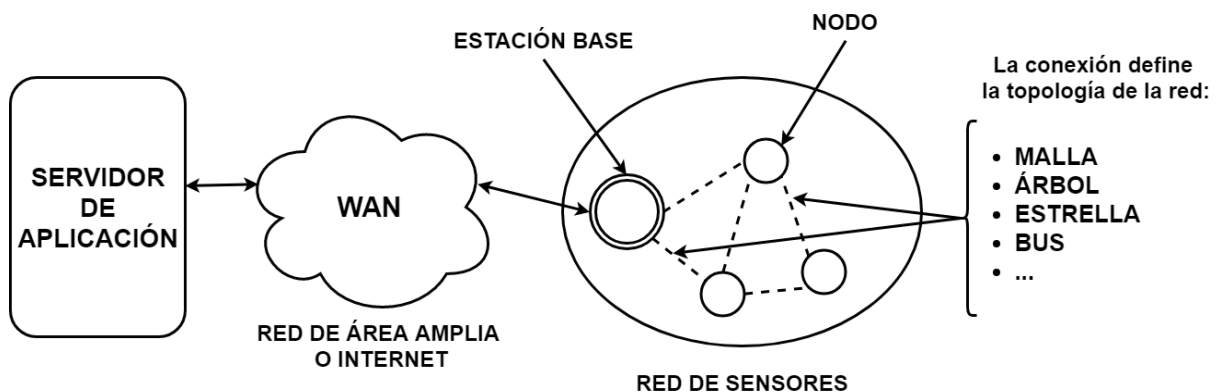
A continuación, se describen las principales redes inalámbricas de sensores disponibles en la literatura, mostrando sus características principales y sus usos para la obtención de datos en

distintos contextos. Más adelante se hace una revisión de las principales LPWANs empleadas en la literatura para la obtención de datos y un estudio comparativo entre ellas teniendo en cuenta su eficiencia energética, cobertura y transferencia de datos para evaluar qué opción es la más idónea para el sistema propuesto.

### 2.2.1 Redes inalámbricas de sensores

Las redes inalámbricas de sensores han ganado popularidad en la comunidad científica debido a que proveen de una infraestructura muy útil para el desarrollo de aplicaciones de monitorización y control. Una red inalámbrica de sensores o *Wireless Sensor Network* en inglés (a partir de ahora WSN) consiste en un largo número de dispositivos con una conexión inalámbrica, conocidos como nodos, que son desplegados para observar o medir cierto fenómeno o detectar determinados eventos y actuar si cuentan con las capacidades necesarias para ello [Mohamed et al. 2017]. Sus componentes básicos, por tanto, son:

- **Nodo:** dispositivo que tiene la capacidad de tomar medidas mediante un sensor o conjunto de sensores, que puede disponer de actuadores para realizar determinadas acciones, con capacidad de procesamiento limitado y que posee una comunicación inalámbrica con una estación base, también conocida como *Gateway* o *Sink* en inglés.
- **Estación base:** responsable de capturar la información transmitida por los nodos y retransmitirla, o proveer de acceso a terceras aplicaciones. Pueden poseer capacidad de almacenamiento y procesamiento (este último se ha ido incrementando debido al auge del *Edge Computing* [Roman et al. 2018] ), además de actuar como pasarelas y proveer acceso y control remoto a los nodos.

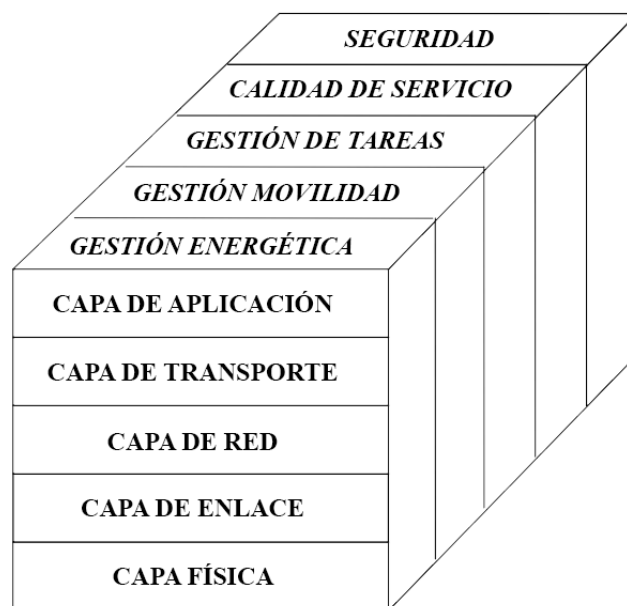


**Figura 1** Esquema básico de comunicación de una red inalámbrica de sensores

En la **Figura 1** se muestra un esquema básico de la configuración de una red inalámbrica de sensores. La comunicación entre el nodo y la estación base u otros nodos se realiza mediante un enlace de radio. Esta comunicación puede darse de manera directa entre el nodo y la estación

base o puede requerir de otros nodos intermedios que retransmitan la información para que esta llegue a su destinatario. El tipo de comunicación entre sensores y la estación base define la topología de la red, siendo las más conocidas: topología en malla, bus, árbol y estrella, entre muchas otras.

De acuerdo con Fahmy et al. [Fahmy 2016] las redes inalámbricas de sensores emplean lo que se conoce como *protocol stack* o pila de protocolos que está íntimamente relacionada con el modelo OSI [Zimmermann 1980] y posee las siguientes capas: física, enlace, red, transporte, y aplicación. La capa física es responsable de la frecuencia de comunicación inalámbrica, de la modulación y la detección de la señal. La capa de enlace es la responsable de la multiplexación de los flujos de datos, de la detección de cada trama de datos, el control de acceso al medio y la corrección de errores. Asegura una conexión fiable punto a punto entre sensores en la red. La capa de red es la encargada de enrutar los paquetes de datos obtenidos de la capa de enlace y debe tener en cuenta aspectos como la eficiencia energética, la centralización de la comunicación, la agregación, el procesamiento de los datos, etc. La capa de transporte es la encargada del mantenimiento de la conexión y de la comunicación con otras redes externas como Internet. Finalmente, la capa de aplicación es la encargada de la utilización de la información recibida y de realizar acciones sobre la red de sensores si esta lo permite.



**Figura 2** Pila de protocolos de las redes inalámbricas de sensores [Wang and Balasingham 2010]

Como se muestra en la **Figura 2**, la pila de protocolos de las redes inalámbricas de sensores debe tener en cuenta otra serie de factores para funcionar eficientemente, que pueden no darse

en otro tipo de redes. Estos son: la movilidad, el consumo energético, la gestión de tareas, la calidad de servicio o *Quality of Service* (QoS) en inglés y la seguridad. La gestión de la movilidad es responsable de detectar y registrar el movimiento de los nodos con el fin de habilitar una ruta de comunicación con las estaciones base. La gestión del consumo energético implica minimizar este y para ello puede ser necesario desactivar ciertas funcionalidades o habilitar periodos para la transferencia de información con una determinada frecuencia. La gestión de tareas es la encargada de programar la activación de ciertos nodos para las tareas de sensorización y que el resto se responsabilice de tareas como la retransmisión y agregación de los datos. La gestión de la calidad del servicio es un aspecto clave si la aplicación requiere de servicios con una interacción en tiempo real. Se encarga de la gestión de tolerancia a fallos y optimización del rendimiento en términos de métricas de calidad del servicio. Por último, un factor importantísimo como es la gestión de la seguridad, encargada de la monitorización y control de la red. Esta incluye la integración de diferentes módulos como la encriptación e integridad de los datos, la autenticación de los nodos en la red y la detección de intrusos.

Las redes inalámbricas de sensores habitualmente emplean estándares de redes inalámbricas de área personal -*Wireless Personal Area Network* (WPAN)- o de redes de área amplia de bajo consumo -*Low Power Wide Area Network* (LPWAN)- para la comunicación entre nodos y estaciones base. No existe una única solución para todas las WSN y la elección de uno de estos estándares depende completamente de los requisitos de comunicación y las restricciones de recursos de cada aplicación en concreto [Pule et al. 2017a].

### 2.2.2 Redes de área amplia y bajo consumo LPWAN

Dentro de las diferentes redes inalámbricas de sensores se ha popularizado el uso de los estándares LPWAN [Sinha et al. 2017] que potencian dos características fundamentales frente al resto de redes: alta eficiencia energética y establecimiento de conexiones a largas distancias.

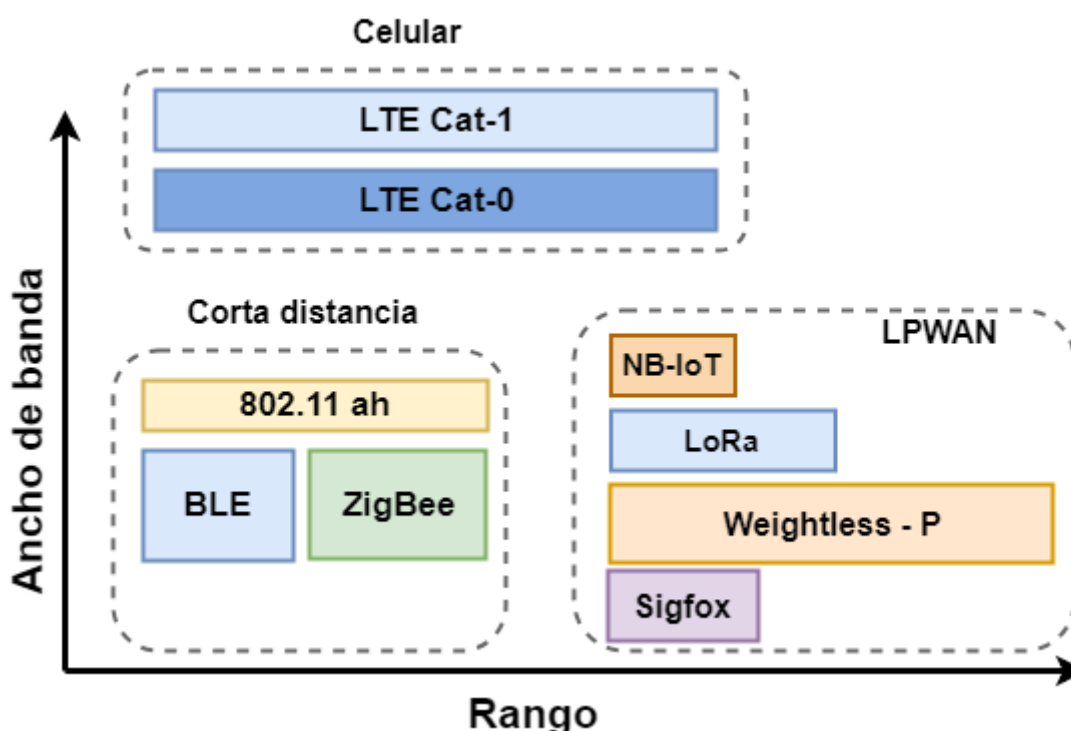
La alta eficiencia energética hace posible desarrollar nodos que porten baterías que permitan su funcionamiento durante largos periodos de tiempo, haciendo que el mantenimiento de estos se vea reducido y puedan ser colocados en lugares aislados de una fuente de energía eléctrica.

Por otra parte, muchas de estas tecnologías permiten conexiones a grandes distancias lo que hace posible dar cobertura a áreas extensas con un despliegue de estaciones base muy limitado. Este factor de cobertura, al igual que el energético, hace posible la instalación de nodos en lugares remotos donde se carezca de fuente de energía eléctrica y conexión a una red móvil.

Sin embargo, estas tecnologías están especialmente indicadas para un tipo de aplicaciones con unos requisitos concretos de cantidad y frecuencia de transferencia de datos. En la **Figura 3** se presenta un diagrama que muestra dónde se encuentran enmarcadas las tecnologías mencionadas junto a los estándares LPWAN, teniendo en cuenta el rango de comunicación y

el ancho de banda requerido. Estas tecnologías están indicadas para aplicaciones en las que las tasas de transferencia sean bajas y se necesite un amplio rango de acción.

En la literatura es posible encontrar múltiples tecnologías que han surgido dentro de esta categoría (LPWAN) entre las que destacan: LTE-M, Sigfox, LoRa (y su especificación LoRaWAN) y NB-IoT [Wang et al. 2017].



**Figura 3** Gráfica de tecnologías inalámbricas en función de la transferencia de datos y el rango

Sigfox [Sigfox 2016] es una tecnología que se sitúa en las redes de comunicaciones de banda estrecha o más conocida en inglés como *Ultra Narrow Band* (UNB) [Walker 2004]. Como se puede ver en la **Figura 3**, posee un rango de comunicación muy amplio y una tasa de transferencia de datos muy limitada. Esta tecnología posee un modelo de negocio similar a un operador de red de telefonía convencional donde este provee señal a un área extensa o un país y los clientes pagan por la transferencia de datos en la red. Este operador puede ser la misma empresa Sigfox, como ocurre en Francia, Alemania o España [SigFox 2017], u otra empresa que pertenezca a la red de empresas autorizadas por Sigfox, como ocurre en el caso de Croacia con la empresa IoTNet o en Irlanda con la empresa VT-IoT.

LoRa (Long Range) [Lora Alliance 2017] [Augustin et al. 2016a] es, por un lado, una tecnología de radio propiedad de la compañía Semtech basada en la modulación *Chirp Spread Spectrum* (CSS). La empresa Semtech adquirió esta tecnología en 2012 y actualmente produce

los transceptores que emplean la tecnología LoRa. Por otro lado, LoRaWAN [Casals et al. 2017] es un protocolo de la capa de acceso al medio, *Media Access Control* (MAC), empleado para manejar la comunicación entre las estaciones base o *gateways* LoRa y los dispositivos. Este protocolo define una arquitectura del sistema y es mantenido por LoRa Alliance [Lora Alliance 2017]. LoRaWAN emplea como capa física el protocolo definido por LoRa.

Al contrario que Sigfox, la especificación LoRaWAN es abierta y puede ser desplegada sin necesidad de pagar por una licencia o por transferencias de datos. Actualmente, muchas plataformas están basadas en la especificación del protocolo LoRaWAN y están disponibles en el mercado, entre otras cabe destacar Actility [Yang 2014], Lorient [LORIENT], LoraServer.io y Senet [Senet 2018].

*Narrow-Band for Internet of Things* (NB-IoT) [Wang et al. 2017] es una tecnología LPWAN orientada para Internet de las Cosas desarrollada por la *3rd Generation Partnership Project* 3GPP. NB-IoT emplea un espectro de frecuencia con licencia actualmente usado por diferentes proveedores de servicios de internet o *Internet Service Providers* (ISPs) en inglés. La red puede ser desplegada de dos formas distintas: “*in-band*” o “*standalone*”. La opción de despliegue “*in-band*” se realiza en un espacio del espectro perteneciente a la tecnología *Long Term Evolution* (LTE). El tipo de desarrollo “*standalone*” implica el uso de una banda dedicada del espectro de frecuencias. La principal ventaja de esta tecnología frente al resto de LPWANs es que está especialmente indicada para aplicaciones que necesiten una mínima latencia y que requieren de comunicaciones frecuentes [Sinha et al. 2017]. Actualmente está siendo desplegada a lo largo de Europa [GSMA 2018] pero su disponibilidad está siendo limitada a áreas urbanas donde se están llevando a cabo test de operatividad. En este tipo de redes, los operadores proveen la red y ofrecen servicios de suscripción como los operadores de telefonía clásicos.

Aparte de las tecnologías mencionadas anteriormente, existen otras que están comenzando a estar disponibles en Europa como *Random Phase Multiple Access* (RPMA) de la compañía Ingenu o *Weightless de Weightless SIG* [Weightless 2017].

La **Tabla 1** muestra una comparación de las diferentes tecnologías mencionadas que son habitualmente empleadas para WSN. Se hace hincapié en su rango, consumo energético, modulación y disponibilidad geográfica. Es necesario tener en cuenta estos datos a la hora de seleccionar la tecnología para el desarrollo de los nodos y el despliegue de la red. Estos datos indican que para un despliegue en zonas donde se requiera un rango de varios kilómetros de longitud o cobertura en áreas remotas, no se pueden aplicar tecnologías como WiFi ah o tecnologías de telefonía celular. Otro factor crucial es el consumo energético producido por los nodos tanto cuando están en reposo como cuando se encuentran activos, transfiriendo y recibiendo datos de la red. Incrementar la autonomía energética del dispositivo es un objetivo fundamental en aplicaciones en las que los mantenimientos de los nodos pueden llegar a darse cada 6 meses o periodos más largos, como ocurre en los *Smart Waste Collection Systems* en el



momento de la limpieza y reparación de los contenedores. Considerando estos factores, tres opciones de las presentadas anteriormente se postulan como idóneas para el sistema propuesto: Sigfox, LoRa and NB-IoT.

**Tabla 1** Comparación de tecnologías para WSN

Nombre	Rango (km)	Consumo en espera	Consumo en transmisión	Modulación	Disponibilidad
RPMA	10-100	0,5 $\mu$ A	85 mA	RPMA+DSS S	Zonas específicas
Weightless P	2	0,7 $\mu$ A	<70 mA	GMSK+QPSK	Mundial
ZigBee	0.100	3 $\mu$ A	30 mA	BPSK	Mundial
LoRa	5-15	0,5 $\mu$ A	<90 mA	LoRa	Mundial
Sigfox	3-10	0,5 $\mu$ A	<70 mA	BPSK	Mundial
Celular	2-5	10 mA	800 mA	8PSK	Mundial
WiFi ah	<1	-	<100 mA	QPSK/256QAM	-
NB-IoT	2-5	5 $\mu$ A	<100 mA	QPSK	Zonas específicas

La disponibilidad limitada de NB-IoT junto al elevado precio que presentan sus módulos hace que esta tecnología sea difícil de implantar actualmente en un entorno de producción real. Por otro lado, Sigfox es una tecnología que solo se encuentra disponible bajo un esquema de suscripción, lo que significa un coste adicional en una posible solución final. Por tanto, la tecnología que se plantea más indicada para el sistema propuesto tanto por características técnicas como comerciales es LoRa.

Una vez seleccionada esta tecnología, es necesario seleccionar el protocolo de comunicación que trabajará ella. Las alternativas actuales en la literatura [Ayoub et al. 2018] son principalmente DASH7[Haystack], Symphony [Symphony Link] y LoRaWAN. En este trabajo se ha optado por la utilización del protocolo LoRaWAN por su carácter abierto y por haber sido utilizada previamente en la literatura en trabajos como “*An experimental performance evaluation of LoRaWAN over a real environment in Bangkok*” por Vatcharatiansakul et al. [Vatcharatiansakul et al. 2017] y “*A long range wide area network-based smart pest monitoring system*” por Yu et al. [Yu et al. 2017], entre muchos otros [Catherwood et al.]

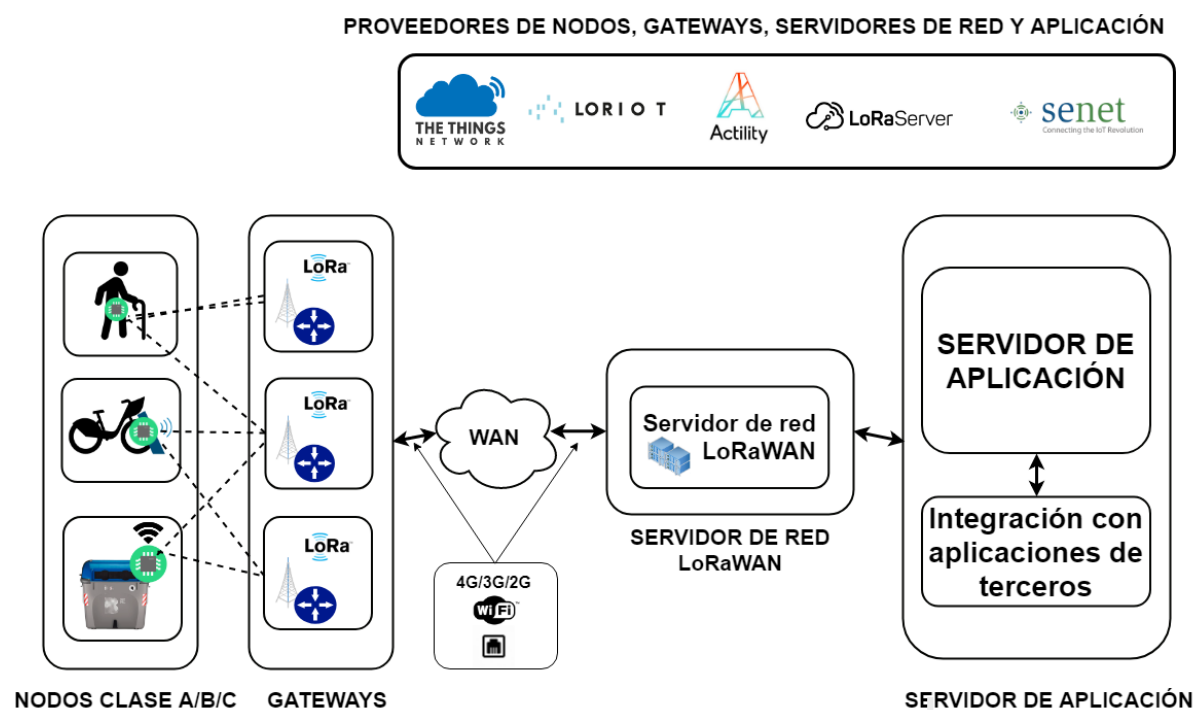
LoRaWAN es un protocolo de control de acceso al medio (*Media Access Control*, MAC) construido sobre una capa física basada en LoRa y define el protocolo de comunicación y la arquitectura que presenta la red inalámbrica de sensores. Está indicado para redes de sensores donde los nodos intercambian paquetes con el servidor de aplicación periódicamente con grandes intervalos de tiempo (horas o días) y con tasas de datos muy bajas.

El protocolo y la arquitectura de red tienen una influencia directa en la cantidad de batería consumida por los nodos finales, la capacidad de la red, la calidad del servicio, la seguridad y las aplicaciones finales que pueden utilizar la red. Este protocolo es desarrollado y mantenido por LoRa Alliance [Lora Alliance 2017].

Los componentes mínimos necesarios para formar una red, especificados en la versión 1.0 del protocolo LoRaWAN, son: nodos o dispositivos finales, pasarelas o *gateways* (estaciones base) y servidores de red:

- **Nodo:** dispositivo de bajo consumo que se comunica con los *gateways* empleando un enlace inalámbrico LoRa.
- **Gateway:** dispositivo intermedio responsable de redirigir los paquetes procedentes de los nodos a los servidores de red a través de una interfaz de red de retorno o *backhaul* IP como Ethernet WiFi o 3G/4G, permitiendo un mayor rendimiento. En un despliegue de una red LoRaWAN pueden existir múltiples *gateways* y estos pueden recibir un mismo paquete de un mismo nodo.
- **Servidor de red:** dispositivo responsable de decodificar la información de los paquetes, identificar paquetes duplicados enviados por los nodos y generar los paquetes de respuesta que deben ser enviados a los nodos.

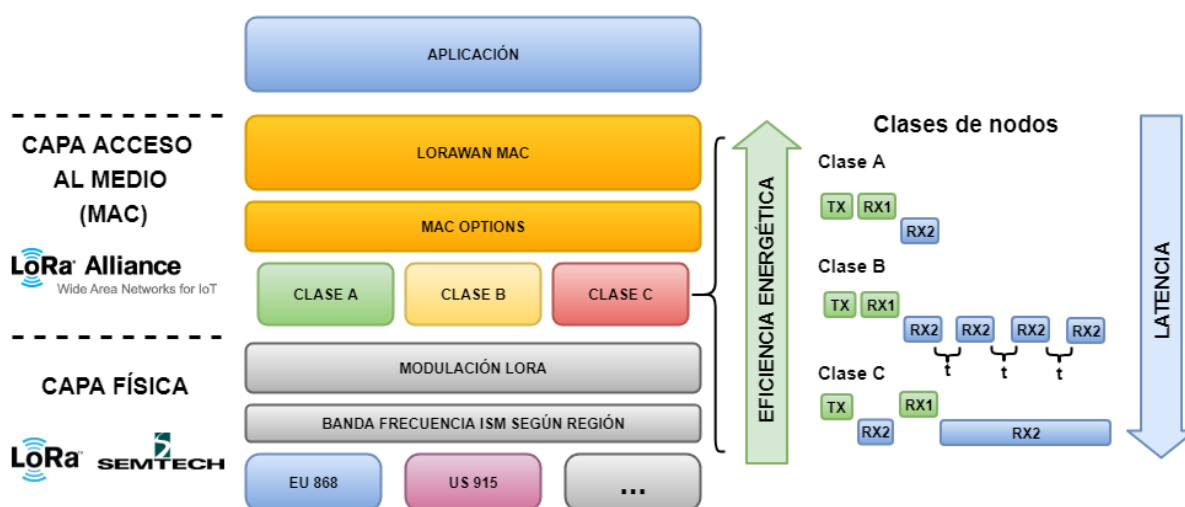
A diferencia de las tradicionales redes de telefonía celular, los nodos no están vinculados con un *gateway* en particular para acceder a la red. Los *gateways* se emplean sencillamente como un enlace encargado de retransmitir los paquetes recibidos de los nodos a los servidores de red, incluyendo una serie de datos relativos a la calidad de la recepción. Por tanto, un nodo se asocia con un servidor de red, que es el encargado de detectar paquetes duplicados, escogiendo el *gateway* apropiado para mandar una respuesta si esta es necesaria. Los *gateways* actúan de manera transparente para los nodos finales. En la **Figura 4** se muestra un esquema arquitectónico de los elementos mencionados anteriormente correspondiente al protocolo LoRaWAN [Petäjärvi et al. 2017].



**Figura 4** Diagrama básico de la arquitectura LoRaWAN

Como se puede apreciar en esta figura, estas redes presentan una topología de “estrella de estrellas” donde los nodos se comunican con un gateway y estos retransmiten la información a un servidor de red. A su vez estos servidores de red se comunican con los denominados servidores de aplicación que permiten la interacción de aplicaciones de terceros con la red de sensores.

En la **Figura 5** se muestra un esquema de la pila del protocolo LoRaWAN, en el que se puede observar cada una de las capas que posee este protocolo [Augustin et al. 2016b] y su correspondencia con las capas 1 (capa física) y 2 (capa de enlace) del modelo OSI. Como se ha mencionado anteriormente, se emplea LoRa en la capa física que opera en bandas de frecuencia ISM de distintas frecuencias, dependiendo de la región geográfica. El *payload* de cada transmisión oscila entre 2 y 255 bytes y la tasa de transferencia de datos puede alcanzar hasta los 50 Kbps. Esto implica que la información intercambiada por los nodos será de reducido tamaño y con conexiones muy poco frecuentes. La técnica de modulación es una tecnología patentada por Semtech [Semtech 2018]. En la misma figura, se puede ver que el protocolo LoRaWAN especifica cómo deben usar los nodos y *gateways* la capa física ya que se trata de un protocolo de control de acceso al medio.



**Figura 5** Pila del protocolo LoRaWAN

El protocolo LoRaWAN especifica tres clases de nodos para dar respuesta a las distintas necesidades de las aplicaciones finales:

- **Clase A:** cuentan con una comunicación bidireccional en la que los nodos pueden programar una transmisión (*uplink*) basándose en sus necesidades. Cada transmisión (*uplink* TX en la **Figura 5**) se ve seguida de dos cortas ventanas de recepción de datos (*downlink*, RX1 y RX2 en la **Figura 5**): RX1 en la misma frecuencia de emisión y RX2 en una frecuencia preconfigurada para la recepción. Transmisiones desde el servidor al nodo receptor deberán esperar hasta que se efectúe la próxima transmisión del nodo. Este tipo de nodos son los que tienen un consumo energético menor, pero también son lo que ofrecen una rigidez mayor en la comunicación.
- **Clase B:** al igual que la clase anterior se trata de una comunicación bidireccional, pero en este caso con ventanas de recepción programadas. Este tipo de nodos requiere de un paquete (*beacon*) enviado por los gateways periódicamente de tal forma que los servidores de red son capaces de detectar si el dispositivo está en una de las ventanas de recepción programadas.
- **Clase C:** cuentan con una comunicación bidireccional al igual que las anteriores pero esta vez el dispositivo está continuamente disponible para recibir datos. Este último tipo es el que más energía consume y está especialmente indicado para sensores con una conexión eléctrica.

Cabe destacar que la especificación LoRaWAN no permite comunicaciones nodo a nodo (*peer to peer* P2P), es decir, los paquetes solo pueden ser transmitidos desde el nodo al servidor de red y viceversa. Adicionalmente, la especificación indica que las redes deben usar bandas de frecuencia ISM lo que conlleva una serie de limitaciones relativas al máximo número de transmisiones, potencia de emisión y tiempo de duración de las transmisiones o ciclo de trabajo

(*duty cycle*). Esto se traduce en retardos en la transmisión de paquetes por un nodo, por ejemplo, si la limitación es de un 1 %, es necesario esperar 99 veces la duración del último paquete enviado antes de poder volver a enviar en el mismo canal.

A la decisión de emplear LoRaWAN como protocolo para la comunicación, está ligada la selección de un servidor de red LoRaWAN y todo el software relacionado. Existen distintas opciones a la hora de desplegar una arquitectura LoRaWAN que pueden ser resumidas en tres:

- (1) **Soluciones gratuitas y de código abierto:** como LoraServer.io [LoraServerIO 2018] que provee el software necesario para implementar las características necesarias para cumplir la especificación LoRaWAN, concretamente, el software incluido en los gateways, conocido como *gateway bridge* o *packet forwarder* (“redireccionador de paquetes”), el software que implementa el servidor de red LoRaWAN y, por último, el servidor de aplicación que permite la administración e integración de la red de sensores con terceras aplicaciones. Tanto el mantenimiento como el hosting de este software recae en el usuario de esta opción.
- (2) **Solución gratuita y colaborativa:** como *The Thing Network* (TTN) [The Things Network 2016] que, al igual que la opción anterior, proporciona una opción de hosting propio y, además, proporciona otra opción que consiste en una plataforma gratuita y colaborativa, ofertada como “plataforma como servicio” (*Platform as a Service*). La plataforma es gratuita, pero cada gateway incluido en ella puede recibir datos de cualquier nodo perteneciente a cualquier aplicación de la plataforma. Así, los usuarios pueden crear de manera colaborativa una infraestructura que a cambio puede ser utilizada por cualquier persona en *The Things Network*. Adicionalmente ofrece integraciones con aplicaciones de terceros.
- (3) **Soluciones comerciales:** como por ejemplo Actility [Yang 2014], Lorient [LORIENT] and Senet [Senet 2018] proporcionan soluciones comerciales llave en mano para la infraestructura que van desde los nodos y gateways hasta el almacenamiento y análisis de datos en sus plataformas.

La **Tabla 2** muestra una comparativa entre algunas de las distintas opciones disponibles para la implementación de un servidor de red LoRaWAN y todo el software relacionado.

Una vez evaluadas las diferentes alternativas, se ha optado por la utilización de TTN [The Things Network 2016] en este trabajo como servidor de red LoRaWAN debido a su carácter colaborativo y su opción de hosting gratuito. TTN provee de servicios de integración con aplicaciones de terceros que facilitan la integración con sistemas externos. Además, gracias a la selección de esta opción, el sistema puede utilizar toda la infraestructura de la comunidad TTN, ampliando la posible cobertura de este más allá de los gateways desplegados para tal fin.

**Tabla 2** Opciones disponibles para un servidor de red LoRaWAN

Nombre	Hosting	Código abierto	Plan de precios
LoraServer.io	Propio	si	gratuito
TTN	Propio / Terceros	si	gratuito / pago
Actility	Terceros	no	gratuito (limitado) / pago
Loriot	Terceros	no	gratuito (limitado) / pago
Senet	Terceros	no	gratuito (limitado) / pago

De la misma forma, la infraestructura de gateways desplegada por este sistema sería útil para todo aquel usuario de la comunidad TTN, habilitando así el despliegue de otros proyectos relacionados con esta tecnología en el área donde se encuentra la red de sensores desplegada, lo que favorece el posible desarrollo tecnológico de la zona.

Finalmente, el sistema que se propondrá en este trabajo deberá presentar el mínimo acoplamiento al servidor de red elegido para que sea posible cambiar esta opción en el futuro si fuese necesario.

### 2.2.3 Aplicaciones de redes inalámbricas de sensores LPWAN

En esta sección se describen algunas de las principales aplicaciones de redes inalámbricas de sensores LPWAN en la literatura. Habitualmente estas aplicaciones comparten un mismo patrón de interacción, que puede ser alguno de los siguientes:

- **Detección de eventos:** tanto la detección y clasificación de eventos que desencadenen una acción posterior en el sistema que hace uso de la WSN.
- **Toma de datos periódica:** muchas de las aplicaciones consisten en la toma de datos de sensores de manera periódica con una frecuencia relativamente baja, dependiendo de las restricciones de la tecnología de red empleada o de los datos a recoger.
- **Seguimiento (tracking):** unido a la toma de datos periódica y a la detección de eventos se podrían incluir las funciones de seguimiento, si el nodo final dispone de un sistema de geoposicionamiento, simplemente, empleando la propia red para localizarlo. Esto dependerá de la precisión necesaria en la localización.

A continuación, se describen algunos casos de uso estudiados en la literatura con este tipo de redes inalámbricas de sensores entre muchos otros existentes.

- **Salud y monitorización:** el uso de redes LPWAN para la monitorización remota de la salud y bienestar tanto de pacientes como de personas de la tercera edad ha sido ampliamente estudiado en los últimos años [Petäjälä et al.]. Es posible encontrar

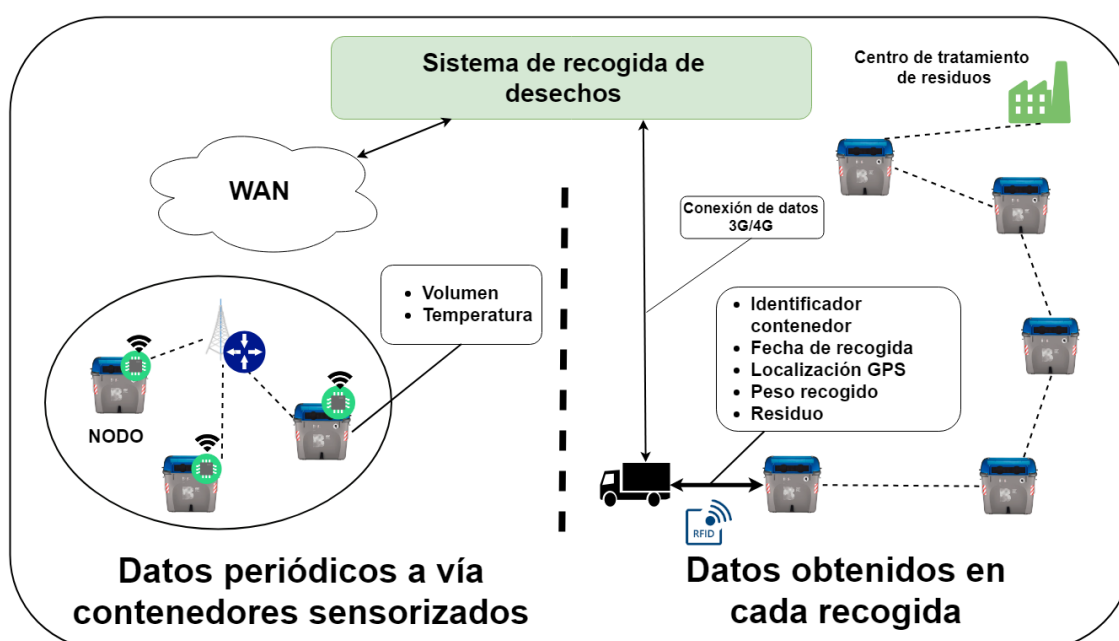
trabajos como el realizado por Chen et al. [Chen et al. 2017] en el que desarrollan un dispositivo destinado a detectar ataques al corazón y envía una alerta empleando LoRa.

- **Monitorización del ambiente:** multitud de trabajos como los citados en el de Pule et al. [Pule et al. 2017b] describen de qué manera emplear distintas redes inalámbricas de sensores LPWAN para la monitorización de la calidad del agua. Es posible encontrar trabajos como el de Sujuan Liu et al. [Sujuan Liu et al. 2016], centrados en la monitorización de la calidad del aire, o el trabajo de Park et al. [Park et al.] para la detección de fuegos en bosques.
- **Control del ganado y agricultura de precisión (*Smart Farming*):** la utilización de redes LPWAN en el ámbito de la ganadería y agricultura inteligente se está haciendo muy popular recientemente y es posible encontrar multitud de trabajos en la literatura [Villarrubia et al. 2017a] [Barriuso et al. 2018] relacionados con el control del ganado y la toma de datos relativos al suelo para la creación de sistemas inteligentes que, por ejemplo, detecten y notifiquen el inminente parto de una res o que controlen la cantidad de agua necesaria para un riego.
- **Sistemas de préstamo de bicicletas (*Bike Sharing Systems*):** en la literatura es posible encontrar redes de sensores aplicadas a la toma de datos de sistemas de préstamo de bicicletas en las ciudades (trabajos como el de [Lundh 2017]).
- **Sistemas de recogida de residuos:** al igual que en los casos anteriores, las redes inalámbricas de sensores se han empleado en la literatura para la toma de datos en este tipo de sistemas. En la siguiente sección se describe ampliamente la toma de datos en este tipo de sistemas.

#### 2.2.4 Toma de datos en sistemas de recogida de residuos

En la literatura encontramos diferentes formas de obtener información acerca del llenado de los contenedores en un sistema de recogida de residuos inteligente o *Smart Waste Collection Systems* [Hannan et al. 2015]. Sistemas como el propuesto por M.A Hannan et al. [Arebey et al. 2010] emplean tecnología RFID para la identificación de cada uno de los contenedores en el sistema. Los camiones de la flota de recogida incorporan un lector RFID, un GPS, un sistema de comunicación GPRS y, habitualmente, un sistema de pesaje incorporado en el propio camión [Chowdhury and Chowdhury 2007]. Durante la recogida de cada contenedor, el camión identifica el contenedor gracias al tag RFID, almacena su posición GPS actual, mide la cantidad de desechos que contiene y proporciona toda esta información a un servidor central. En algunos trabajos como el de Karadimas et al. [Karadimas et al. 2016] es el propio contenedor el que realiza la medida de su peso y ésta es suministrada al camión y, en otros, es el operario el que mediante un mando realiza la medida del peso bruto del contenedor gracias a una célula de carga que incorpora el gancho del camión y, posteriormente, otra medida con el contenedor vacío que proporciona la tara del contenedor. Este tipo de sistemas son más antiguos y se pueden encontrar ampliamente implantados en sistemas de recogida tanto rurales como

urbanos. Sus principales ventajas son su bajo coste de despliegue ya que el de la identificación RFID es mínimo y el principal coste corresponde a la flota de camiones y no a la flota de contenedores. La principal desventaja de estos sistemas es la imposibilidad de obtener medidas de los contenedores en tiempo real y de manera periódica, ya que solo es posible obtener las medidas cuando se está realizando la recogida. Esto posibilita la captura de datos históricos de recogida de desechos, pero no permite conocer con un nivel de detalle alto la demanda diaria y el llenado en el momento de realizar las rutas de recogida, imposibilitando la optimización con datos deterministas de ésta. En la parte derecha de la **Figura 6** se puede observar este enfoque de obtención de datos.



**Figura 6** Toma de datos en sistemas de recogida de desechos

Como respuesta a esta problemática, autores como Longhi et al. [Longhi et al. 2012] y Ramson et al. [Ramson and Moni 2017a], entre muchos otros, proponen el empleo de redes inalámbricas de sensores para la obtención de datos relativos a los contenedores (temperatura, volumen, peso, etc.) de forma remota (**Figura 6** parte izquierda). La proliferación de sensores con dimensiones más reducidas y protocolos de comunicación de bajo consumo hacen posible la integración de sensores en los contenedores que reporten estos datos tanto de forma periódica (medidas continuas de peso y volumen) como de manera eventual, notificando una alerta debido a un problema (exceso de temperatura). Este tipo de sistemas se comienzan a integrar en forma de proyectos piloto y cada vez más número de contenedores poseen esta sensorización, haciendo posible una optimización de rutas de recogida más eficiente. En este trabajo de tesis se pretende integrar estos dos tipos de obtención de datos en el sistema



propuesto para que posteriormente los datos puedan ser utilizados en la creación de modelos de predicción.

## 2.3 Modelado predictivo

Esta sección está dedicada a la revisión de algoritmos de *Machine Learning* para crear modelos predictivos que puedan ser empleados en la predicción de la demanda en distintos casos de estudio a partir de datos históricos.

A continuación, se detallan algunas de las técnicas disponibles en la literatura que pueden ser empleadas en el modelado predictivo y de las que se hará uso en la propuesta de este trabajo de tesis.

### 2.3.1 Random Forest Regressors

*Random Forest* o “bosque aleatorio” en castellano, es un tipo de técnica que tiene su origen en los árboles de decisión y que está formado por un conjunto de estos, de ahí parte de su nombre *Forest* (bosque). Esta técnica fue desarrollada por Leo Breiman en 2001 [Breiman 2001].

Se trata de un modelo que se encuentra dentro de la categoría de métodos conocidos como *ensemble methods* los cuales son un tipo de algoritmos de aprendizaje que emplean a su vez múltiples algoritmos de aprendizaje automático para conseguir mejores resultados en la predicción de los que se conseguirían empleando únicamente uno de esos algoritmos.

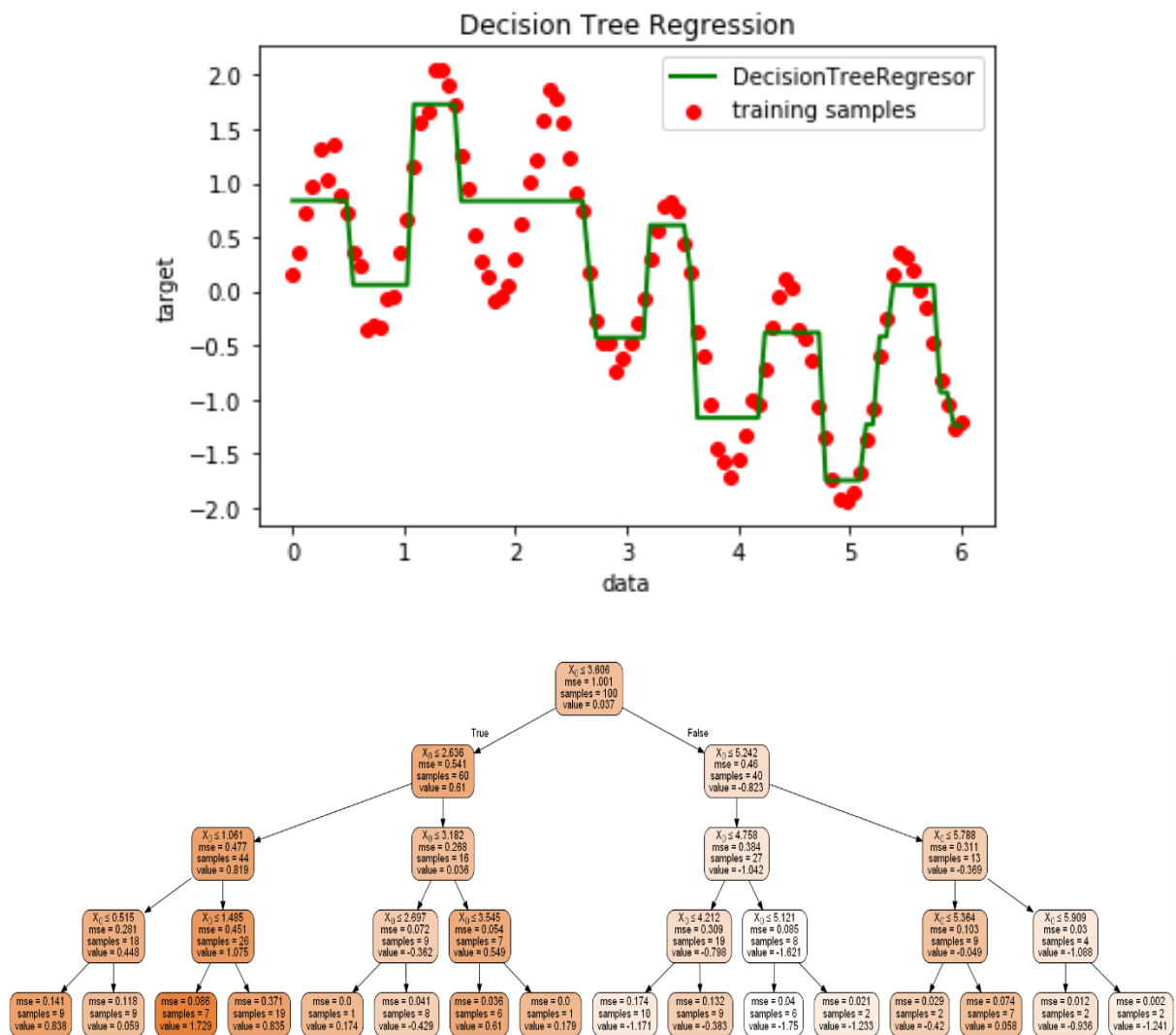
Esta técnica se fundamenta en los árboles de decisión, que pueden ser empleados tanto para tareas de clasificación como de regresión (árboles de clasificación o de regresión), por lo tanto, este tipo de *ensemble* puede ser empleado para cualquiera de esas tareas.

Un árbol de regresión consiste en un modelo predictivo que realiza una partición recursiva del espacio definido por las dimensiones del problema de regresión hasta que obtienen pequeñas regiones del espacio donde un modelo simple puede ser aplicado. Por tanto, este modelo tiene 2 partes fundamentales:

- Una partición recursiva del espacio formado por las dimensiones del problema en  $J$  regiones distintas y no superpuestas:  $R_1, R_2, \dots, R_J$
- Para cada muestra que se ubique en la región  $R_j$ , se realiza la misma predicción, que es simplemente la media de los valores de las muestras de entrenamiento localizadas en dicha región.

Las particiones se representan a través de una estructura de árbol binario en el que cada uno de los nodos terminales o hojas del árbol representan una celda de la partición. Una muestra  $X$

pertenecerá a una hoja si  $X$  se encuentra en la correspondiente celda. Para averiguar en qué celda será colocada dicha muestra es necesario comenzar por el nodo raíz del árbol y tomar una serie de decisiones en función de los valores de cada dimensión que tenga dicha muestra. Los nodos internos del árbol se encuentran etiquetados con una serie de condicionales que delimitan cada una de las subparticiones y, dependiendo del valor que posea la muestra, el proceso continuará por una rama o por otra hasta llegar a un nodo hoja. Una vez llegado a este punto, para los árboles de regresión clásicos se obtiene el valor predicho tras realizar la media del conjunto de muestras de entrenamiento que pertenecen a ese nodo.



**Figura 7** (Parte superior) Salida de un Árbol de regresión (Parte inferior) Árbol de regresión

En la parte superior de la **Figura 7** se puede observar el resultado de aplicar un árbol de regresión y, en la parte inferior, la estructura del árbol creado. Este ejemplo se ha generado a

partir de un conjunto de datos aleatorio empleando el *framework* Scikit Learn [Pedregosa et al. 2011].

El proceso de creación del árbol se realiza habitualmente en un enfoque *top-down*, esto quiere decir que comienza en el nodo raíz del árbol (donde todas las muestras pertenecen a una única región) y sucesivamente divide el espacio en dos partes, eligiendo el corte en cada paso que mejor divida el conjunto de muestras. Este procedimiento se conoce como *recursive binary splitting* y consiste en elegir el mejor corte de todos los posibles que minimice el coste de una métrica definida. Existen distintas métricas para evaluar cuál es el mejor “corte” como son *Gini impurity*, *Information Gain* y *Variance Reduction* (utilizando unos u otros, dependiendo de si el problema es de regresión o clasificación), empleados en algoritmos de árboles de decisión como (*Classification and Regression Trees* CART) [Breiman et al.] y C4.5 [Salzberg 1994]. Este proceso continúa recursivamente empleando, a su vez, las sucesivas subdivisiones del espacio producidas en cada corte. Además, este proceso debe conocer cuándo detenerse y para ello se han definido distintos criterios de parada en la literatura, como, por ejemplo, un número de muestras mínimo en cada hoja o la profundidad del árbol para reducir la complejidad del árbol creado.

Dado que los árboles de decisión presentan problemas de alta varianza y tienden a crear un modelo con sobreajuste (*overfitting*), es necesario introducir la técnica *Bootstrap Aggregation* o más conocida como *Bagging* que consiste en un procedimiento de propósito general para reducir la varianza de un método de aprendizaje automático y es frecuentemente empleada en el contexto de árboles de decisión.

Como se describe en [John Lu 2010], dado un conjunto de  $n$  observaciones independientes  $Z_1, Z_2, \dots, Z_n$ , cada una de ellas con una varianza  $\sigma^2$ , la varianza de  $\bar{Z}$  viene dada por  $\sigma^2/n$ , es decir, promediando un conjunto de observaciones se reduce la varianza. De hecho, una forma habitual de reducir la varianza e incrementar la precisión de un método de aprendizaje automático es tomar varios conjuntos de entrenamiento de una población, crear un modelo de predicción con cada uno de ellos y promediar las predicciones de todos ellos. Por lo tanto, es posible calcular  $\hat{f}^1(x), \hat{f}^2(x), \dots, \hat{f}^B(x)$  empleando  $B$  conjuntos de entrenamiento y promediarlos para obtener un único modelo de predicción con baja varianza, que presenta la siguiente forma ecuación (1):

$$\hat{f}_{average}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x) \quad (1)$$

Este enfoque no es práctico porque habitualmente no se tienen múltiples conjuntos de entrenamiento. En su lugar, es posible realizar un proceso de remuestreo conocido como *Bootstrapping*, tomando muestras repetidas de un único conjunto de entrenamiento. Se generan,

por tanto,  $B$  conjuntos de entrenamiento empleando *Bootstrapping* y se generan los modelos  $\hat{f}^b(x)$  empleando dichos conjuntos tal que (ecuación (2)):

$$\hat{f}_{bagging}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x) \quad (2)$$

Este proceso es el conocido como *Bagging* y es posible emplearlo en múltiples modelos de regresión, siendo especialmente utilizado en árboles de regresión. Se construyen  $B$  árboles de regresión usando  $B$  conjuntos de entrenamiento generados con *Bootstrapping*. Estos árboles se generan permitiéndoles crecer profundos, provocando una alta varianza y un sesgo bajo individualmente en cada árbol. Finalmente, se promedian todos consiguiendo de este modo reducir la varianza. Esta técnica ha demostrado proporcionar grandes resultados en precisión, utilizando cientos o miles de árboles en el mismo procedimiento.

El enfoque tomado por el algoritmo *Random Forest* es ligeramente distinto y proporciona una mejora frente a la utilización de, simplemente, *Bagging* que consiste en crear árboles que no estén correlacionados. Al igual que en *Bagging*, se generan múltiples árboles con conjuntos de entrenamiento empleando *Bootstrapping*, sin embargo, en este caso, a la hora de construir los árboles, en cada corte se toma únicamente una muestra aleatoria de  $m$  características como candidata para evaluar el mejor corte. Este se realiza únicamente teniendo en cuenta una única característica de las seleccionadas. Este proceso se realiza continuamente en cada corte y habitualmente se toma  $m \approx \sqrt{p}$ , siendo  $p$  el número de características. Este proceso se denomina, en ocasiones, *feature bagging*.

El principal motivo de seleccionar solo una parte aleatoria de características en cada corte es que, si existe una característica que tiene una gran relevancia, ésta siempre se encontrará en la parte superior del árbol, por lo tanto, todos los árboles generados a partir de los distintos conjuntos de entrenamiento serán similares y estarán muy correlacionados. Lamentablemente, si se realiza un promediado de múltiples árboles muy correlacionados, esto no deriva en una disminución de la varianza como sí ocurriría con unos que no lo estuviesen. *Random Forest* consigue que los árboles no estén correlacionados empleando este método y obteniendo mejoras en la reducción de la varianza y la precisión obtenida.

Este tipo de técnicas han sido ampliamente utilizadas en la literatura en ámbitos muy distintos obteniendo buenos resultados: trabajos en campos como ingeniería en la predicción de la demanda de agua [Herrera et al. 2010] o energía eléctrica [Dudek 2011], *e-commerce* en la predicción de la demanda de la compra de artículos [Joshi et al. 2018], ciencias medioambientales y geoespaciales [Nguyen et al. 2015], estudios financieros [Kumar and Thenmozhi 2014], medicina [Kane et al. 2014] etc.

Una extensa descripción de su funcionamiento se puede encontrar en trabajos como [John Lu 2010] [Biau and Scornet 2015] [Verikas et al. 2011].

### 2.3.2 ExtraTree Regressors

Esta técnica, propuesta por [Geurts et al. 2006], va un paso más allá en el proceso de inclusión de aleatoriedad y añade un paso adicional al proceso visto anteriormente en el algoritmo de *Random Forest*. En esta técnica, los árboles son entrenados de la misma forma que en el algoritmo *Random Forest* con la salvedad de que el proceso empleado para realizar el corte se hace aleatorio, es decir, en lugar de calcular la mejor combinación característica/corte, para cada característica que se esté teniendo en cuenta se selecciona un valor aleatorio para el corte. Este valor se selecciona de un rango empírico obtenido del conjunto de entrenamiento en cuestión.

La principal ventaja de esta técnica frente a *Random Forest* es que computacionalmente presenta un mejor rendimiento.

### 2.3.3 Gradient Boosting Regressor

Esta técnica está basada en la conocida como *Boosting*. Se trata de otro enfoque destinado a mejorar las predicciones realizadas por un predictor débil, es decir, esta técnica puede ser aplicada a cualquier técnica de aprendizaje automático (como, por ejemplo, un árbol de decisión) ya sea para problemas de regresión o de clasificación.

*Boosting* es un *ensemble method* al igual que *Bagging* pero en este caso los modelos de predicción no son creados independientemente, sino de manera secuencial. En esta técnica, subsecuentes modelos de predicción aprenden de los errores cometidos por los anteriores. De esta forma, las muestras del conjunto de entrenamiento tienen una probabilidad desigual de aparecer en subsecuentes modelos y aquellas con el mayor error aparecerán más a menudo, por tanto, las muestras no son elegidas basándose en *Bootstrapping*, como sucedía en *Bagging*, sino basadas en el error.

La idea principal es filtrar muestras del conjunto de entrenamiento, dejando aquellas que el modelo de predicción débil puede manejar y, posteriormente, crear nuevos modelos que se centren en el resto de las muestras que fueron difíciles de manejar por los modelos anteriores.

Uno de los primeros algoritmos que ha empleado este tipo de técnicas ha sido *Adaptive Boosting* o AdaBoost [Freund and Schapire 1997]. En él, los modelos de predicción débiles son árboles de decisión con una única subdivisión. Su funcionamiento se basa en asignar distintos pesos a las muestras de entrenamiento, asignando más a aquellas que son más difíciles de clasificar y menos a las que se han manejado bien, los nuevos modelos creados son añadidos

secuencialmente y focalizan su entrenamiento en patrones más difíciles. Las predicciones son realizadas por votación y ponderadas por el peso de cada una de sus precisiones.

Más adelante, Friedman et al. [Friedman and Friedman 2000a] desarrollaron el algoritmo conocido como *Gradient Boosting* que consiste en un *framework* estadístico donde se trata *Boosting* como un problema de optimización numérica en el que el objetivo es optimizar una función de coste del modelo añadiendo modelos de predicción débiles mediante un procedimiento como *Gradient Descent* o descenso del gradiente. Esta clase de algoritmos son descritos como modelos aditivos por etapas debido a que se añade un nuevo modelo de predicción débil en cada paso y los modelos previos existentes en el modelo permanecen inalterados. *Gradient Boosting* involucra 3 elementos básicos:

- Una función de coste que debe ser optimizada: dependiendo de la tarea (regresión o clasificación), la función puede variar (*Squared Error*, *LogLoss*, etc). La gran ventaja de este *framework* es que es lo suficientemente genérico como para ser usado con distintas funciones de coste.
- Un modelo de predicción débil para realizar las predicciones: habitualmente son empleados los árboles de decisión.
- Un modelo aditivo que permita añadir los modelos débiles con el objetivo de minimizar la función de coste: cada uno de los modelos son añadidos de uno en uno y los existentes en el modelo no varían. Cuando se añade un nuevo modelo, se realiza un proceso de descenso del gradiente con el fin de optimizar la función de coste. Habitualmente, el descenso del gradiente se emplea para minimizar parámetros como coeficientes en una ecuación de regresión o pesos en una red neuronal, después de calcular el coste, los pesos se actualizan para minimizar dicho coste. En este caso, en lugar de parámetros se tienen modelos de predicción débiles (en este caso, árboles de decisión). Después de calcular el coste, para realizar el proceso de descenso del gradiente es necesario añadir un árbol al modelo que reduzca el coste. Esto se realiza parametrizando el árbol y, por tanto, modificando los parámetros de dicho árbol y moviéndolo en la dirección correcta (reduciendo el coste residual). Este enfoque es llamado *Functional Gradient Descent* o descenso del gradiente con funciones. Este proceso de adición se detiene cuando un número determinado de árboles se han añadido o bien una vez el coste alcanza un nivel aceptable o no mejora frente a un conjunto de datos de validación.

A partir de este *framework* básico es posible introducir mejoras tanto en la creación de los árboles (limitando el número de árboles, su profundidad, número de nodos finales, etc.), el proceso de actualización de los pesos (actuando sobre el *learning rate*), la utilización de muestras aleatorias como se realiza en *Random Forest*, procesos de regularización, etc.

Este tipo de técnicas han sido empleadas ampliamente en la literatura en distintos trabajos como en la predicción de la demanda eléctrica [Nassif 2016], predicción de tiempos de viaje [Zhang

and Haghani 2015] y predicción de la de generación de residuos [Johnson et al. 2017], entre muchos otros.

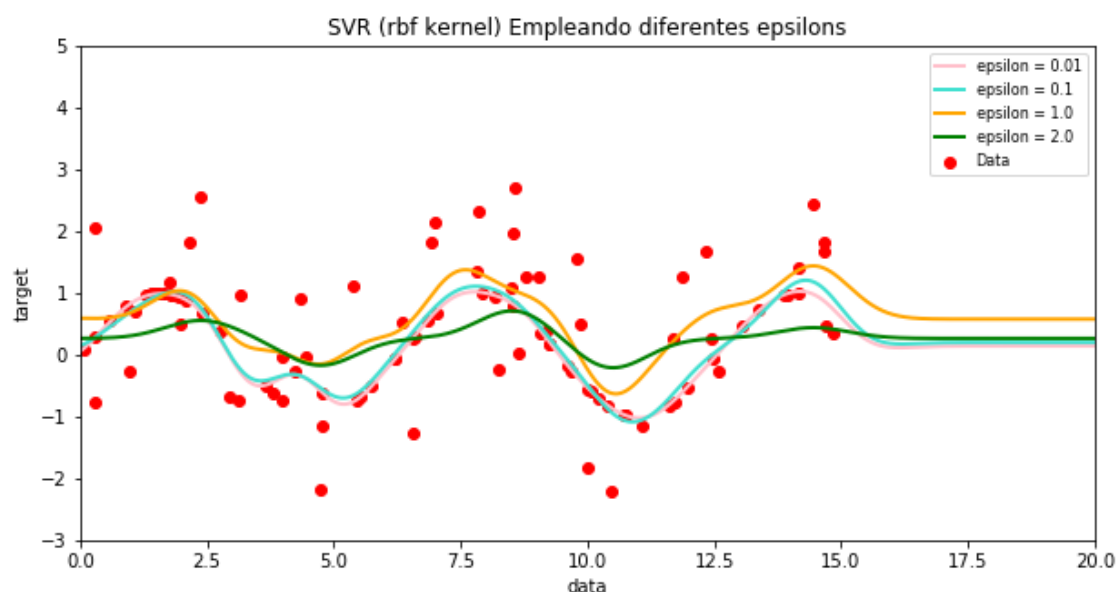
### 2.3.4 Support Vector Regressors

Este tipo de técnica es totalmente distinta a las vistas con anterioridad y está íntimamente relacionada con las conocidas como *Support Vector Machines*. De hecho, se trata de la aplicación de esta técnica a un problema de regresión.

*Support Vector Machine* (SVM) [Cortes and Vapnik 1995] o máquina de vector soporte en castellano, es un algoritmo de aprendizaje supervisado desarrollado por Vladimi Vapnik, empleado principalmente para tareas de clasificación. Su funcionamiento se basa en la construcción de un hiperplano (o conjunto de estos) en un espacio de alta dimensionalidad que separe las clases del conjunto de entrenamiento. Una buena separación de estas clases se consigue mediante el hiperplano que posee la mayor distancia a las muestras del conjunto de entrenamiento de cualquier clase ya que, si la distancia es mayor, el error de generalización del clasificador será menor. Por otra parte, en conjuntos de datos que no son linealmente separables en su espacio de dimensiones es necesario aplicar lo que se conoce como el truco del kernel o “*kernel trick*” [Boser et al. 1992] que permite “mapear” estos datos a un espacio de dimensionalidad mucho mayor donde si es posible realizar esta separación.

La técnica *Support Vector Regression* (SVR) [Drucker et al. 1997] emplea los mismos principios que SVM para el problema de clasificación con ciertas diferencias. Se trata de seleccionar el hiperplano regresor que mejor se ajuste al conjunto de datos de entrenamiento. En este caso no existen clases que se deban separar, sin embargo, la idea se basa en considerar un margen de tolerancia  $\varepsilon$ , de tal manera que se espera que todas las muestras del conjunto de entrenamiento se localicen en esa banda o “tubo” en torno al hiperplano que se desea calcular, es decir, que disten del hiperplano una distancia menor que  $\varepsilon$ . A la hora de definir el hiperplano solo se tendrán en cuenta las muestras del conjunto de entrenamiento que disten más de  $\varepsilon$  del hiperplano, que serán considerados los vectores soporte en este caso.

En la **Figura 8** se muestra un ejemplo de la aplicación de un modelo SVR a un conjunto de datos aleatorio con distintos valores para  $\varepsilon$ .



**Figura 8** Aplicación del algoritmo SVR a un conjunto de datos empleando distintos  $\epsilon$

Una amplia descripción de esta técnica se puede encontrar en el trabajo [Smola et al. 2004]

### 2.3.5 Redes neuronales artificiales

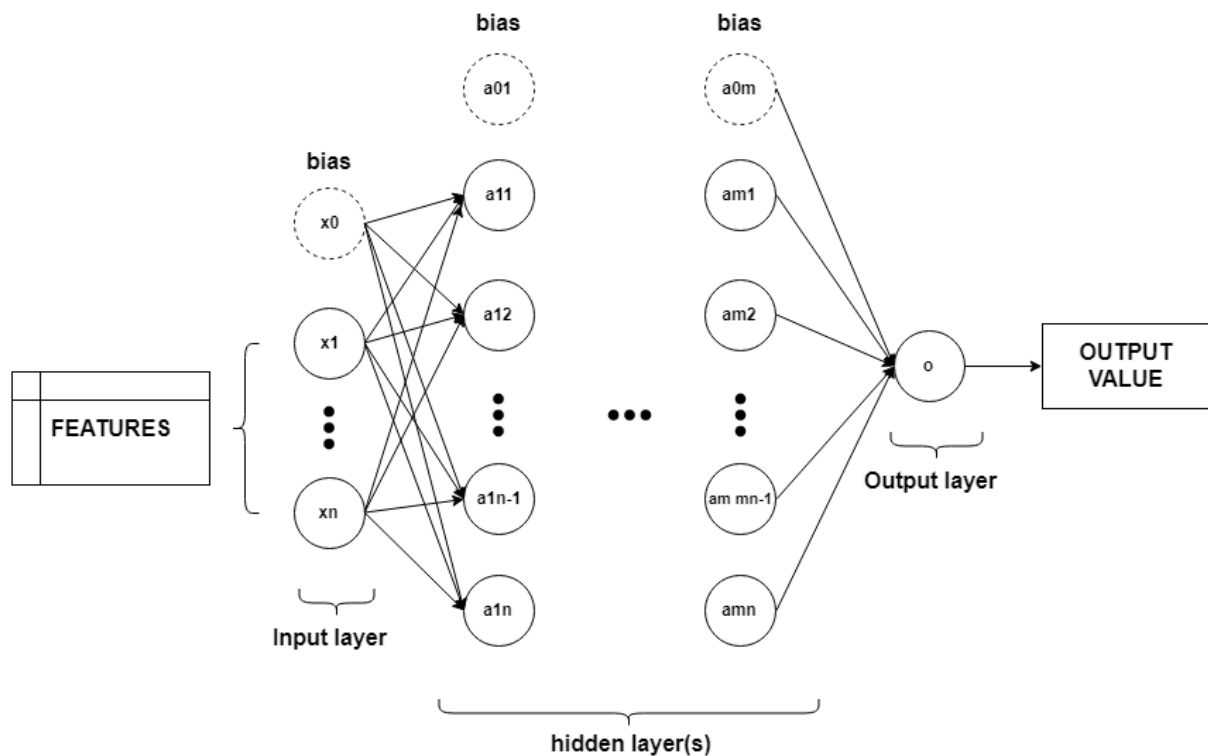
En el campo de las redes neuronales artificiales es posible encontrar desde el conocido como perceptrón, modelo más simple que consta de una sola neurona, hasta complejas estructuras con múltiples neuronas agrupadas en capas formando distintas arquitecturas.

Estas técnicas se emplean tanto para tareas de regresión como clasificación y su poder proviene de su habilidad para aprender la representación que presentan los datos de entrenamiento y como se relacionan dichos datos con su variable de salida. Según el teorema de aproximación universal [Kolmogorov 1957] [Cybenko 1989], una red neuronal con una sola capa oculta y el suficiente número de neuronas en ésta, es capaz de aproximar cualquier función no lineal  $f(\cdot): R^m \rightarrow R^o$  donde  $m$  es el número de dimensiones para la entrada y  $o$  el número de dimensiones para la salida.

Como arquitectura más simple empleada en la literatura es posible encontrar el perceptrón multicapa o (Multi Layer Perceptron MLP). En la **Figura 9** es posible ver una representación de este tipo de red neuronal. La capa situada a la izquierda, conocida como la capa de entrada, está formada por un conjunto de neuronas  $\{x_i | x_1, x_2, x_3, \dots, x_m\}$  que representan las características de entrada y  $x_0$  el término constante 1 conocido como bias. Cada neurona en cada capa oculta transforma los valores de la capa previa mediante una suma ponderada de la forma  $w_0x_0, w_1x_1, w_2x_2, \dots, w_mx_m$ , seguida por una función de activación no lineal  $g(\cdot): R \rightarrow$



$R$  (como por ejemplo sigmoide o tangente hiperbólica entre otras). Pueden existir múltiples capas ocultas hasta llegar a una capa final conocida como la capa de salida. Esta última capa produce los valores de salida de la red, por tanto, puede tener una o varias neuronas de salida. Este tipo de redes se conocen como redes alimentadas hacia adelante o redes *feedforward* debido a que las conexiones siempre son hacia la siguiente capa.



**Figura 9** Diagrama simple de la arquitectura de un perceptrón multicapa (MLP)

Existen otras arquitecturas de redes neuronales como Convolutional Neural Networks (CNN) o Recurrent Neural Networks (RNN), entre muchas otras, que incluyen distintos tipos de conexiones con propiedades orientadas a diferentes problemas. Este tipo de técnicas tienen un proceso de entrenamiento muy costoso, pero, en los últimos años, el campo conocido como Deep Learning, estas técnicas han tenido mucho auge debido al aumento en la capacidad de procesamiento y a las nuevas arquitecturas de redes neuronales con cada vez más capas (más profundas) y más complejas (arquitecturas con distintos tipos de conexiones y neuronas).

Estas técnicas han sido ampliamente utilizadas en la literatura en trabajos como el de [Shahabi et al. 2012], [Adamović et al. 2017] [Abbasi and El Hanandeh 2016] [Kannangara et al. 2018] para la predicción de desechos sólidos

## 2.4 Optimización de rutas de vehículos

En esta sección se aborda una revisión del estado del arte del problema de enrutamiento de vehículos o *Vehicle Routing Problem* VRP en inglés. La optimización de las rutas de recogida de desechos en los Smart Waste Collection Systems se plantea como un problema de este tipo, donde el principal objetivo es minimizar los costes de recogida buscando las rutas con menor coste para la flota de vehículos disponibles y contenedores que se deben recoger con sus respectivas demandas (cantidad de desechos).

A lo largo de esta sección se presentan la definición formal del problema VRP, los métodos empleados para su resolución, las distintas variantes que existen sobre el problema clásico y, dentro de estas, las que contemplan la incertidumbre en el problema. Por último, se detallarán los enfoques de resolución de problemas VRP que incorporan incertidumbre, como la optimización estocástica, robusta u otras alternativas como Simheuristics.

### 2.4.1 Definición de VRP

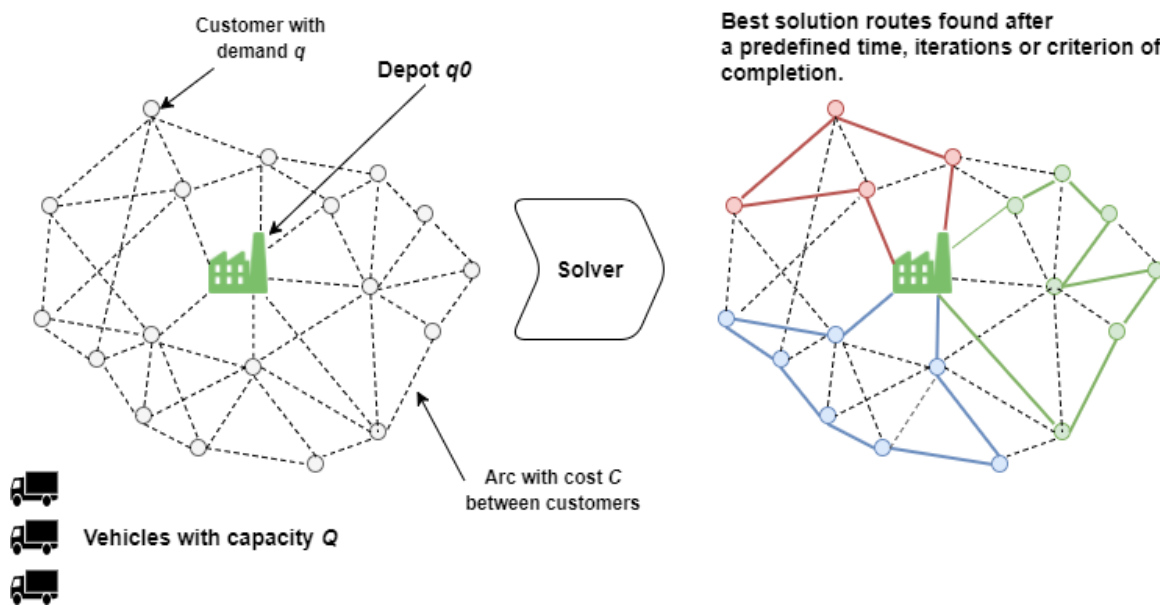
El problema de enrutamiento de vehículos es conocido en la literatura como VRP por sus siglas en inglés (*Vehicle Routing Problem*) [Dantzig and Ramser 1959] [Braekers et al. 2016]. Este problema presenta multitud de aplicaciones como son las rutas para empresas de mensajería, transporte de personas, suministro de mercancías, etc. Se trata de un problema con muchas aplicaciones prácticas pero con una alta complejidad [Lenstra and Kan 1981], ya que es un problema *NP-Hard*. Esto se debe a que el espacio de soluciones crece exponencialmente con el incremento del número de nodos y un algoritmo exacto no puede hallar una solución óptima para instancias grandes en un tiempo de cómputo finito razonable. Debido a este motivo ha proliferado el desarrollo de métodos de resolución heurísticos o aproximados y metaheurísticos

El nombre de VRP se emplea para toda una categoría de problemas dirigidos a encontrar un número de rutas para un conjunto de vehículos localizados en uno de los nodos denominado *depot*. Estos vehículos deben satisfacer la demanda de un conjunto de nodos distribuidos geográficamente y el objetivo principal es encontrar las rutas que satisfagan esas demandas a la vez que minimicen el coste total de estas rutas. Todas ellas deben empezar y acabar en el mismo *depot*. La versión más clásica del problema VRP es el conocido como CVRP (*Capacitated Vehicle Routing Problem*) en el que existen restricciones de capacidad para los vehículos de la flota, es decir, toda la flota posee una capacidad uniforme y existe un único *depot*.

La **Figura 10** muestra los elementos básicos de un CVRP clásico con una flota homogénea. Este problema es formalmente definido como:

Dado  $G(V, A)$  un grafo no dirigido donde  $V$  indica un conjunto de  $n + 1$  vértices.  $A$  es un conjunto de arcos definidos como  $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ .

El conjunto  $V = \{0, 1, 2, \dots, n\}$  define el número de clientes (en este caso contenedores). Cada arco  $(v_i, v_j)$  del conjunto  $A$  tiene un coste no negativo asociado:  $C_{ij}$ . Cada cliente del conjunto  $V$  tiene una demanda (en este caso, la cantidad de residuos en cada contenedor)  $q_i$  que debe ser recogida y transportada al *depot*  $q_0$ . La demanda del *depot* es siempre 0,  $q_0 = 0$ . El conjunto  $m$  corresponde a la flota de vehículos con la misma capacidad  $Q$  (si la flota tuviese capacidades distintas, correspondería a otro tipo de VRP) que puede ser empleada para la recogida, en este caso, de residuos de los  $n$  clientes (contenedores). Los  $m$  vehículos deben comenzar y terminar sus rutas en el *depot*. Una ruta es definida como un ciclo de bajo coste del grafo  $G$  que pasa por el *depot* y la demanda total del conjunto de vértices no excede la capacidad total del vehículo  $Q$ . El objetivo principal es reducir la distancia, tiempo o coste de la flota de  $m$  vehículos a la vez que se satisfacen todas las restricciones: (1) el *depot* es el punto inicial y final de cada ruta, (2) cada cliente es visitado una única vez por un solo vehículo y (3) la demanda total de cada ruta no excede la capacidad del camión  $Q$ .



**Figura 10** Representación gráfica de los parámetros de entrada de un problema CVRP clásico y una posible solución.

El problema CVRP es formulado como:

$$\min \sum c_{ij}x_{ij} \tag{3}$$

Con  $(i, j) \in V, V = \{0, 1, 2, 3, \dots, n\}, x_{ij} \in \{0, 1\}$  sujeto a:

$$\sum x_{0j} = m \quad (4)$$

$$\sum x_{i0} = m \quad (5)$$

$$\sum_{j=1} x_{ij} = 1 \quad (i = 1, \dots, n) \quad (6)$$

$$\sum_{j=1} x_{ij} = 1 \quad (j = 1, \dots, n) \quad (7)$$

$$m \geq 1 \quad (8)$$

La función objetivo que se busca minimizar es la ecuación (3), se trata del coste total de la solución. Las ecuaciones (4) y (5) hacen referencia a las restricciones que hace que  $m$  sea el número total de vehículos empleados para las soluciones y que todos ellos deben volver al *depot*. Las restricciones (6) y (7) hacen que cada cliente sea un nodo intermedio de la ruta y la restricción (8), que la flota de vehículos empleada sea mayor o igual que 1.

Esta definición formal corresponde a la versión más básica de un CVRP ya que, dependiendo de las restricciones del problema, este pertenecerá a otra categoría dentro de las múltiples variantes del mismo que se revisarán en la siguiente sección.

### 2.4.2 Variantes del problema VRP

A continuación, se detallan algunas de las múltiples variantes existentes en la literatura del problema CVRP; los autores Toth y Vigo hacen una revisión más detallada y extensa en su libro [Vigo and Toth 2014]. En este apartado se persigue revisar de forma general las distintas tipologías para poder enmarcar la tipología concreta que se tratará en este trabajo.

Como principales tipos de problemas VRP es posible encontrar:

- **Multiple Depot VRP (MDVRP):** se trata de una extensión del VRP clásico en el que la diferencia radica en la posibilidad de incorporar varios *depots*. Wren y Holliday [Wren and Holliday 1972] describieron por primera vez este problema.

- ***Periodic VRP (PVRP)***: se refiere a un caso de problema VRP en el que se diseña un número de rutas de reparto dado un periodo de tiempo, al contrario que en el resto de los casos, en los que se planifican las rutas para un único día. Trabajos como los de Christofides et al. [Christofides and Beasley 1984] o Francis et al. [Francis et al. 2008] tratan en profundidad este problema.
- ***Split Delivery VRP (SDVRP)***: se trata de una variante del problema VRP en el que es posible dividir las demandas de los clientes en otras más pequeñas. En este problema la demanda de los clientes podría ser mayor que la capacidad de los vehículos, por esta razón, los clientes podrían ser visitados más de una vez por distintos vehículos. Este tipo de problemas ha sido estudiado inicialmente por Dror et al. [Dror and Trudeau 1989] y más adelante por Archetti et al. [Archetti and Speranza 2008]
- ***VRP with Backhauls (VRPB)***: este tipo de problema VRP trata con dos tipos de clientes. Los conocidos como clientes *linehaul*, a los que se les debe entregar un determinado bien que se porta desde el *depot*, y los clientes *backhaul*, en los que se aplica el proceso opuesto, los bienes son recogidos y llevados hasta el *depot*. En este problema primero son entregados todos los bienes a los clientes y, posteriormente, se realizan las recogidas. Este tipo de problemas son ampliamente estudiados en el trabajo de Parragh et al. [Parragh et al. 2008].
- ***VRP with Pick-Up and Delivering (VRPPD)***: esta variante del problema difiere de la anterior en que los clientes pueden devolver bienes en cualquier momento y se debe contemplar que exista espacio suficiente en los camiones para llevarlos. Autores como Gribkovskaia et al. [Gribkovskaia et al. 2008] han trabajado en soluciones para este tipo de problemas.
- ***VRP with Time Windows (VRPTW)***: consiste en una de las variantes más utilizadas en la literatura. En esta variante, Solomon en 1987 [Solomon 1987] introdujo una ventana de tiempo en la que cada uno de los clientes debía ser visitado. Por lo tanto, los vehículos deben atender la demanda de cada cliente dentro de su ventana de tiempo. En esta variante se busca minimizar la flota de vehículos, el tiempo de viaje y el tiempo de espera. Este último tiempo hace referencia a la situación en la que un vehículo llega a un cliente antes del límite inferior de su ventana de tiempo, lo que causa una espera y un tiempo extra en toda la ruta. Sin embargo, se consideran soluciones no factibles aquellas en las que los vehículos lleguen después del límite superior de la ventana de un cliente. Muchos autores han estudiado este tipo de problemas y sus variantes como Taillard et al. [Taillard et al. 1997], Liberatore et al. [Liberatore et al. 2011], Hashimoto et al. [Hashimoto et al. 2010], entre otros.

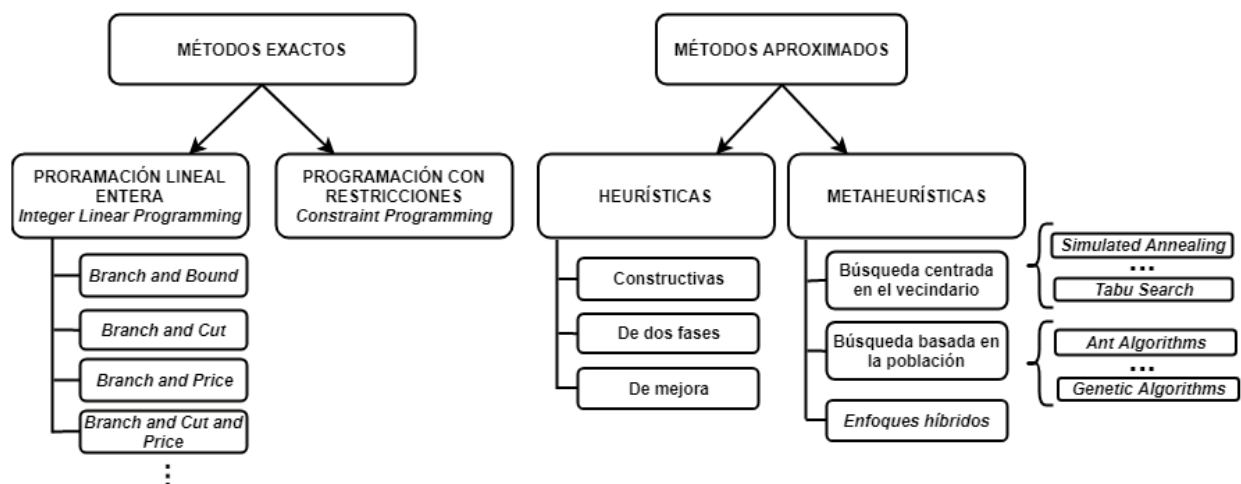
- ***Heterogeneous Fleet VRP (HFVRP)***: esta variante del problema incluye vehículos con distintas capacidades y con distintos costes asociados. Este problema ha sido ampliamente estudiado en la industria ya que muchas de las flotas de vehículos no son totalmente homogéneas. Asimismo, dentro de este tipo de problemas se pueden encontrar los que consideran una flota de vehículos ilimitada y los que, por el contrario, tienen una flota de vehículos limitada.
- ***Stochastic VRP (SVRP)***: contempla la inclusión de la incertidumbre en distintas partes del problema como es en la existencia o no de clientes, en la cantidad de demanda de los clientes o los costes asociados a los arcos (tiempo de viaje). Esta variante del problema será estudiada en profundidad en la sección 2.4.4 y es la que se empleará en el sistema propuesto del presente trabajo.

### 2.4.3 Métodos de resolución

Como se ha visto anteriormente existen multitud de tipologías dentro de la familia de los problemas VRP. Dependiendo de sus variantes, la formulación de cada uno de los problemas puede variar y también sus métodos de resolución. En esta sección se describen los enfoques seguidos en la literatura de manera general para encontrar soluciones a estos problemas.

Los métodos de resolución cambian en función de si es el objetivo es encontrar una solución óptima al problema o bien una solución subóptima en un tiempo razonable. Por lo tanto, es posible realizar una primera división entre los métodos de solución conocidos como exactos, que garantizan encontrar una solución óptima al problema realizando una exploración intensiva del espacio de búsqueda, y los métodos aproximados, en los cuales se busca una solución suficientemente buena en un tiempo de cómputo razonable. Los métodos aproximados son los que se usan principalmente en la literatura ya que, debido a la complejidad que presentan estos problemas en instancias con un número ligeramente elevado de clientes, los métodos exactos ya no son aplicables. Dentro de los métodos aproximados se encuentran los métodos heurísticos que suelen obtener una buena solución, pero no garantizan que sea la óptima ya que pueden detenerse en mínimos locales. Por otro lado, se encuentran los métodos metaheurísticos que son una serie de técnicas que no dependen del problema concreto y que permiten explorar el espacio de búsqueda para encontrar soluciones cercanas a la óptima de forma iterativa, permitiendo guiar el proceso de búsqueda de otras heurísticas.

A lo largo de esta sección se presentan los métodos de resolución mostrados en la **Figura 11** empleados en la literatura para solventar problemas VRP.



**Figura 11** Diagrama general de los métodos de resolución para problemas VRP.

### 2.4.3.1 Métodos exactos

Estos métodos optan por calcular cada solución en el espacio de búsqueda hasta que se encuentra la óptima. Gracias a estos métodos, existen en la literatura múltiples instancias de problemas VRP de las cuales se han encontrado sus soluciones óptimas y cuyos enunciados se emplean para poder comparar nuevos métodos de resolución. En este tipo de métodos destacan dos enfoques: Programación Lineal Entera (*Integer Linear Programming* - ILP) y Programación con Restricciones (*Constraint Programming* - CP).

Los problemas VRP pueden ser planteados como un problema de programación lineal entera [Dantzig 1963][Nobert 1987], es decir, parte (en este caso programación lineal entera mixta) o todas las variables del problema deben ser enteros, definiendo la función objetivo y las restricciones como conjuntos de desigualdades lineales. Se presentan a continuación diferentes estrategias seguidas en la literatura sobre este enfoque.

#### 2.4.3.1.1 Branch and Bound

Traducido al español como Ramificación y Acotación, es uno de los algoritmos más conocidos para solventar un problema VRP [Fisher 1994]. Este método realiza particiones del problema completo obteniendo subproblemas y reduciendo el espacio de soluciones, utilizando la estrategia “divide y vencerás”. En este algoritmo se explora el espacio de soluciones  $S$ , posteriormente, en el proceso de acotación se realiza una relajación del problema (un problema relajado es aquel donde las variables enteras se les permite tomar valores reales) por tanto, se admiten soluciones que no forman parte del conjunto de soluciones factibles. Seguidamente, se obtiene un límite inferior del valor de la solución óptima del problema relajado. Si el valor del problema relajado es mejor o igual que el espacio de soluciones, entonces, se termina, si no, se continúa la ramificación para continuar el algoritmo y seleccionar uno de los posibles subproblemas candidatos. El objetivo es descartar todos los candidatos no prometedores, basándose en el límite superior en inferior hasta que se obtenga el óptimo.

#### 2.4.3.1.2 Branch and Cut

Esta clase de métodos exactos se pueden encontrar en muchos de los problemas de optimización combinatoria. Este enfoque resulta de la combinación de métodos de planos de corte y el algoritmo anterior *Branch and Bound*.

Los autores [Padberg and Rinaldi] propusieron una mejora del algoritmo *Branch and Bound*, integrando esta técnica de corte de planos que dio origen a estos algoritmos. Este tipo de algoritmos proponen añadir nuevos cortes o desigualdades válidas como soluciones del problema. Más adelante, estos mismos autores plantearon emplear este algoritmo para la



resolución de un problema STSP (*Symmetric Travel Salesman Problem*) [Padberg and Rinaldi 1991]. Estos métodos suponen una gran mejora desde el punto de vista computacional.

### 2.4.3.1.3 Branch and Price

El procedimiento del algoritmo Branch and Price [Barnhart et al. 1998][Gutiérrez-Jarpa et al. 2010] está considerado como una generalización del algoritmo *Branch and Bound*. Se combina el algoritmo *Branch and Bound* con un algoritmo de generación de columnas. La idea principal de este es similar a *Branch and Cut* pero la diferencia reside en la generación de columnas en lugar de filas.

### 2.4.3.1.4 Branch, Cut and Price

Se trata de otro método exacto basado en programación lineal y es una extensión del algoritmo *Branch and Bound*. En este tipo de algoritmos los cortes y las variables pueden ser generadas dinámicamente por todo el árbol de búsqueda para mejorar las relajaciones. El algoritmo Branch and Price puede ser transformado en uno del tipo Branch Cut and Price, realizando un procedimiento de separación de planos de corte. Trabajos recientes como el de [Contardo et al. 2014] han logrado avances en este tipo de algoritmos.

### 2.4.3.1.5 Programación con restricciones

A diferencia de los enfoques previamente mencionados, que emplean programación lineal entera, *Constraint Programming (CP)* o Programación con Restricciones en castellano, puede emplear restricciones simbólicas y/o globales para describir el problema, además de expresiones que usen operadores lógicos y aritméticos. Este paradigma puede emplear *Branch and Bound* como método para resolver problemas VRP.

Se trata de un paradigma para representar y resolver una larga variedad de problemas y constituye una técnica eficiente para acelerar el proceso de búsqueda. Por este motivo hay una fuerte motivación en el desarrollo de este tipo de *frameworks* [Rossi et al. 2006].

Los problemas son representados en términos de variables, dominios correspondientes a esas variables y restricciones. Las expresiones involucradas no son solamente representadas como lógicas o expresiones aritméticas, pueden ser también restricciones simbólicas complejas. Además, el espacio de búsqueda es explorado empleando métodos de propagación de restricciones. Si se puede deducir el valor de una variable, el mecanismo de propagación en el árbol de búsqueda difunde la información acerca de esa inferencia por todas las restricciones para reducir los dominios, eliminando las variables inconsistentes. Si se eliminan todos los valores posibles en el dominio de una variable restringida, entonces no puede existir una

solución en esa rama del árbol de búsqueda y debe ejecutarse un proceso de *backtracking*, dando otro set de valores distintos al dominio.

En el caso particular del problema VRP las principales restricciones son:

- Tiempo: para los casos de problemas que involucren ventanas de tiempo de los distintos clientes.
- Capacidad: restricciones de peso, volumen, etc. Uno de los métodos para modelar las restricciones de capacidad es propagar el espacio libre disponible en cada vehículo a lo largo de la ruta.
- Restricciones de la ruta: por ejemplo, el tiempo inicial de un cliente a lo largo de la ruta calculada.

La riqueza del lenguaje de este *framework* hace a CP un candidato idóneo para problemas VRP. Sin embargo, debido a la naturaleza del problema y a su dificultad para resolverlo, este enfoque no es viable para instancias del problema de más de 30 clientes.

### 2.4.3.2 Métodos aproximados

Como se ha visto anteriormente, los métodos de búsqueda completos no pueden proporcionar una solución en un periodo corto de tiempo para instancias de un tamaño mediano o grande, por lo que se han desarrollado métodos aproximados para poder sobreponerse a esta limitación.

Estas técnicas se basan en diferentes estrategias, empleadas para buscar buenas soluciones, en algunos casos están basadas en la experiencia de los creadores sobre cómo realizar una búsqueda más efectiva y, en otros, incluso, se inspiran en comportamientos que se dan en la naturaleza.

El principal inconveniente del uso de este tipo de técnicas es la falta de garantía de obtener una solución óptima para el problema que se pretende resolver, pero, gracias a estas técnicas, se han podido encontrar buenas soluciones a instancias con elevado número de clientes.

Estas técnicas se dividen principalmente en dos: heurísticas y metaheurísticas.

### 2.4.3.3 Heurísticas

Los métodos heurísticos clásicos producen soluciones de una buena calidad, incluso, si realizan una exploración limitada del espacio de búsqueda. Por esta razón, obtienen mejores tiempos de computación que los métodos exactos. Cuando el problema VRP a resolver es grande o es necesario encontrar una solución rápida, este tipo de métodos son los más indicados.

Dentro de las diferentes heurísticas clásicas encontramos: (1) heurísticas constructivas, (2) heurísticas de dos fases y (3) los métodos o heurísticas de mejora. A continuación, se describen cada uno de ellos.

#### 2.4.3.3.1 Heurísticas constructivas

En las heurísticas denominadas constructivas se crea una solución de forma progresiva. Estos métodos suelen partir de una solución parcial (un conjunto de rutas que no visita la totalidad de los clientes) que progresivamente crece en cada iteración del algoritmo. Como contraposición a esta técnica se encuentra La Búsqueda Local (*Local Search*) que toma una solución completa y paso a paso trata de mejorarla con movimientos locales.

Uno de los algoritmos más conocidos es el algoritmo de los ahorros de los autores Clarke y Wright 1964 [Clarke and Wright 1964]. Este consiste en crear  $n$  rutas factibles y a partir de estas se intenta unir una ruta que termine en un nodo  $i$  con otra que empiece en un nodo  $j$  mediante un arco  $(i,j)$  calculando el ahorro de cada posible movimiento. Una explicación detallada de este algoritmo se encuentra en el trabajo de [Toth and Vigo 2002].

Este tipo de algoritmos han tenido éxito gracias al incremento de la capacidad de cómputo que ha permitido reducir los tiempos de ejecución de esta clase de algoritmos.

#### 2.4.3.3.2 Heurísticas de dos fases

Las heurísticas de dos fases descomponen el problema en 2 subproblemas que se resuelven secuencialmente. Es posible dividir estas heurísticas en dos categorías: (1) asignar primero y enrutar después y (2) enrutar primero y asignar después.

- **Asignar primero y enrutar después:** se persigue realizar una partición del conjunto de clientes en distintos subconjuntos (un proceso de *clustering*) que formarán parte de la misma ruta en la solución final. Posteriormente, para cada uno de los subconjuntos o *clusters* se crea una ruta que visite todos sus clientes. Las restricciones de capacidad son evaluadas en la primera etapa, asegurando de esta forma que la demanda total de cada *cluster* no sobrepase la capacidad total del vehículo. De esta manera, el proceso de creación de rutas implica resolver un problema del viajante o TSP (*Travel Salesman Problem*) para cada *cluster*, que será resuelto empleando métodos exactos o aproximados, dependiendo de la magnitud del problema. Este tipo de algoritmos son muy próximos a la solución visual que daría un humano a la hora de planear la ruta y existen múltiples trabajos en la literatura sobre este tipo de técnicas: El algoritmo de Fisher y Jaikumar [Fisher and Jaikumar 1981], el algoritmo de barrido de [Gillett and Miller 1974] o el algoritmo de los pétalos de [Foster and Ryan 1976], entre otros.

- **Enrutar primero y asignar después:** estos métodos buscan, en una primera fase, calcular una única ruta que visite todos los clientes del problema, aplicando el algoritmo del viajante. Evidentemente, la ruta generada no respetará las limitaciones de capacidad, por lo que es necesario seccionar esta ruta en un número de sub-rutas factibles, empezando y terminando en el *depot*. Existen varios trabajos en la literatura que aplican este enfoque [Beasley 1983] [Newton and Thomas 1974][BODIN and BERMAN 1979].

### 2.4.3.3 Heurísticas de mejora (Búsqueda Local)

También conocidas como técnicas de Búsqueda Local o (*Local Search*). Este tipo de métodos trabajan sobre una solución factible e intentan determinar otra de menor coste a partir de esta, realizando una serie de movimientos en las rutas. En cada iteración, el algoritmo se mueve hacia una solución mejor que la anterior y el proceso global termina cuando no es posible encontrar un movimiento que haga que la solución mejore. Debido al proceso que siguen este tipo de técnicas, es habitual que obtengan soluciones correspondientes a óptimos locales. Para solventar este hecho, debe existir la posibilidad de que el algoritmo se mueva a soluciones peores para encontrar nuevas vías de avance hacia óptimos distintos. Lin-Kernighan [Lin and Kernighan 1973] proponen un algoritmo basado en este hecho, que realiza movimientos compuestos donde es posible que uno de sus movimientos simples no mejore la solución, pero que, sin embargo, en el cómputo global de los movimientos si consiga una mejor solución.

Dentro de esta categoría destacan los algoritmos de intercambio r-óptimos [Lin 1965]. Entre los distintos tipos existentes se encuentran los del tipo intra-ruta que realizan movimientos de arcos dentro de la misma ruta y entre los que destacan las heurísticas 2-opt, 3-opt y la heurística de Lin-Kernighan [Lin and Kernighan 1973]. Por otro lado, se encuentran los métodos extra-ruta, que realizan cambios de arcos entre dos o más rutas distintas, como por ejemplo la heurística 2-swap.

### 2.4.3.4 Metaheurísticas

Las metaheurísticas son procedimientos de mejora que parten de una solución previamente obtenida por una heurística. Su objetivo es explorar el espacio de búsqueda para obtener una solución óptima o subóptima, realizando un número finito de cambios; habitualmente estos métodos obtienen mejoras manipulando componentes básicos de la solución inicial.

El término metaheurística fue usado por primera vez por Fred Glover en 1986 [Glover 1986] para describir una larga clase de heurísticas y métodos que realizaban procesos de búsqueda después de encontrar una solución óptima local. Por este motivo, una posible definición de este término podría ser “heurísticas que guían a otras heurísticas”.

Este tipo de técnicas juegan un papel fundamental en la Investigación Operativa ya que principalmente se aplican a problemas de optimización combinatoria, aportando muy buenos resultados. Muchas de estas técnicas se inspiran en procesos biológicos, fenómenos naturales, procesos físicos, etc.

Dentro de los distintos tipos de metaheurísticas podemos distinguir 3 grandes grupos: (1) Métodos basados en la búsqueda en el vecindario, (2) Métodos basados en la búsqueda en la población y (3) enfoques híbridos.

Se presentan a continuación distintas metaheurísticas de estos tres grupos aplicadas en la literatura a la resolución de problemas VRP.

#### 2.4.3.4.1 Métodos basados en la búsqueda en el vecindario

Estos métodos exploran de forma iterativa el vecindario de una única solución. Los algoritmos de búsqueda en el vecindario se encuentran principalmente en el estudio de problemas VRP a gran escala, ya que la mayoría de las soluciones se basan en procesos de búsqueda local. Entre algunos de los algoritmos existentes es posible destacar:

- ***Simulated Annealing (SA)***: o Recocido Simulado en castellano, es un método probabilístico propuesto por [Kirkpatrick et al. 1983] y [Černý 1985]. La idea fundamental de esta técnica es encontrar el mínimo global de una función que puede tener varios mínimos locales no factibles. Este proceso toma su nombre de un proceso físico donde un metal sólido es calentado a una temperatura máxima en la cual todas las partículas se colocan de forma aleatoria en un estado líquido. Posteriormente este líquido es enfriado lenta y cuidadosamente a través de una disminución de temperatura hasta el estado sólido de nuevo y las partículas se colocan en una estructura con mínima energía molecular. En la versión computacional, la temperatura es discretizada y, como analogía con la situación física, se permiten soluciones peores que la mejor encontrada con una probabilidad que disminuye según transcurren las iteraciones del algoritmo (enfriamiento). Esto hace que en el comienzo del algoritmo sea posible aceptar peores soluciones y posteriormente sea más difícil, permitiendo, así, evitar óptimos locales. Su implementación es sencilla y posee varios parámetros que pueden ser configurados para mejorar su funcionamiento como el patrón de enfriamiento o la probabilidad de aceptar una solución peor. Esta versión tiene carácter estocástico, sin embargo, existe una versión determinista de este algoritmo denominada *Deterministic Annealing*, que presenta un funcionamiento similar, pero sigue una regla determinista para aceptar el movimiento.
- ***Tabu Search***: o Búsqueda Tabú en castellano, fue introducida por primera vez por Glover en 1973 como una técnica metaheurística general para la resolución de

problemas de optimización combinatoria. En este algoritmo se analizan secuencias de soluciones, al igual que en el algoritmo anterior, con la diferencia de que en este caso se selecciona el mejor vecino de la solución actual. Para evitar comportamientos cíclicos, se prohíben (se hacen tabú) las soluciones examinadas recientemente para un determinado número de iteraciones (memoria a corto plazo). Este algoritmo posee también una memoria a largo plazo que se tiene en cuenta para la exploración global del algoritmo. El mecanismo de la búsqueda tabú puede ser perfeccionado incluyendo estrategias de intensificación y diversificación. Este algoritmo ha sido aplicado a problemas VRP y mejorado gradualmente por multitud de autores, entre ellos, [Toth and Vigo 2003] que incorporaron una búsqueda tabú granular con buenos resultados.

#### 2.4.3.4.2 Métodos basados en la búsqueda en la población

Esta clase de métodos genera nuevas soluciones, evolucionando las soluciones actuales y creando nuevas combinaciones a partir de ellas. Este tipo de técnicas a menudo se inspiran en comportamientos que se dan en la naturaleza.

Entre los más utilizados en la literatura cabe destacar:

- **Algoritmos genéticos:** se trata de técnicas aleatorias de búsqueda global que son capaces de resolver problemas mediante la imitación de procesos observados durante la evolución natural. El paradigma de resolución de este tipo de problemas fue inicialmente propuesto por [Holland 1975]. Un algoritmo genético puro es una técnica de resolución de problemas genéricos que aplica una pequeña cantidad de información heurística a todo el ámbito del problema. De esta forma, es posible aplicarlo en un amplio rango de problemas poco definidos que no es posible solventar con métodos más específicos. Su funcionamiento consiste en generar una población de cadenas de dígitos (bits), también denominada cromosomas, donde cada uno de estos presenta una solución codificada para un determinado problema. La evolución se lleva a cabo a partir de la aplicación de unos operadores, que imitan los fenómenos observados en la naturaleza como son la mutación, etc. Este tipo de operaciones evitan que el algoritmo se estanque en óptimos locales.
- **Colonias de hormigas:** este tipo de técnicas se basan en que en la naturaleza varias hormigas exploran distintas direcciones del espacio de soluciones factibles, dejando tras de sí un rastro de feromonas que indican a la siguiente hormiga las direcciones más prometedoras que deben ser exploradas. Las direcciones son tomadas con una probabilidad proporcional al nivel de feromona existente, intentando de esta forma evitar los mínimos locales. Asimismo, la cantidad de feromonas disminuye progresivamente tras cada iteración (se desvanecen). Este algoritmo fue usado por

primera vez por Dorigo et al. [DORIGO and M. 1992], en trabajos con problemas VRP por Bullnheimer et al. [Bullnheimer et al. 1999] y en trabajos más recientes como Bell et al. [Bell and McMullen 2004] y Ting et al. [Ting and Chen 2013], entre otros.

#### 2.4.3.4.3 Métodos híbridos

Los métodos metaheurísticos híbridos combinan características de distintas metodologías para aprovecharse de sus diferentes fortalezas. Los elementos de una metaheurística pueden ser incluidos en una metodología totalmente distinta o incluso los algoritmos pueden ser empleados consecutivamente. Esta categoría de métodos metaheurísticos híbridos no solo se refiere a conceptos de metaheurísticas, también incluyen procedimientos de procesos de búsqueda, programación matemática, programación con restricciones, etc.

Se trata de una categoría difícil de delimitar ya que puede cubrir numerosas estrategias y las propias metaheurísticas pueden ser vistas como combinación de heurísticas. Para clarificar esta categoría, se puede definir una metaheurística híbrida como aquella que busca explotar las ventajas de diferentes metodologías y sus ideas para obtener mejores resultados.

Existe una amplia variedad de métodos híbridos en la literatura relacionada con problemas VRP. Estas soluciones híbridas pueden combinar metodologías centradas en la búsqueda de vecindario como *Simulated Annealing* y *Tabu Search* por [Osman 1993] y muchas otras combinaciones. Esta área de investigación se presenta prometedora en el campo de los métodos de resolución VRP.

#### 2.4.4 La incertidumbre en el CVRP

En las secciones anteriores se han descrito las tipologías de problemas VRP y se han descrito varias técnicas para la resolución de la versión determinista de este problema. Sin embargo, los problemas en un entorno real de aplicación presentan incertidumbre en sus parámetros, por lo que es necesario introducir esta incertidumbre en los métodos de resolución.

Como se ha visto anteriormente, la familia de problemas VRP que tiene en cuenta la incertidumbre en su formulación es la denominada SVRP (*Stochastic VRP*). Se define, de forma general, como un VRP donde uno o varios de los parámetros son estocásticos y siguen una distribución de probabilidad conocida.

En la literatura este tipo de problemas se subdividen en función de dónde tiene lugar dicha incertidumbre:

- **Cientes estocásticos (*VRP Stochastic Clients - VRPSC*):** En este caso no se conoce si el cliente será atendido o no, es decir, cada cliente presenta una demanda determinista

y existe una probabilidad de que éste se encuentre presente o no para la elaboración de las rutas.

- **Demandas estocásticas (*VRP Stochastic Demands - VRPSD*):** es la versión más frecuente y se trata de un problema en el que no se conoce la cantidad exacta de demanda de cada uno de los clientes; solo se conoce una distribución de probabilidad de la demanda para cada uno de ellos.
- **Combinación de clientes y demandas estocásticas:** esta versión del problema considera una combinación de las dos anteriores.
- **Tiempos de viaje estocásticos (*VRP Stochastic Travel Times - VRPSTT*):** se trata de incluir incertidumbre relacionada con el tráfico, clima y habilidades de los conductores en el problema VRP, traducándose en valores de costes de desplazamiento de un lugar a otro estocásticos.

Otra parte importante de estos problemas es cuándo la información se convierte en determinista, es decir, cuándo se revela y los parámetros con incertidumbre se determinan. Dependiendo de cuándo se revele dicha información, los métodos de resolución variarán ya que se podrían realizar cambios en las rutas calculadas considerando la nueva información obtenida. En este trabajo se toma como punto de partida el problema VRPSD y se considera que la información se revela en el momento en el que se visita a un cliente.

El tipo de métodos de resolución también es específico para esta familia de problemas. Entre los distintos tipos de métodos de resolución que se han desarrollado para abordar estos problemas se pueden encontrar en la literatura:

- ***Stochastic programming with recourse*:** este enfoque fue desarrollado por separado por Dantzig [Dantzig 1955] y por Beale en 1955[Beale 1955]. La idea clave de este enfoque consiste en separar el problema en dos etapas distintas entre las cuales la información es revelada. El caso más básico de este enfoque comprende solo 2 etapas: la primera, en la que se implementa una solución que tendrá que lidiar con las consecuencias generadas de la información revelada en el futuro, y la segunda que lidiará con las llamadas *recourse actions* o “acciones correctoras”, llevadas a cabo para adaptar las rutas después de obtener la información determinista. Es importante destacar que las acciones correctoras (de segunda etapa) dependen de la decisión tomada en la primera etapa y de la información revelada posteriormente. Por tanto, este tipo de técnicas buscan encontrar soluciones para la primera etapa que optimicen el valor esperado de todas las decisiones.



- ***Dynamic programming***: propuesto por Bellman en 1958 [Bellman 1958]. Este enfoque es desarrollado para manejar de forma efectiva problemas de decisiones secuenciales. Este tipo de métodos se basa en la descomposición temporal del problema en etapas, intenta explotar el principio de optimalidad. Estos enfoques son buenos para problemas con un número limitado de posibles estados y acciones que se vayan a llevar a cabo.
- ***Chance-constrained programming***: propuesto por Charnes and Cooper en 1959 [Charnes and Cooper 1959]. La idea clave de este enfoque es permitir que algunas de las restricciones sean satisfechas solo con cierta probabilidad.
- ***Robust optimization***: en este enfoque, la incertidumbre es representada como un vector de parámetros estocásticos debe pertenecer a un conjunto poliédrico (sin ninguna probabilidad definida). Por ejemplo, en un VRPSD, tener un conjunto de límites superiores e inferiores para cada demanda junto con un límite superior de la demanda total. Este tipo de enfoque es el más conservador y busca una solución que proporcione el mejor “peor caso”. Su principal problema es que proporciona soluciones demasiado conservadoras, pero aseguran la factibilidad de la ruta.

Aparte de estos métodos de resolución clásicos, han surgido recientemente nuevas metodologías de resolución como la que se verá en la siguiente sección.

### 2.4.5 Simheuristic

En los últimos años se ha desarrollado una metodología propuesta por Juan et al. [Juan et al. 2011] que trata los problemas SVRP con una metodología diferente a la vistas anteriormente; en ella se combinan técnicas de optimización y simulación recibiendo el nombre de Simheuristic [Juan et al. 2015b].

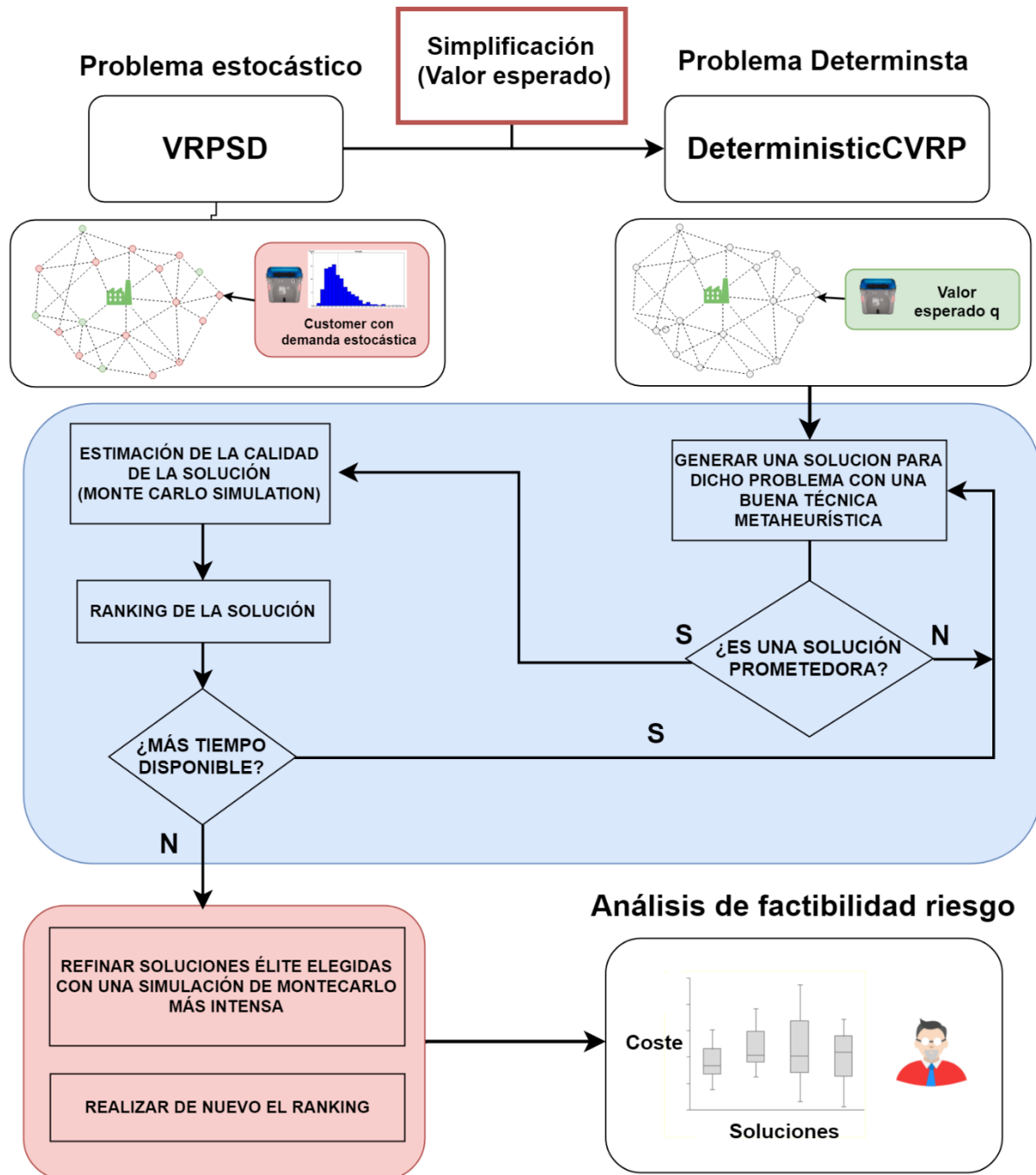
Este enfoque asume que, en escenarios con una incertidumbre moderada, soluciones de alta calidad válidas para un problema VRP determinista pueden ser también soluciones de alta calidad válidas para su versión estocástica. Por supuesto esto no implica que la mejor solución encontrada para el problema determinista sea la mejor solución para el problema estocástico. Los autores afirman que esta metodología no debería ser aplicada en escenarios con una variabilidad muy alta debido a los grandes cambios que pueden existir entre el escenario determinista y el real.

Esta premisa en la relación entre soluciones en el plano estocástico y el determinista permite a los autores generar una serie de soluciones prometedoras de alta calidad para la versión estocástica del problema a través de una versión determinista de este. Esta última se consigue con una simplificación del problema estocástico, transformando sus demandas estocásticas en variables deterministas a través de, por ejemplo, el valor esperado (en este trabajo se detallará más adelante cómo se propone modificar este punto de la metodología). A continuación, se realiza una búsqueda de soluciones con un método de solución eficiente para problemas deterministas (algoritmos del estado del arte vistos con anterioridad) que iterativamente proporcione soluciones de alta calidad que serán posteriormente evaluadas. Esta evaluación consiste en estimar la calidad y factibilidad de la solución encontrada, empleando métodos de simulación de escenarios para manejar la aleatoriedad de estas variables.

Estos escenarios podrán ser satisfechos por la solución determinista o, por el contrario, se deberá aplicar una *recourse action* que incrementará el coste de la solución y disminuirá el porcentaje de factibilidad de esta respecto a todos los escenarios evaluados en la simulación. Los valores estimados de esta etapa permiten crear un ranking de soluciones para el problema estocástico. Una vez haya finalizado el tiempo, se realiza una simulación de escenarios más intensiva en el número de réplicas. Todo ello, con el objetivo de realizar un nuevo ranking de las soluciones y ofrecer al *decision maker* una visión general de esas soluciones élite y sus probabilidades de tal forma que pueda realizar un *risk/reliability* análisis de ellas. Este último punto es una herramienta especialmente útil que ofrecen enfoques como el de *Simheuristic approach* a los *decision maker* ya que pueden evaluar qué solución aplicar dependiendo del riesgo que estén dispuestos a asumir.

En la **Figura 12** se muestra un diagrama de la metodología propuesta por Juan et al. [Juan et al. 2015b] descrita anteriormente. Cabe, destacar en la **Figura 12** el proceso de simplificación

del problema estocástico que se verá modificado en el sistema propuesto en este trabajo de tesis.



**Figura 12** Esquema completo de la metodología *Simheuristic* propuesta por Juan et al. [Juan et al. 2015b]

---

## CAPÍTULO 3

---



**VNiVERSiDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

## SISTEMA PROPUESTO

---

## Capítulo 3 Sistema propuesto

Este capítulo presenta el sistema propuesto en este trabajo de tesis. Se presentan tanto la arquitectura general del sistema como cada una de las partes que lo componen.

A lo largo del presente capítulo se presenta el sistema propuesto en este trabajo de tesis doctoral. Se trata del diseño de un sistema de recogida de residuos que tiene como objetivo dar respuesta a la hipótesis planteada en este trabajo. Incluye un sistema de obtención de datos mediante redes inalámbricas de sensores que se pueda integrar con los sistemas actuales de recogida, un sistema de predicción de la generación de residuos a partir de los datos previamente obtenidos, finalmente, un sistema de optimización de rutas de recogida que considere la incertidumbre presente en la recogida de desechos.

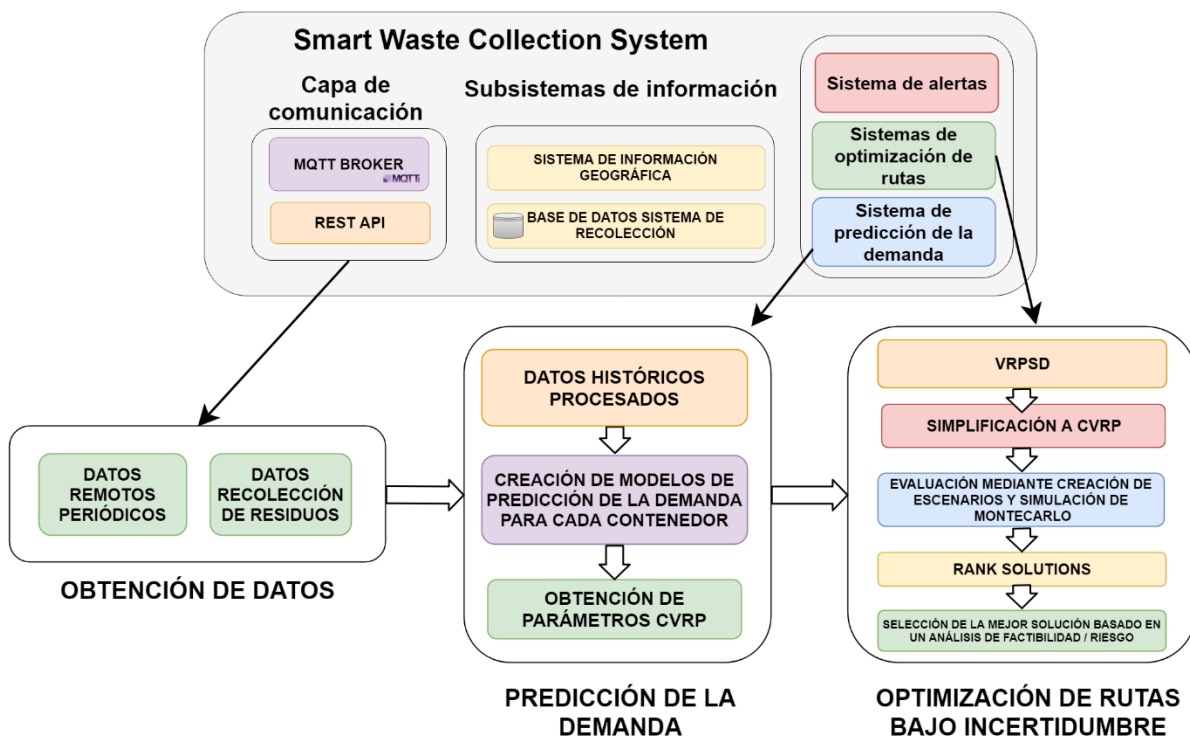
En primer lugar, se describirá la arquitectura general del sistema, identificando las distintas partes que conforman el diseño de este. Posteriormente se describirá en detalle cada una de las partes del sistema, detallando la propuesta de funcionamiento y cómo se emplean las técnicas previamente revisadas en el capítulo del estado del arte.

### 3.1 Arquitectura general del sistema

El sistema propuesto abarca tres ámbitos fundamentales: la obtención de los datos por una red de sensores especialmente diseñada para un sistema de recolección de residuos, la utilización de dichos datos para generar modelos de predicción que permitan obtener los valores de demanda más precisos y el posterior empleo de estos en la optimización de las rutas de recogida incluyendo metodologías que manejen la incertidumbre en las demandas. En la **Figura 13** se muestra un diagrama esquemático general del sistema propuesto. En la parte superior de la misma se encuentra un diagrama que representa un sistema de recolección de residuos inteligente o *Smart Waste Collection System* con diferentes partes:

- **Capa de comunicación:** tanto para obtener datos de sensores como para proveer datos generados en el sistema. Esto incluye datos generados por los contenedores del sistema, tanto obtenidos remotamente como durante la propia recolección.
- **Subsistemas de información:** cada uno de ellos desempeña una función determinada en el sistema. Se destacan los siguientes:

- **Sistema información geográfica:** este sistema es el encargado de suministrar al resto de subsistemas de información geográfica necesarios para la optimización y creación de rutas de recogida de residuos.
- **Sistema de base de datos de recogida de residuos:** almacena la información relativa a cada uno de los contenedores, sensores, vehículos y rutas de recogida de residuos del sistema. Se almacenan los datos históricos obtenidos por la red de sensores y las rutas de recogida realizadas.
- **Sistema de alertas:** sistema encargado de la monitorización de los datos obtenidos por los sensores y de la generación de alertas relativas a un mal funcionamiento o a valores extremos de temperatura en alguno de los contenedores.
- **Sistema de predicción de la demanda:** esta parte del sistema es la encargada de la predicción de la cantidad de residuo que presentará cada contenedor para las sucesivas recogidas. Los resultados de este sistema serán empleados por el sistema de optimización para elaborar las rutas de recogida.
- **Sistema de optimización de rutas:** esta parte tiene como función principal realizar una optimización de las rutas de recogida, incluyendo metodologías que incorporen la incertidumbre en la demanda de cada uno de los contenedores y empleando los modelos de predicción generados en el sistema de predicción de la demanda.



**Figura 13** Diagrama general del sistema propuesto

En la parte inferior de la **Figura 13** se puede observar un esquema de las tres partes fundamentales del sistema propuesto en las que se centra este trabajo de tesis:

- **Obtención de datos:** tanto de forma remota como en la propia recolección de residuos (parte inferior izquierda de la **Figura 13**), incluyendo estos dos tipos de datos en el sistema. Se propone una WSN para obtener los datos de los contenedores junto con un sensor de bajo consumo que obtenga valores de peso, volumen y temperatura.
- **Predicción de la demanda:** a partir de los datos históricos obtenidos y de los datos obtenidos de otras fuentes (calendario, tiempo climatológico, etc.) se generan modelos de predicción de la cantidad de residuos que va a presentar cada uno de los contenedores (parte inferior central de la **Figura 13**), con el objetivo de que estas predicciones sean útiles para la posterior optimización de rutas.
- **Optimización de rutas bajo incertidumbre:** en esta parte se propone una modificación de la metodología propuesta por Juan et al. [Juan et al. 2011] con el objetivo de optimizar rutas de recogida, incluyendo la incertidumbre en la demanda y empleando modelos de predicción (parte inferior derecha de la **Figura 13**).

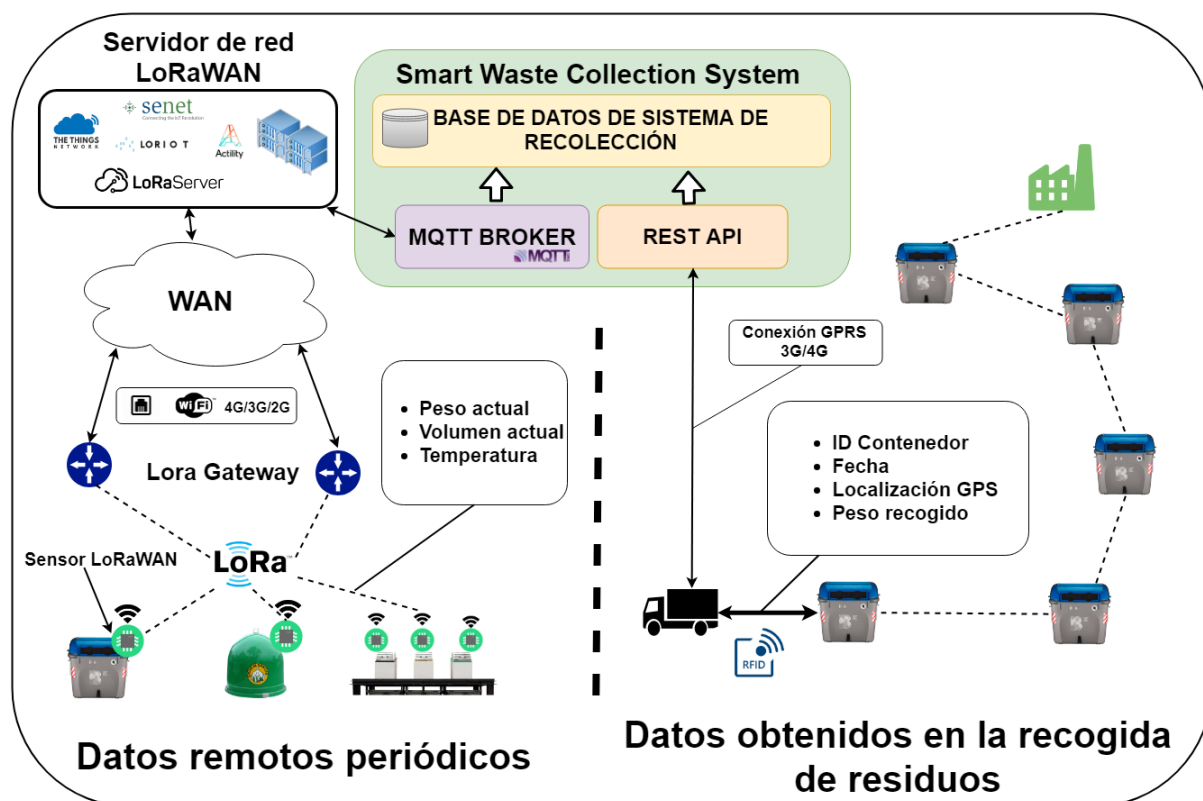
A continuación, se detallan cada una de estas tres partes de la propuesta en profundidad, describiendo su funcionamiento e interacción con el resto de los elementos del sistema.

## 3.2 Subsistema de obtención de datos

Se plantea un subsistema de toma de datos relativos a los contenedores de desechos que integre información de distintas fuentes. De forma general, existen dos tipos de datos, obtenidos de dos fuentes distintas: datos periódicos de llenado de contenedores de desechos obtenidos de forma remota y datos de llenado obtenidos durante la propia recogida de desechos. La **Figura 14** muestra un esquema general de ambos tipos de obtención de datos.

En la parte derecha de la figura **Figura 14** se puede observar un esquema que representa el método de obtención de datos durante la recogida de desechos. Este método de obtención de datos es el que presentan los sistemas de recogida de residuos convencionales. En él, cada uno de los vehículos identifica el contenedor que se dispone a recoger mediante un lector RFID. Para obtener el peso del contenedor, el operario realiza una medición antes de depositar los desechos en el vehículo gracias a las células de carga que incorpora el gancho empleado por el vehículo para levantar el contenedor. Una vez depositado el desecho, el operario realiza otra medida, obteniendo de este modo el peso del contenedor vacío. Posteriormente, la información relativa al identificador del contenedor, a la localización GPS, a la fecha de la recogida y al peso recogido de dicho contenedor se envía al *Smart Waste Collection System* mediante una

conexión GPRS y a través de una API de servicios web REST. Esta información se almacena posteriormente en el sistema de base de datos del *Smart Waste Collection System* para su posterior utilización por el resto de las partes del sistema.



**Figura 14** Diagrama general del sistema de obtención de datos

En la parte izquierda de la figura **Figura 14** se presenta un esquema de la red de sensores inalámbrica propuesta en este trabajo como forma de obtención remota de datos de forma periódica.

Los datos relativos al volumen, peso y temperatura del contenedor se obtienen periódicamente mientras que valores anómalos de temperatura se reportan inmediatamente si estos son detectados. A continuación, estos datos se almacenan en el sistema de base de datos para su posterior uso.

De esta forma, en el sistema existirán tipos de datos de nivel de llenado de contenedores que pueden estar disponibles el mismo día en el que se realizará la planificación de las rutas (los recibidos de aquellos contenedores dotados de un sensor) y, por otro lado, datos históricos del nivel de llenado obtenido tras la recogida.



En los *Smart Waste Collection Systems* actuales la implantación de sensores se realiza de forma progresiva, de forma que la flota de contenedores se dota de sensores progresivamente dada la inversión inicial necesaria para el despliegue. Debido a este motivo, estos sistemas frecuentemente presentan una flota de contenedores mixta en la que una parte está dotada de sensores y la otra no, lo que provoca que los contenedores convencionales presenten una incertidumbre en el nivel de desechos durante la recogida. Por esta razón, el sistema propuesto incluye datos de las dos fuentes mencionadas, teniendo en cuenta la incertidumbre que presentan aquellos contenedores que no disponen del sensor de bajo consumo propuesto.

A continuación, se detalla la red inalámbrica de sensores propuesta junto con el diseño del sensor LoRaWAN.

### 3.2.1 Red inalámbrica de sensores LoRaWAN y sensor de bajo consumo

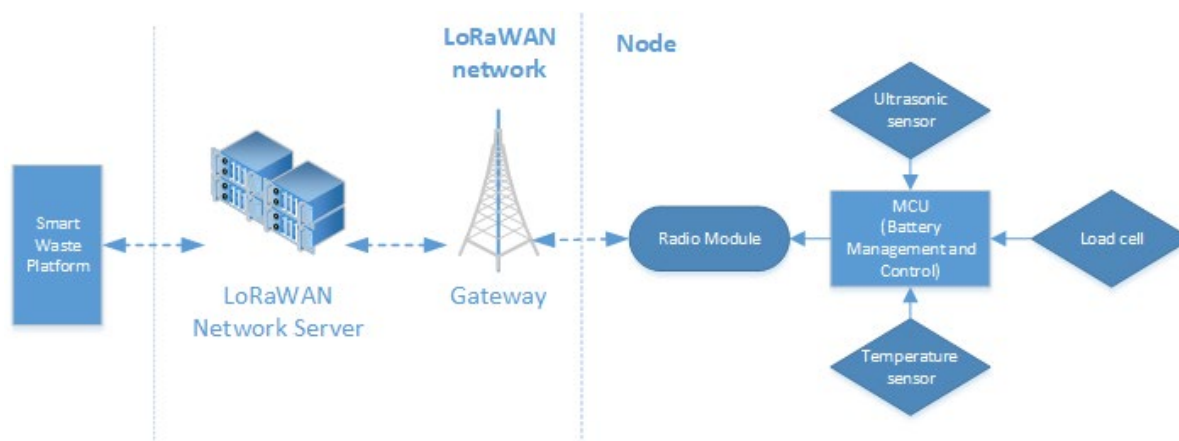
Esta red de sensores inalámbricos está basada en el protocolo LoRaWAN y, por tanto, presenta 3 componentes básicos:

- **Sensor LoRaWAN:** sensor de bajo consumo encargado de tomar medidas de peso, volumen y temperatura del contenedor. Se ha diseñado un sensor de estas características que se detalla más adelante.
- **Gateways:** encargados de dar cobertura a la zona donde se desplegarán los sensores y serán los responsables de actuar como pasarelas entre los servidores de red y los nodos.
- **Servidor de red LoRaWAN y servidor de aplicación:** se ha optado para el diseño del sistema por la utilización de The Things Network [The Things Network 2016] como plataforma que incorpora tanto el servidor de red LoRaWAN como la aplicación con la que integrar el bróker MQTT del sistema propuesto. De este modo el sistema permanece desacoplado de esta tecnología.

En la **Figura 15** se muestran los elementos de la red de sensores descritos anteriormente. En la parte derecha de la misma figura se describen los módulos funcionales del sensor propuesto y su interacción con el resto del sistema.

Se ha optado por el diseño de un dispositivo propio en lugar de utilizar otro tipo de placas disponibles en el mercado como, por ejemplo, RN2483 LoRa Mote de la empresa [Microchip Technology Inc] o *Waspote* de la empresa [Libelium], como base del desarrollo. Esta opción permite la elección de cada uno de los componentes del nodo de manera individualizada (desde el microcontrolador hasta cada uno de los sensores), haciendo posible incrementar la eficiencia energética del dispositivo final. Además, este enfoque permite diseñar una solución que se

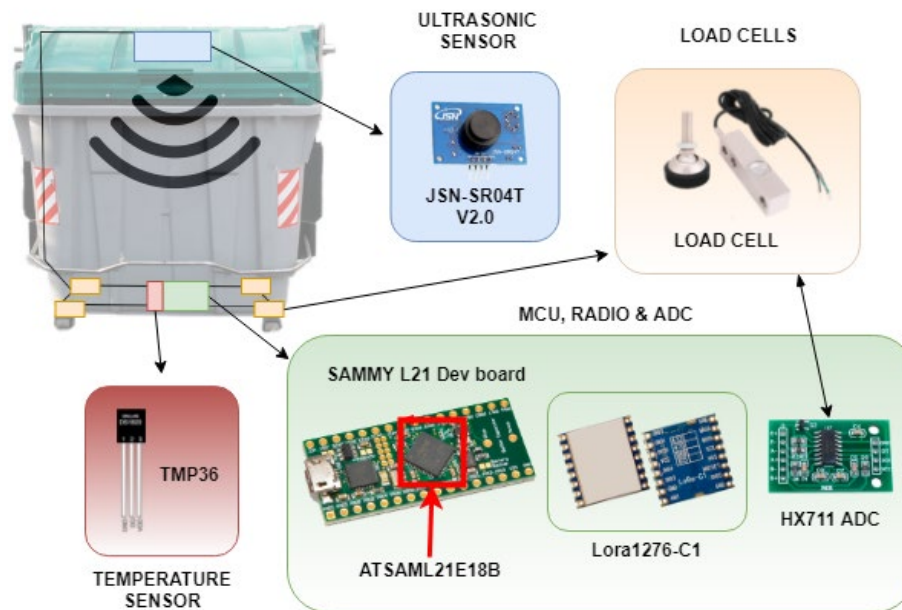
adapte específicamente a los requisitos relativos a la adquisición de datos que necesita el sistema, empleando los sensores más indicados para cada tipo de dato.



**Figura 15** Diagrama de comunicación general del sistema propuesto

El dispositivo se ha diseñado para obtener datos relativos a 3 variables: volumen estimado de llenado de los contenedores, peso de los residuos dentro de ellos y temperatura en su interior. Al contrario que otros trabajos de la literatura, en los que únicamente se estima el volumen de llenado [Ramson and Moni 2017b], en este trabajo se propone el desarrollo de un nodo que sea capaz de obtener adicionalmente el peso para su empleo en futuras optimizaciones de rutas y la temperatura para la detección de situaciones peligrosas. Estas pueden ser debidas a la alta temperatura y existencia de residuos dentro del contenedor o a la posible aparición de una combustión en este. Como se muestra en la parte derecha de la **Figura 15**, el dispositivo cuenta con distintos módulos funcionales: un sensor de ultrasonidos, células de carga y sensor de temperatura (cada uno de estos para obtener los datos previamente mencionados) y otra parte encargada de la gestión del consumo eléctrico y las comunicaciones con la red (microcontrolador MCU y módulo de radio LoRa).

El nodo emplea la estructura del contenedor y un espacio reservado por el fabricante para su localización en la parte inferior. La **Figura 16** muestra la disposición de los módulos empleados para el prototipado en el contenedor: la parte inferior contiene la placa incluyendo los distintos módulos y sensores empleados, las células de carga se encuentran situadas en cada una de las patas del contenedor y el sensor de ultrasonidos en la parte superior del mismo.



**Figura 16** Localización de los módulos funcionales del nodo y las placas de desarrollo empleadas para el prototipo.

Los sensores empleados para el desarrollo del nodo se detallan a continuación:

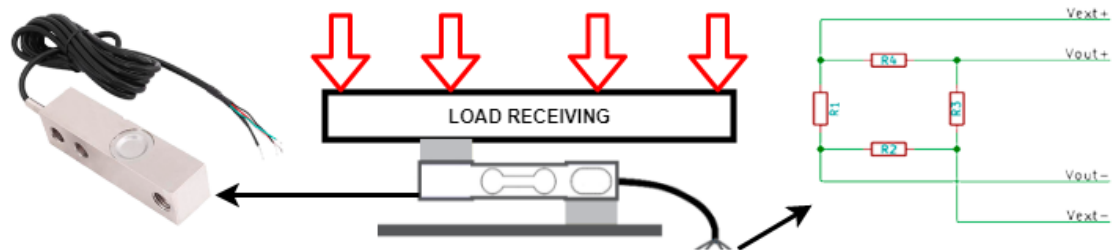
- **Sensor de ultrasonidos:** es el modelo JSN-SR04T V2.0 [Bakar et al. 2017]. Se trata de un sensor resistente al agua y con un rango de operación de 20 cm a aproximadamente 4m, con una precisión de 3mm y un ángulo de operación de 15°. Su funcionamiento se basa en la estimulación de un *pinger* y en la medición del tiempo que transcurre entre la estimulación y el retorno al receptor. Cuando se recibe la señal, se calcula la distancia siguiendo la siguiente ecuación (9):

$$d = \frac{T * v_s}{2} \tag{9}$$

Donde  $d$  es la distancia desde el dispositivo a la primera superficie que alcanza el ultrasonido,  $T$  indica el tiempo transcurrido desde que el ultrasonido se emitió hasta la recepción de su eco y  $v_s$  es la velocidad del sonido. Con este sensor se calcula el volumen aproximado del contenedor. En la **Figura 16** se muestra una ilustración de este módulo.

- **Células de carga:** empleadas para la medición del peso del contenedor. Se han empleado 4 células de carga acopladas a cada una de las patas del contenedor. Este tipo de sensores están basados en el principio de funcionamiento de un puente de

Wheatstone. En la **Figura 17** se puede observar un diagrama de funcionamiento de una célula de carga.



**Figura 17** Diagrama de una célula de carga y el circuito eléctrico asociado

Cuando se produce una variación de peso en el transductor, la resistencia se ve modificada debido a sus características intrínsecas. Dado que existe una diferencia de voltaje continua ( $V_{ext+} - V_{ext-}$ ), se puede calcular la diferencia de voltaje generada mediante la ecuación (10):

$$V_{out} = V_{ext} * \left( \frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) \quad (10)$$

Donde cada  $R_i$  es cada una de las resistencias. Debido a que los cambios en estas medidas son realmente pequeños, es necesario emplear un amplificador analógico-digital (modelo HX711, de 24 bits de resolución, en este caso) para obtener la suficiente precisión y rango. Este módulo se muestra en la **Figura 16** junto con el microcontrolador y el módulo de radio. El amplificador recibe la señal de las células de carga y convierte las pequeñas variaciones en medidas digitales que son enviadas al microcontrolador. El peso del contenedor se puede obtener calibrando la diferencia de voltaje obtenida con un valor de peso conocido.

- **Sensor de temperatura:** se ha empleado un sensor analógico modelo TMP36 para tal fin. Este sensor proporciona medidas con una precisión de un grado centígrado. En este caso la precisión en la temperatura no es un factor crucial en el desarrollo del nodo, pero su comunicación, de tipo analógico, es vital para la posterior eficiencia energética del nodo. En la **Figura 16** se puede observar el sensor de desarrollo seleccionado.
- **Microcontrolador (MCU):** uno de los aspectos clave del diseño del nodo propuesto en este trabajo es el microcontrolador elegido. Se trata de un Atmel modelo SAML21, con un procesador RISC de 32 bits que pertenece a la familia de procesadores ARM Cortex-M-0+ de ultra bajo consumo. Este microcontrolador presenta un consumo en activo de 35  $\mu\text{A}/\text{MHz}$  y menos de 0.5  $\mu\text{A}$  en *standby* cuando trabaja a un voltaje de

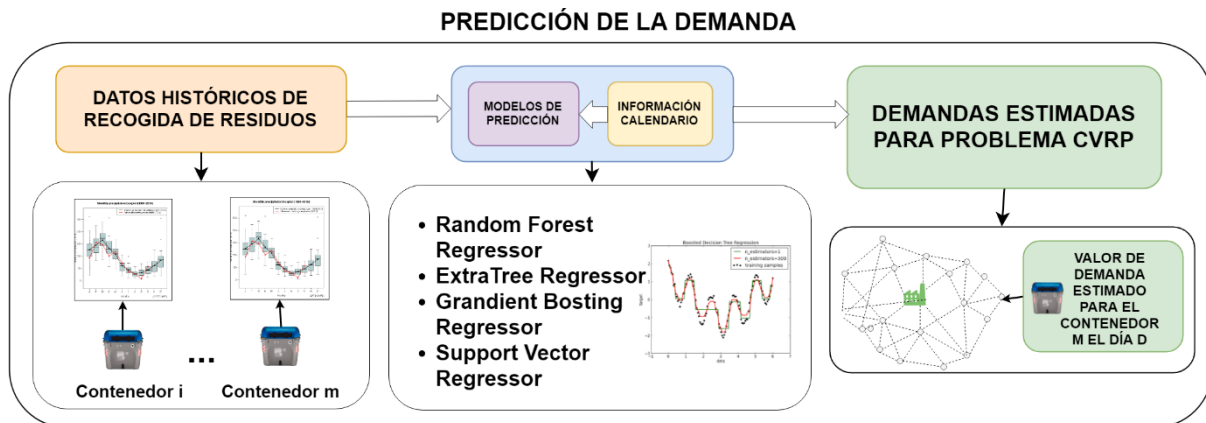
3.3V. Puede realizar varias tareas al mismo tiempo, en este caso, permite obtener los valores de los distintos sensores en paralelo, así como almacenarlos para su posterior transmisión. La elección del microcontrolador adecuado implica estudiar rankings como ULPMark [UPLMark]; el seleccionado para este diseño es el único que se encontraba disponible en dicho ranking con una arquitectura ARM Cortex-0+. Se ha empleado inicialmente como placa de desarrollo S Sammy-L21[Sammy-L21], como puede observarse en la **Figura 16**. Otro aspecto clave de este microcontrolador es que puede realizar lo que se conoce como *Customized Logic Operations* CCL a través de una unidad para tal fin. Esta unidad es realmente útil ya que permite realizar operaciones con las medidas de temperatura, empleando previas mediciones; de este modo, es posible evaluar si la temperatura se ha incrementado de forma brusca o ha sobrepasado cierto umbral para enviar una alerta al sistema. Esta utilidad es creada empleando puertas lógicas, lo que permite que no dependa de un programa síncrono en ejecución en el microcontrolador. Por tanto, es posible el funcionamiento de este tipo de unidad aun cuando el microcontrolador se encuentra dormido, pudiendo despertarlo si fuese necesario y aumentando, así, la eficiencia energética de todo el nodo.

- **Módulo de radio:** dado que se trata de un nodo en una red LoRaWAN, es necesario un módulo de radio LoRa para transmitir la información obtenida por los sensores mencionados previamente. El módulo elegido para tal fin es el modelo LoRa1276-C1 desarrollado por NiceRF [Nice RF]. Posee un gran aislamiento frente al ruido y presenta un bajo consumo en modo dormido de alrededor de 0.2  $\mu$ A.

Todo el sensor recibe la energía eléctrica de una batería de reducidas dimensiones. La capacidad de dicha batería variará dependiendo de los requerimientos del caso de estudio. Más adelante, se presenta un caso de estudio en el que se analiza el funcionamiento del sensor junto a una red de *gateways* desplegada para dar cobertura a un sistema de recogida de desechos desplegado en la provincia de Salamanca.

### 3.3 Subsistema de predicción de la demanda

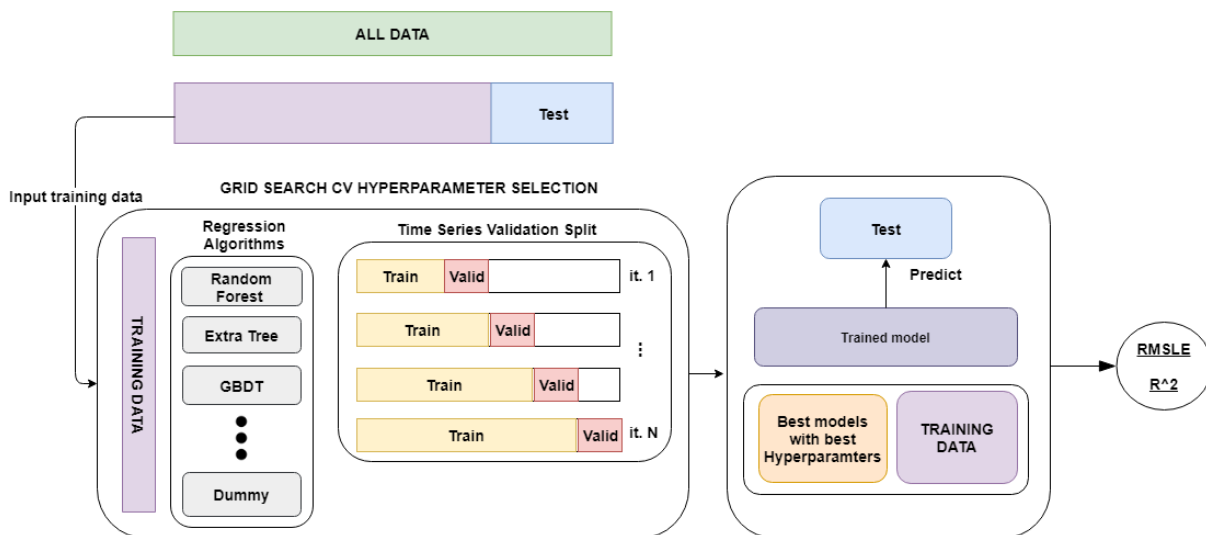
Subsistema encargado de la estimación de la demanda de cada uno de los contenedores, empleando los datos históricos de recogida que han sido previamente adquiridos. Este subsistema obtiene los datos históricos de las recogidas de cada uno de los contenedores, los procesa, y, posteriormente, entrena modelos de predicción para cada uno de ellos con el fin de emplearlos para estimar la demanda en los contenedores de los que no se disponen datos provenientes de sensores.



**Figura 18** Predicción de la demanda a partir de datos históricos de recogida

En la **Figura 18** se puede observar un diagrama general del proceso de predicción de la demanda. En primer lugar, se extraen los datos históricos de demanda, realizando un pre-procesado de los datos para limpiar posibles valores nulos o inválidos. Seguidamente, estos se combinan con datos históricos de las recogidas y del calendario laboral: día del mes, día de la semana, semana del mes, mes, día festivo, presencia de festivo en días anteriores y días desde la última recogida.

La metodología utilizada para la selección de hiperparámetros, creación de los modelos y selección final de los modelos empleados para predecir se muestra en la **Figura 19**.



**Figura 19** Diagrama de la metodología para el entrenamiento de modelos, selección de hiperparámetros y selección de los modelos de predicción

Una vez pre-procesados los datos, se realiza una primera separación entre datos de entrenamiento y datos de test. Posteriormente, se emplean estos datos de entrenamiento para

seleccionar los hiperparámetros de distintos modelos de algoritmos como *Random Forest Regressor*, *Extra Trees Regressor*, *Gradient Boosting Regressor*, *Support Vector Regressor* y, finalmente, un algoritmo *Dummy* que se emplea como base para ver el funcionamiento del resto de los algoritmos (en este caso, su predicción es la demanda media del mes anterior).

Dichos modelos se generan empleando un esquema *Grid Search* para la selección de hiperparámetros. Este método permite seleccionar una grilla de hiperparámetros y genera tantos modelos como combinaciones posibles entre los distintos hiperparámetros especificados en la grilla. Al mismo tiempo, se realiza un proceso de validación cruzada (*Cross Validation - CV*) llamado *Time Series Split Cross Validation*. Este método de validación cruzada se emplea para series temporales, de tal forma que se realizan los conjuntos de entrenamiento y validación de forma progresiva, ya que no es posible entrenar con datos del futuro para predecir posibles instancias del pasado, como ocurriría si se aplica un esquema de validación cruzada como *k-fold Cross Validation*. Finalmente, se estudia el error obtenido tras realizar predicciones con el conjunto de validación separado inicialmente.

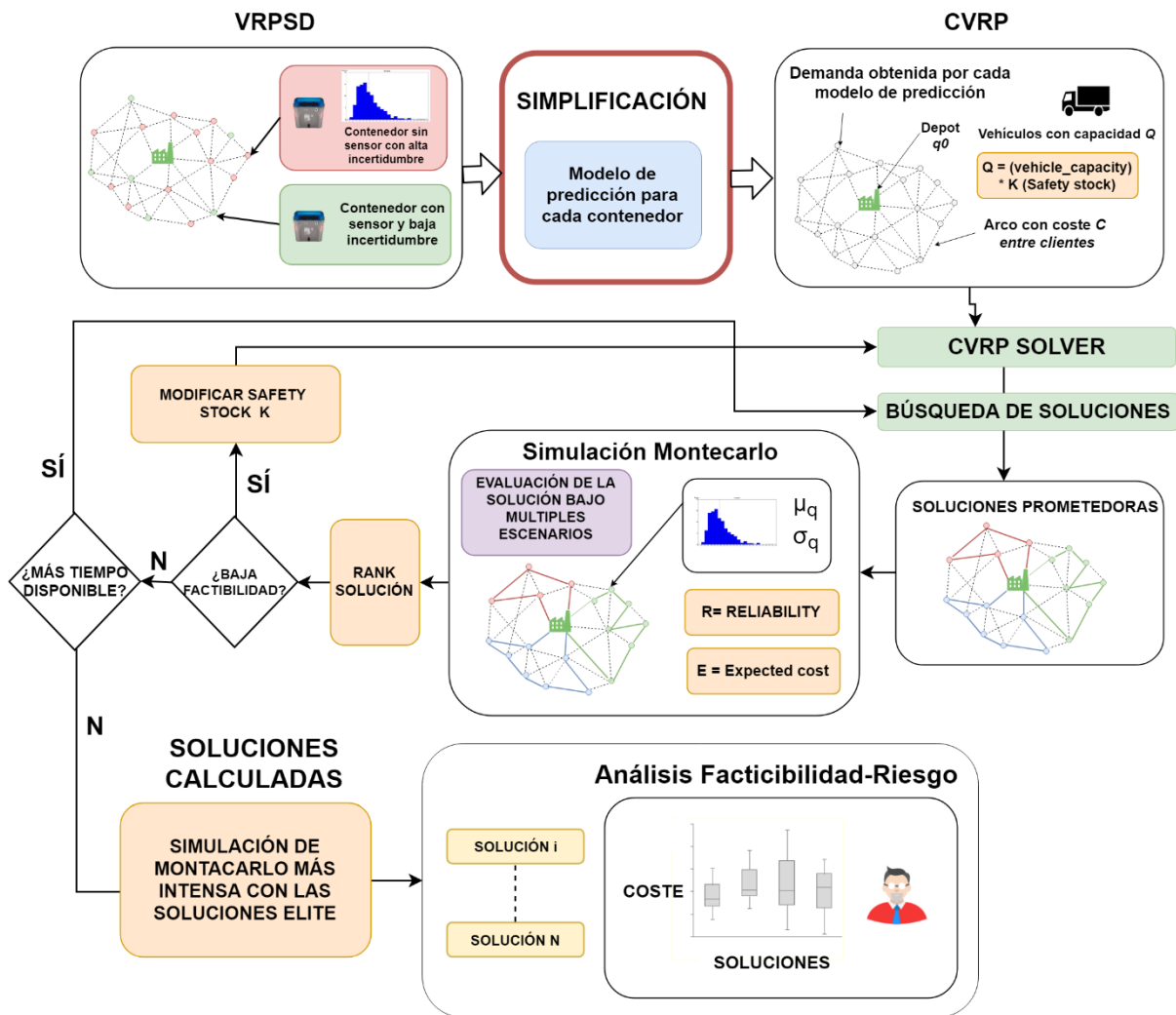
### 3.4 Subsistema de optimización de rutas bajo incertidumbre

Este subsistema es el encargado de realizar la optimización de las rutas de recogida de desechos teniendo en cuenta el estado actual del sistema de recogida del *Smart Waste Collection System* y la incertidumbre asociada a cada uno de los contenedores de recogida de los que dispone. Se plantea la resolución de un problema del tipo VRPSD en el que existen, por un lado, contenedores dotados de sensores en los que la incertidumbre durante la recogida es mínima y, otros, de los que solo se dispone de datos históricos y modelos de predicción para cada uno de ellos que permiten estimar la cantidad de desechos estimada para el día de la recogida.

Para la búsqueda de soluciones óptimas de recogida se emplea como base la metodología propuesta por Juan et Al. [Juan et al. 2011], realizando una modificación con el fin de mejorar el desempeño de esta en la optimización de rutas.

En la **Figura 20** se detalla el proceso de cálculo seguido en este subsistema. En primer lugar, se obtienen los datos relativos a los contenedores de los cuales se va a realizar la recogida; estos pueden incluir contenedores tanto con sensores como sin ellos (**Figura 20** parte superior izquierda). Siguiendo la metodología de Simheuristics se simplifica el problema estocástico a un problema CVRP determinista (**Figura 20** parte superior central). En esta propuesta se emplean los modelos de predicción que mejores resultados han obtenido durante el entrenamiento para obtener el valor estimado de la demanda de cada contenedor para el día de la recogida a diferencia de la propuesta original de Juan et Al. en la que se emplea el valor esperado de la distribución elegida para cada contenedor. Este paso puede ser fundamental ya que, si los modelos generados se ajustan adecuadamente a la demanda real, la solución

determinista se encontrará mucho más cercana a la real y, por tanto, los valores obtenidos en las simulaciones presentarán valores de factibilidad más altos. Por ejemplo, para contenedores que presenten una demanda muy alta en determinadas épocas del año, la simplificación mediante modelos de predicción será más próxima a la real que mediante el valor esperado o la media de una distribución obtenida para dicho contenedor.



**Figura 20** Diagrama de la metodología Simheuristics incluyendo modelos de predicción

Posteriormente (**Figura 20** parte superior derecha), se plantea un problema CVRP clásico añadiendo una variable  $K$  de *safety stock* [Juan et al. 2011] [Gruler et al.]. Esta variable se emplea con el objetivo de reservar espacio en cada camión para posibles excesos de demanda en las posteriores simulaciones. La carga  $Q$  considerada para camión pasa a ser (ecuación (11)):

$$Q = Q' * K \tag{11}$$



Siendo  $Q'$  la capacidad real del camión y  $K$  el porcentaje de vehículo que se desea emplear para portar desechos, de tal forma que a menor  $K$  el espacio reservado es mayor y la factibilidad de las rutas aumenta en caso de exceso de demanda.

Una vez establecidos los parámetros del problema CVRP, se realiza una búsqueda de soluciones mediante el método más eficiente que se seleccione del estado del arte para un problema CVRP (habitualmente los métodos heurísticos y metaheurísticos son los que mejor resultados dan cuando el problema es grande, como se ha visto con anterioridad). A medida que el algoritmo implementado en el denominado *solver* encuentra soluciones prometedoras, estas son evaluadas mediante una simulación de Monte Carlo generando múltiples escenarios con el fin de obtener valores acerca de los costes asociados de cada una de las soluciones y un índice de factibilidad (parte central de la **Figura 20**).

Durante la simulación se tiene en cuenta el espacio reservado en los camiones para excesos de demanda, pero, aun así, pueden darse escenarios en los que la solución falle y sea necesario realizar una *recourse action*, que en este caso se ha determinado como un retorno al *depot* desde el contenedor donde se ha producido el fallo y posteriormente la continuación de la ruta desde el último contenedor visitado. Si cualquiera de las rutas falla, se considera la solución completa como no factible a efectos del cálculo del índice de factibilidad. Al terminar la simulación se obtiene un índice de factibilidad expresado como un porcentaje y un conjunto de costes de cada una de las soluciones. Con estos datos se realiza un *ranking* de soluciones (parte central izquierda de la **Figura 20**). Dependiendo de la  $K$  elegida para el planteamiento del problema, los niveles de factibilidad serán más bajos o más elevados, por lo que, si éstos no cumplen un criterio introducido de antemano, es posible modificar la  $K$  y obtener valores mayores de factibilidad al reservar más espacio. Cabe mencionar que, al reservar más espacio, los costes aumentarán y, en ocasiones, será necesario utilizar más vehículos para dar respuesta a toda la demanda.

Si no hay tiempo disponible para seguir buscando soluciones, se pasa a realizar una simulación de Montecarlo intensiva con las soluciones mejor posicionadas en el *ranking* en función de su factibilidad y coste. Para finalizar, se presentan estas soluciones con los nuevos índices de factibilidad y valores de coste al *decisión-maker* para que evalúe qué solución desea implementar (parte inferior derecha de la **Figura 20**).



---

## CAPÍTULO 4

---



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

## CASOS DE ESTUDIO

---



## Capítulo 4 Casos de estudio

Este capítulo detalla los 3 casos de estudio llevados a cabo para la validación de la hipótesis inicialmente planteada en el trabajo de tesis. Cada uno de los tres casos aborda una parte específica del sistema propuesto

A lo largo de este capítulo se describen los casos de estudio que se han desarrollado en el presente trabajo de tesis. Se trata de tres casos de estudio, centrados en cada una de las partes del sistema propuesto.

El primer caso de estudio está focalizado en la toma de datos y el desarrollo de un sensor de bajo consumo junto a una red de sensores inalámbricos LoRaWAN que permita que estos presenten una alta eficiencia energética y puedan ser desplegados en contenedores para tomar los datos necesarios que se empleen en una posterior optimización de las rutas de los vehículos.

El segundo caso de estudio se centra en el estudio de la predicción de la demanda, esta vez en otro sistema de vehículos distinto, como son los sistemas de alquiler de bicicletas públicos en las ciudades, pero que presenta la necesidad de predecir la demanda que existirá en cada uno de los puestos al igual que sucede con la cantidad de desechos de cada contenedor.

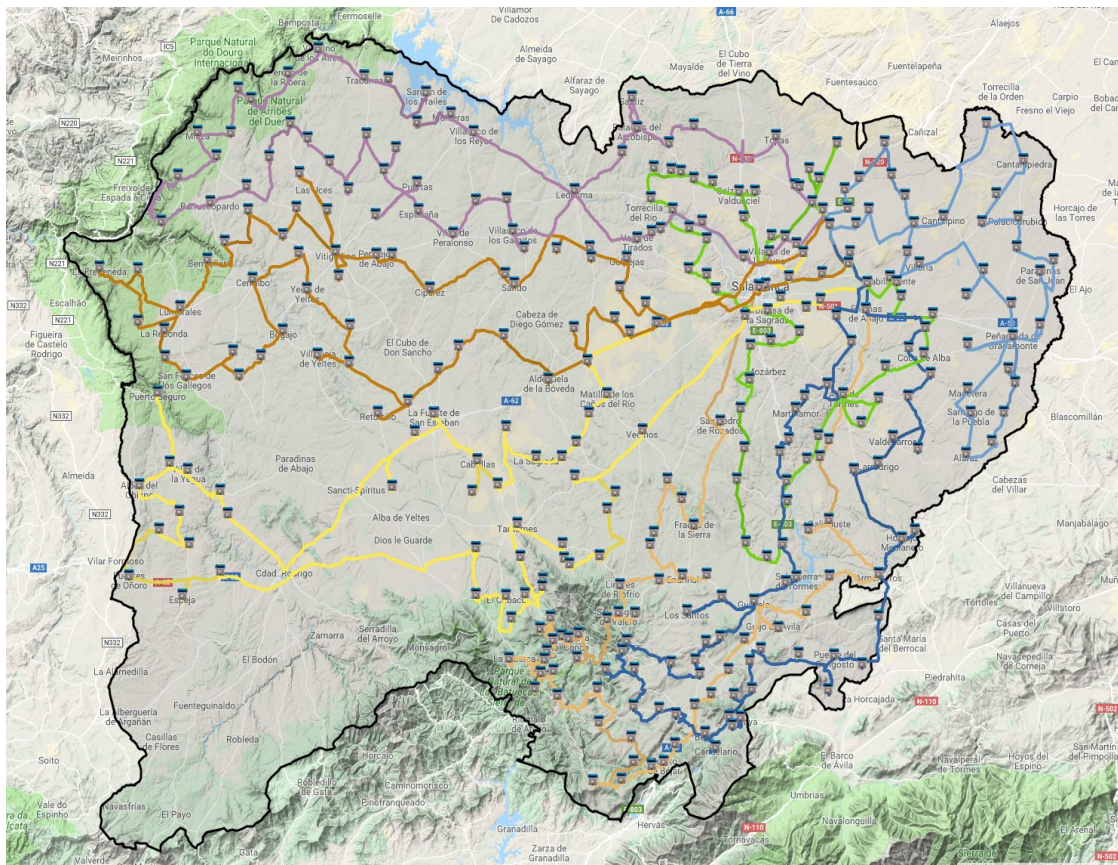
El tercer y último caso de estudio utiliza los datos obtenidos y los modelos de predicción de la demanda que se estudiaron en los trabajos anteriores para introducir mejoras en una metodología de optimización de rutas de vehículos en escenarios en los que existe incertidumbre en la demanda de los clientes a visitar.

### 4.1 Obtención de datos mediante sensores LoRaWAN

Este estudio está basado en la información obtenida de un procedimiento de licitación pública del sistema de recogida de desechos selectivos (papel, cartón y envases) de la Diputación de Salamanca (España) en el año 2016 [Diputación de Salamanca]. En este documento, la Diputación de Salamanca requiere la recolección de desechos para 329 municipios de la región y para sus propias instalaciones. Adicionalmente, se incluye información relevante acerca del número de contenedores de cada pueblo y la producción anual de desechos. En el mismo documento se especifican que es necesaria una flota mínima de 8 vehículos para la recolección.

En este caso de estudio se pretende llevar a cabo un estudio de la red inalámbrica de sensores propuesta en este trabajo para la obtención de datos en un entorno de un Smart Waste Collection

System real: (1) realizando un estudio del consumo energético del sensor propuesto para obtener los datos del contenedor, (2) diseñando un despliegue de los *gateways* necesarios mediante un estudio de la cobertura aplicado a los municipios involucrados en el Smart Waste Collection System, (3) realizando una optimización de rutas, considerando la implantación de los sensores en cada uno de los contenedores y suponiendo que las recogidas se realizan dinámicamente, frente al enfoque fijo convencional.



**Figura 21** Rutas de recogida generadas para el caso de estudio en la provincia de Salamanca

Se ha empleado la información relativa a la licitación para formular el problema de este caso de estudio como un CVRP. Adicionalmente, se ha establecido que los vehículos realizan una recogida cada 15 días como se indica en la licitación y se ha determinado para este caso de estudio que realizan una serie de 8 rutas fijas presentadas en la **Figura 21**. Estas se han obtenido tras plantear un problema CVRP en que se empleen los 8 camiones que indica la licitación que deben estar disponibles para tal fin. La realización de estas rutas tiene un coste aproximado de 3050,725 km en viajes cada 2 semanas.

El caso de estudio ha sido realizado empleando datos sintéticos generados a partir de las estadísticas anuales obtenidas de la licitación de cada uno de los residuos para cada desecho en

cada pueblo. Esta información ha sido cargada en el sistema de bases de datos del *Smart Waste Collection System* y se han incorporado los datos geográficos de la zona a partir de OpenStreetMaps.

En las siguientes secciones se explican y detallan los resultados obtenidos relacionados con el sensor propuesto, la red inalámbrica de sensores y el proceso de optimización de rutas seguido en este caso de estudio con las condiciones expuestas anteriormente para dicho caso.

#### 4.1.1 Resultados obtenidos del nodo desarrollado

Un factor decisivo en el sistema propuesto es la autonomía energética que presentan los nodos desarrollados, ya que, cuanto más eficientes sean, menor será el mantenimiento asociado a cada sensor. En los sistemas de recogida de residuos como el de la licitación del caso de estudio planteado, los contenedores son habitualmente revisados de 2 a 4 veces al año, por lo tanto, implica que el nodo final debe poder realizar su funcionamiento durante al menos 1 año con una batería. Cuanto menor sea el consumo, menor será la batería necesaria y sus costes asociados en el desarrollo.

Para evaluar el consumo energético del nodo desarrollado se ha evaluado el consumo por separado de cada uno de los componentes y, posteriormente, se han agregado todos para dar un consumo total teórico.

Antes de desarrollar el sensor, se han realizado estimaciones teóricas de acuerdo con la información proporcionada por las especificaciones del fabricante de cada componente. En esta información es posible encontrar el consumo de cada componente en distintos modos de funcionamiento y operación que deben ser tenidos en cuenta como, por ejemplo: sensores realizando mediciones, microcontrolador procesando la información o comunicándose a través de la red.

Las diferentes baterías del mercado habitualmente utilizan 2 unidades: Voltios para expresar el Voltaje y Amperios/hora o Ah para expresar su carga eléctrica. Un Ah es una unidad de carga eléctrica que representa la corriente de 1 Amperio pasando a través de un elemento conductor a lo largo de una hora. Esto permite describir la potencia de las baterías y, consecuentemente, calcular la carga que el dispositivo consumirá en su funcionamiento expresado en mAh. Este consumo viene definido por la ecuación (12):

$$Q = \sum_i i_i \cdot t_i = t_{MCU_i} \cdot i_{MCU_i} + t_{radio} \cdot i_{radio} + t_{MCU_a} \cdot i_{MCU_a} + \sum_j t_{sensor_j} \cdot i_{sensor_j} \quad (12)$$

Donde  $Q$  es la carga consumida en (mAh),  $i_i$  es la intensidad consumida por el microcontrolador en modo inactivo y  $t_i$  es el tiempo transcurrido. Del mismo modo,  $MCU_a$  es la intensidad del microcontrolador en modo activo mientras que  $MCU_i$  en modo inactivo, y  $i_{radio}$  es la intensidad consumida por el módulo de radio al realizar un envío. El sumatorio representa el consumo de diferentes sensores  $j$  durante el tiempo de operación de éstos realizando medidas de distintos datos.

El sensor de ultrasonidos genera un consumo máximo de 30 mA en funcionamiento. Debido a sus propiedades, el máximo tiempo de operación correspondería a un contenedor vacío que implicaría un total de unos 40 ms. El sistema implementado de peso consiste en 4 células de carga conectadas a un *hub* y a un amplificador analógico digital que suministra el voltaje requerido por las células de carga para su correcto funcionamiento. Este amplificador emplea únicamente 1.5 mA mientras realiza una medida durante 40 ms. El sensor de temperatura empleado (TMP36) es un sensor analógico que consume 4 mA y puede realizar medidas cada 5ms. El módulo de radio empleado para la comunicación con los *gateways* de la red es el modelo LoRa1276-C1 de NiceRF. Este dispositivo emplea como máximo 140 mA para el envío de datos y manda la información en menos de 10 ms. Esta información de consumo energético debe ser agregada al periodo de tiempo que el microcontrolador se encuentra activo y en operación con el fin de obtener el consumo total requerido para tomar las medidas con los sensores y realizar el envío de la información.

El microcontrolador seleccionado emplea 35  $\mu$ A/MHz en funcionamiento. Como trabaja a una frecuencia de 48MHz, demanda una corriente eléctrica total de 1.68 mA. En modo activo, este sensor realiza 2 tareas: tomar las medidas de los sensores y gestionar los envíos. Para la toma de las medidas de los sensores, el tiempo que aparece en la ecuación es el tiempo máximo que emplean los sensores tomando una medida, en este caso, la realizada por el sensor de peso, es decir, un total de 40 ms. Cuando el microcontrolador se encuentra en modo inactivo, necesita una intensidad de corriente de 200 nA, por lo tanto, usará un total de 0,0002 mAh para estar en modo inactivo en una hora.

Una vez que las estimaciones se han obtenido, se procede a diseñar el módulo y se obtienen las medidas empíricas del módulo desarrollado. Las pruebas consisten en simulaciones para definir el consumo real del nodo bajo diferentes escenarios de envío y toma de datos. Esto permite especificar la mejor configuración envíos/medidas diarias que sea necesaria para el sistema de recolección de residuos y que cumpla los requisitos de autonomía de batería marcados por este.



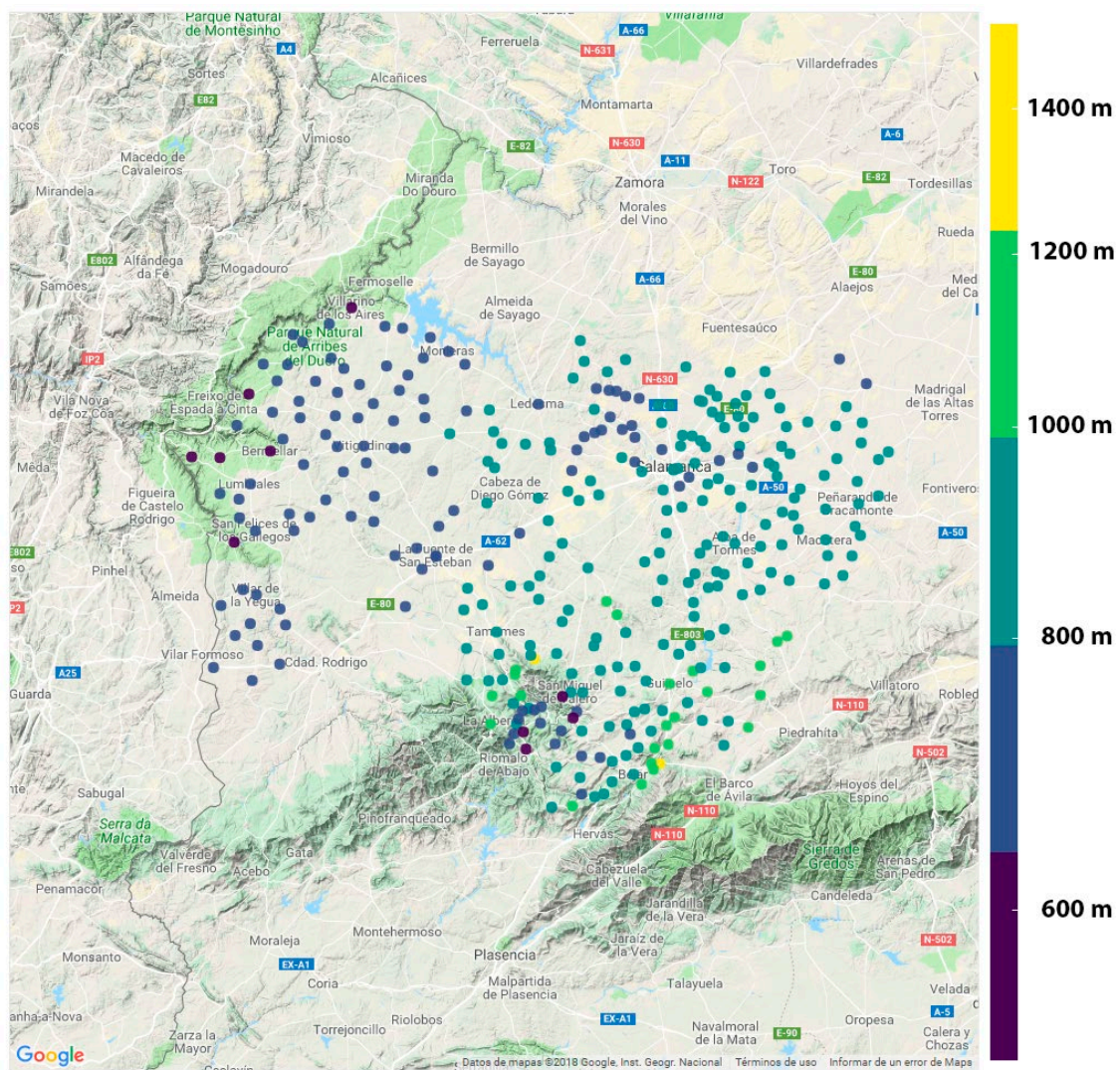
**Tabla 3** Envíos y medidas de datos de sensores por día y sus consumos teórico y empírico

Envíos/día	Medidas/día	Consumo estimado (mAh por día)	Consumo medido (mAh por día)
12	12	6,18	6,84
12	24	7,5	7,8
24	24	25,488	26,88
24	48	28,128	32,16
48	48	104,3232	107,04

La **Tabla 3** describe algunas de las posibles configuraciones de los nodos para estimar el consumo del dispositivo por día de funcionamiento. La primera columna describe el número de envíos por día realizados en intervalos de tiempo iguales. La segunda columna muestra el número de medidas tomadas en un día. La tercera columna muestra el consumo de batería estimado mediante el método expuesto con anterioridad y la cuarta columna, los consumos del dispositivo medidos en el laboratorio. Cabe resaltar que, dependiendo del Spread Factor elegido para las transmisiones LoRa y la longitud del *payload* empleado, el consumo puede variar significativamente como describen en su trabajo [Cheong et al. 2017] y [Bouguera et al. 2018]. En el trabajo de [Cheong et al. 2017] se presentan varios resultados empleando distintos SF tamaños de paquete y frecuencias de envío. Los resultados son cercanos a los estimados teóricamente y que se han obtenido en el laboratorio, siendo estos para un SF 11 y 25 bytes de *payload* 114000 mAh cada 10 años, realizando envíos cada hora, es decir, 31,78 mAh por día, muy cercano a los obtenidos en este trabajo. Como se puede comprobar en la **Tabla 3**, el consumo se ve penalizado principalmente por el número de envíos y no tanto por el número de medidas tomadas, siendo, vital por tanto el uso de una tecnología como LoRa para este tipo de sensores.

#### 4.1.2 Estudio de la cobertura de la red inalámbrica de sensores

Esta sección presenta un estudio de la cobertura de la red LoRaWAN en la región del caso de estudio, la provincia de Salamanca. La orografía ha sido extraída para identificar las mejores localizaciones con el fin de situar cada uno de los *gateways* que compondrán la red desplegada. Las mejores opciones son municipios del área con la mayor altitud y que dispongan de un relativo buen acceso a internet para reducir costes derivados de conexiones de red.

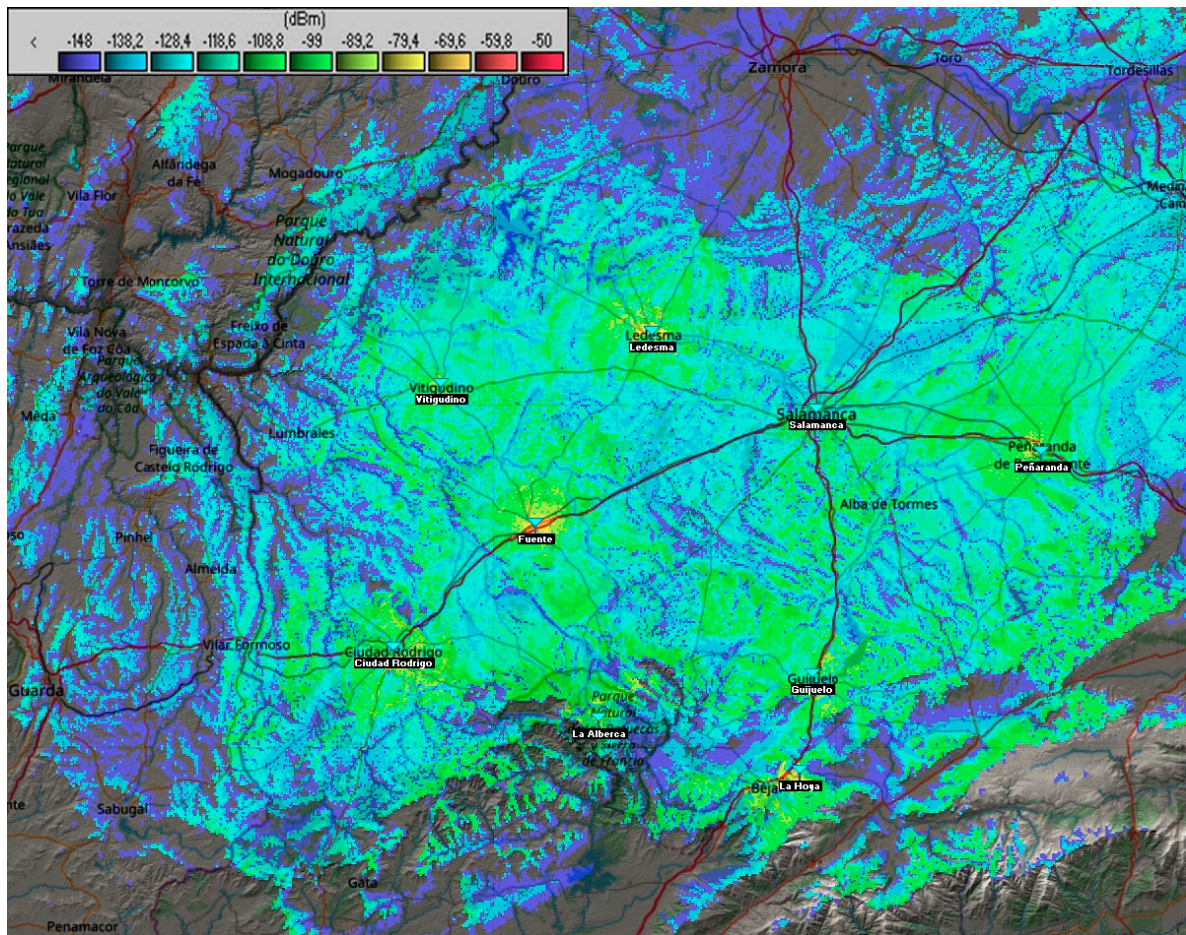


**Figura 22** Municipios del caso de estudio clasificados por su altitud

La **Figura 22** muestra la altitud en metros de cada uno de los municipios involucrados en este caso de estudio y a los que se pretende dar cobertura con esta red.

Para estudiar la cobertura se ha empleado la herramienta Radio Mobile [Cu 2014] [Hopengarten 2009], ampliamente utilizada en el estudio de cobertura en radio frecuencia. Se ha utilizado esta herramienta frente otras disponibles como CloudRF [CloudRF] porque soporta, al igual que CloudRF el modelo de propagación Longley-Rice también conocido como Irregular Terrain Model (ITM) [Kasampalis et al. 2013] y se trata de un software de uso gratuito. Basándose en lo mencionado anteriormente, se ha decidido desplegar una red en la región que consiste en 9 *gateways* dispuestas como se indica en la **Figura 23**, siendo los municipios: Vitigudino, Ciudad Rodrigo y La Fuente de San Esteban (oeste), La Alberca, La Hoya y Guijuelo (sur), Peñaranda (este) y Ledesma y Salamanca (norte).



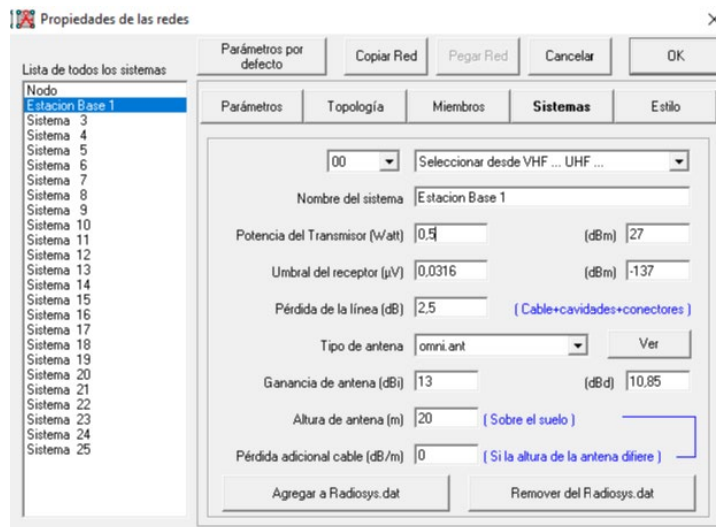


**Figura 23** Mapa de cobertura de la red propuesta realizado con la herramienta Radio Mobile

La **Figura 23** muestra la cobertura de la red en dBm en el mapa. Empleando datos del terreno proporcionados por SRTM y las propiedades de propagación de la tecnología LoRa, es posible proporcionar cobertura a todos los municipios de la región al igual que a sus alrededores situando adecuadamente 9 antenas en esta área.

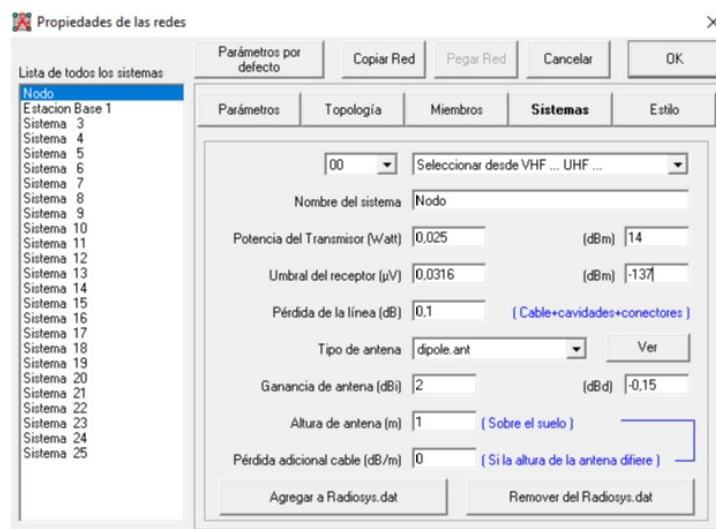
Esto hace que el coste de despliegue de la red sea realmente bajo dado que varios municipios y sus demarcaciones se encuentran cubiertos por un único *gateway*, reduciendo el coste total.

A continuación, se describen las variables empleadas en el software Radio Mobile a la hora de realizar el estudio de cobertura. Para las estaciones base (*gateways*), se establece como potencia de transmisión  $0,5\text{W}$ , un umbral de recepción de  $0,0316\ \mu\text{V}$ , una pérdida de la línea de  $2,5\text{dB}$ , una antena omnidireccional, la ganancia de la antena de  $13\text{dBi}$  y situada a una altura de 20 metros. Estos datos se presentan en la **Figura 24**.



**Figura 24** Configuración de las estaciones base en la herramienta Radio Mobile

Para los nodos se ha establecido una potencia de transmisión de 0,025W, un umbral de recepción de 0,0316  $\mu\text{V}$ , una pérdida de línea de 0,1 dB, una antena dipolo una ganancia de 2dBi y una altura de 1 metro sobre el suelo. Estos datos se pueden observar en la **Figura 25**.



**Figura 25** Configuración de los nodos en la herramienta Radio Mobile

Estos parámetros se han establecido de acuerdo con la recomendación ERC-REC-70-3E [Committee 2015] que sigue la regulación Europea relativa al espectro de frecuencia en la banda de los 868MHz, que es utilizado por múltiples Low Power Networks (LPWAN).

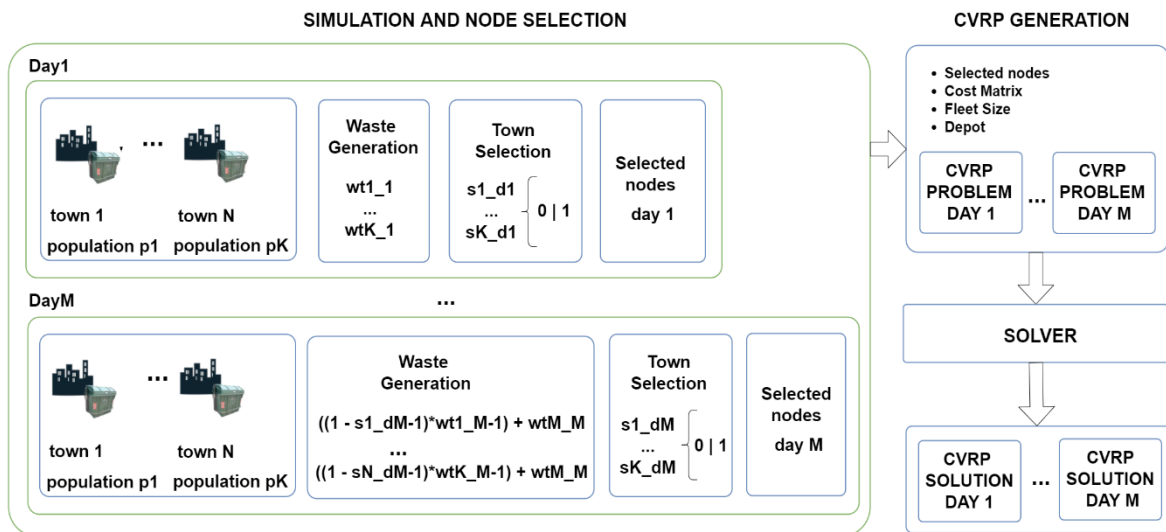
El mapa de cobertura mostrado en la **Figura 23** es una vista de los mapas de cobertura cartesianos para cada uno de los *gateways* donde la transmisión es especificada por las características del nodo.

Tras llevar a cabo el estudio teórico, es necesario realizar varios tests de cobertura en áreas con baja cobertura para comprobar cómo se ajusta el modelo teórico al entorno real para un futuro despliegue. Este estudio demuestra teóricamente la posibilidad del despliegue de una red de sensores que, además puede ser utilizada para otros casos de estudio, empleando sensores en ámbitos muy distintos, por ejemplo, *Smart Farming* o Agricultura de Precisión en zonas rurales. Dotar a una región con este tipo de infraestructura promueve el desarrollo de nuevas soluciones y aplicaciones, empleando sensores que usen esta tecnología. Dado que se trata de una red basada en un servicio como *The Thing Network*, habilita a otros usuarios su utilización y, de la misma forma, a los nodos desplegados emplear otros futuros *gateways* desplegados bajo *The Thing Network*, empleando un esquema colaborativo.

### 4.1.3 Resultados del sistema de optimización de rutas

Esta sección describe una simulación del proceso de optimización de rutas usando como base los datos de (1) producción de desechos del plan de residuos de la administración de Castilla y León [Gobierno 2001a], (2) el documento técnico público del proceso de licitación de 2016 de la recogida de desechos de la Diputación de Salamanca y (3) un informe acerca de la recolección selectiva de papel de Ecoembes [Ecoembes 2008] que será el tipo de residuo elegido en este caso de estudio. Los escenarios de recolección de residuos han sido generados para un año con un total de 26 días de recolección (1 recogida cada 14 días) que supone el tiempo máximo que establecen como recogida. Al no disponer de los datos del número de recogidas que realiza la empresa adjudicataria, se ha decidido fijar ese límite inferior de rutas para poder evaluar los ahorros en el peor de los casos. Estos escenarios de recogida han sido generados teniendo en cuenta la población de cada municipio y la media de producción de residuos por habitante y día de cada uno de ellos.

En la simulación se asume que la visita a un municipio implica la recogida de todos los contenedores de este, por tanto, en este caso de estudio, el desplazamiento realizado por cada camión en cada municipio no penaliza el total de kilómetros realizados en la ruta.



**Figura 26** Simulación de la generación de residuos, selección de nodos, instancias CVRP y sus soluciones.

La **Figura 26** muestra en la parte izquierda el proceso de simulación y selección de los municipios. Inicialmente, se obtiene la población de todos los municipios. Después de esto, se realiza una generación de residuos para cada uno de los municipios, extrayendo un valor aleatorio de desechos. Este valor sigue una distribución normal [Peebles 1987] considerando los desechos generados por habitante empleando la ratio extraído de los documentos previamente mencionados, los desechos generados descritos siguen la ecuación (13):

$$[W_{t11}, \dots, W_{tK1}] = \text{randomNormalValue}(\mu = \text{townwaste}_t, \sigma = \text{townwaste}_t/10) \quad (13)$$

Donde  $[W_{t11}, \dots, W_{tK1}]$  es la generación de residuos de los  $K$  municipios para el día de recolección 1 y  $\text{townwaste}_t$  es la generación de residuos del municipio que corresponde a  $\text{townwaste}_t = \text{inhabitantWasteRatio} * \text{daysBetweenCollection} * \text{population}_t$ .

Posteriormente, se realiza un proceso de selección de nodos para incluir aquellos municipios que se van a considerar para las siguientes rutas de recolección. Este enfoque cambia respecto al enfoque estático de recolección ya que se seleccionan solo aquellos nodos (municipios) que realmente necesitan la recogida. En esta simulación, la variable que dicta la selección de nodos como umbral de recogida es el peso ya que se ha empleado un volumen constante para el papel y el cartón, sin embargo, en un entorno real de producción la selección de nodos involucraría tanto las medidas de volumen como las de peso, empleando en primer lugar el volumen para la selección y, posteriormente el peso para el cálculo de las rutas. Cabe destacar que el volumen puede variar mucho en un entorno real de recogida de papel y cartón.

En este trabajo, como se ha mencionado anteriormente, se ha establecido una densidad del papel de 46,47 kg/m<sup>3</sup> [Ecoembes 2008] para obtener el volumen y es la que ha sido utilizada a la hora de realizar los cálculos. El criterio de selección de los nodos viene dado por la ecuación (14):

$$\text{CurrentPopulationWaste} + \text{PopulationWasteUntilNextDay} \geq 0,80 * \text{PopulationWasteCapacity} \quad (14)$$

Donde *CurrentPopulationWaste* es la demanda total hasta el día actual y *PopulationWasteUntilNextDay* una nueva generación de residuos para los próximos días hasta el siguiente día de recolección. Si este valor es superior al 80% de la capacidad total de los contenedores del municipio, expresada como *PopulationWasteCapacity*, ese municipio será seleccionado para la recolección. Este porcentaje ha sido extraído de la licitación pública de la Diputación de Salamanca [Diputación de Salamanca]. *PopulationWasteCapacity* se calcula a partir de la población del municipio y del número de contenedores disponible para dicho municipio.

Este proceso se realiza para cada día de recolección. El paso de generación de residuos se lleva a cabo considerando la previa selección de nodos para añadir la demanda de residuos remanente pendiente en los nodos que no fueron visitados con anterioridad.

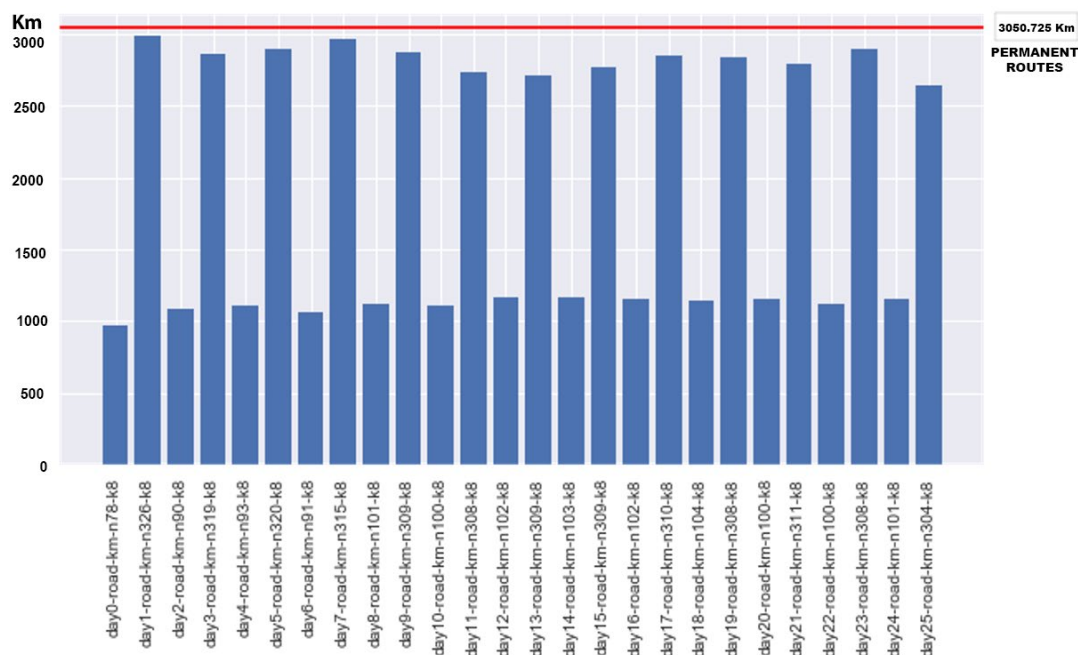
Como resultado de este proceso de simulación se obtienen los municipios seleccionados y sus demandas de recogida de residuos para los M días planteados. El siguiente paso es generar las instancias de los problemas CVRP empleando los nodos seleccionados, la flota de vehículos disponible y la matriz de distancias de los municipios seleccionados.

Una vez obtenidas las instancias de los problemas CVRP, estas son enviadas al *solver* para obtener las rutas de recogida para dicho problema. La etapa de resolución emplea diferentes heurísticas, métodos de búsqueda local de forma paralela, para encontrar una solución factible en un tiempo limitado de 24 horas.

Los resultados de los 26 días de recolección de desechos relativos a un año se presentan en la **Figura 27**. Se expresa la distancia en km para cada una de las instancias CVRP generadas y sus respectivos costes frente al enfoque estático de recolección.

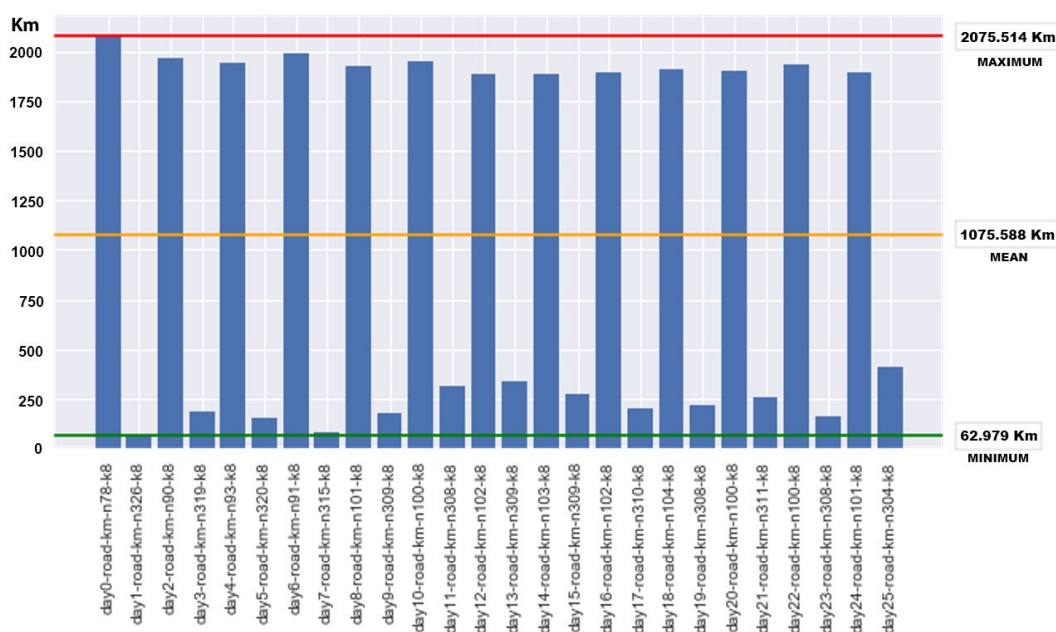
Como se observa en la **Figura 27**, las rutas estáticas representan un coste total de 3050,725 km (especificado por la línea horizontal roja) en cada día de recolección de desechos mientras que, en las rutas optimizadas, como se esperaba, existen ahorros en el coste total de la distancia en todos los casos.





**Figura 27** Resultados de cada día de recolección frente al coste total fijo de rutas estáticas

En la **Figura 28** se muestran los ahorros producidos para cada instancia, con un ahorro máximo de 2075,514 km (línea horizontal superior roja), un mínimo de 62,979 km (línea inferior verde) y un ahorro medio de 1075,588 km (línea naranja) para cada instancia, lo que supone un ahorro medio de un 28%. Cabe destacar que el patrón que se observa en los costes viene directamente generado, por un lado, por los nodos que no se seleccionan en recogidas previas y, por otro, por el número mínimo de recogidas que se deben realizar en los municipios cada cierto tiempo.



**Figura 28** Ahorros para cada día de recolección



#### 4.1.4 Conclusiones

El caso de estudio descrito para la recolección de desechos en la región de Salamanca muestra la mejora proporcionada por el sistema de obtención de datos y de optimización de rutas propuesto frente a un enfoque clásico estático de recolección. Ofrece mejoras en la distancia recorrida, consumo de combustible, uso de la flota de vehículos, ahorro en mano de obra y obtención de datos útiles para la predicción de generación de basura.

En lo relativo al sensor desarrollado, presenta un bajo consumo y una autonomía energética suficiente para el funcionamiento en el caso de uso propuesto en este estudio. El sensor obtiene las medidas y envía los datos a través de la red inalámbrica de sensores propuesta sin que esto penalice la autonomía, gracias a la tecnología de radio empleada en el diseño de la red.

El despliegue de este sensor permite monitorizar los contenedores que lo porten, así como obtener información acerca de la generación de basuras de los habitantes en cada municipio y a nivel de cada vecindario. Esto puede facilitar la detección de la necesidad de desplegar más contenedores en una determinada zona en la que se generan más residuos debido a locales comerciales o industriales cercanos o, incluso, identificar lugares en los que la población crece o decrece en determinadas épocas del año.

De acuerdo con el estudio de cobertura realizado, la utilización de 9 estaciones base para la red de sensores es suficiente para dar cobertura a la provincia de Salamanca y cubrir las necesidades del caso de estudio planteado u otros casos de estudio empleando esta tecnología en la zona. Esto significa un coste reducido en la infraestructura que dota de cobertura a la red ya que un mismo *gateway* cubre grandes áreas de terreno.

Respecto a los resultados obtenidos tras la optimización de rutas, la simulación pone de manifiesto un ahorro medio en distancia del 28% en el caso analizado. Este ahorro no solo afectaría a la distancia recorrida por la flota de camiones, sino que supone ahorro en el uso de la flota de camiones con la reducción de mano de obra necesaria que ello conlleva.

Finalmente, el proceso de simulación de recogida de papel y cartón con los datos obtenidos muestra que, realizando una recogida cada 2 semanas, no se ajusta a la generación, de papel y cartón que presenta cada municipio en la provincia. En algunos municipios la recogida se encuentra sobredimensionada y otros, todo lo contrario, por lo tanto, es necesario seleccionar la frecuencia adecuada para el ahorro de tiempo en esta tarea. La utilización de un sistema que genera las rutas dinámicamente, empleando información actualizada de los contenedores proporciona una herramienta para la toma de decisiones en la recogida de residuos que ayuda a seleccionar de forma automática la frecuencia de recogida necesaria para cada municipio, ahorrando costes.

---

Para concluir, después de realizar este caso de estudio, el sistema propuesto de toma de datos y optimización de rutas muestra un nodo que puede operar eficientemente con una autonomía de más de un año, obteniendo información útil como peso, volumen y temperatura de los contenedores gracias a su bajo consumo. Además, el estudio de cobertura realizado muestra que con una reducida inversión en infraestructura de *gateways* es posible dar cobertura a la provincia de Salamanca. Junto con estos resultados, la optimización de rutas, seleccionando los nodos de forma dinámica y calculando la mejor ruta en cada escenario, proporciona ahorros en costes de desplazamiento, tiempo y mano de obra frente a un enfoque de rutas estáticas.

Los ahorros potenciales del Smart Waste Collection System y las ventajas que supone este tipo de WSNs podrían permitir la inversión en la requerida infraestructura, así como la incorporación de los sensores a la flota de contenedores paulatinamente. Esto supondría una inversión en infraestructura reutilizable por otros proyectos IoT en la región, impulsando el potencial desarrollo tecnológico en la zona.

## 4.2 Predicción de la demanda en sistemas de Bike-Sharing

En este caso de estudio se busca explorar la parte del sistema propuesto encargada de la predicción de la demanda. Se ha planteado realizar la predicción de la demanda en un problema de logística que presenta aspectos comunes con el problema de recolección de residuos. Se trata de la recogida de bicicletas en un sistema de alquiler de estas o también conocido en inglés como *Bike Sharing System* BSS.

En estos sistemas los usuarios toman bicicletas de las distintas estaciones repartidas por la ciudad para realizar un trayecto a otra estación de este. El principal problema existente es la cantidad de bicis disponibles (para que sean alquiladas por los usuarios) y espacios libres (para que sean depositadas las bicicletas) en cada estación a lo largo de la jornada. Por tanto, puede darse la situación en la que un usuario no pueda coger una bicicleta porque no haya suficientes disponibles o, por otro lado, que no pueda depositarla en la estación más cercana a su destino. Por esta razón, es necesario balancear la cantidad de bicis en cada estación, tarea que es realizada por los operarios del sistema a lo largo de la jornada. Para esta tarea es necesario predecir el estado del sistema, basándose en datos históricos y predecir la demanda de las estaciones (ya sea alquiler de bicicletas o depósito de estas) a lo largo del día. Una vez conocida esta información es posible plantear un problema de rebalanceo de bicicletas con una flota de vehículos (Static Bike Relocation Problem SBRP)[Chemla et al. 2013] [Erdoğan et al. 2014] que puede formularse como una formulación del problema como un Pickup and Delivery Vehicle Routing Problems (PDVRPs) dentro de la gran familia de problemas VRP como se indica en [Dell'Amico et al. 2014]. En este caso de estudio se aborda únicamente la parte relativa a la predicción de la demanda en cada una de las estaciones.

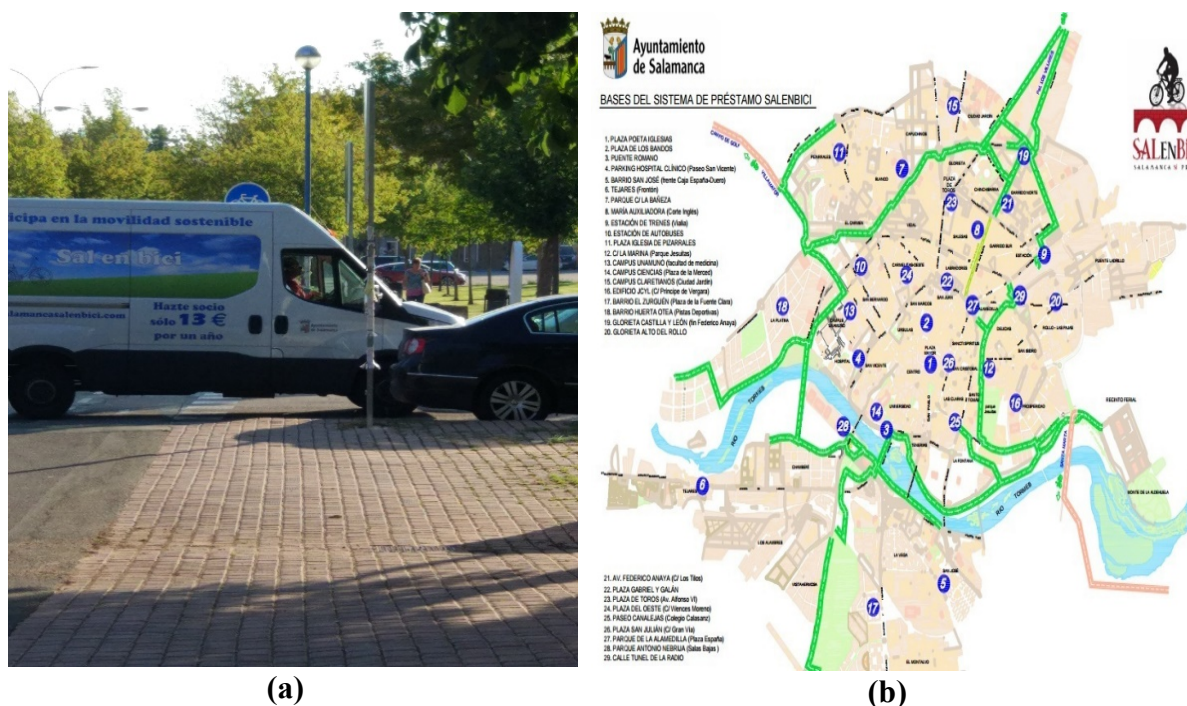
Este trabajo de tesis está relacionado con el proyecto MOVIURBAN [MOVIURBAN] en el que una tarea busca la creación de un sistema para la gestión de flotas urbanas como las pertenecientes a un sistema compartido de bicicletas. Gracias a este proyecto ha sido posible acceder a los datos relativos a los viajes realizados en el *Bike Sharing System* de la ciudad de Salamanca.

Estos datos han sido empleados para realizar el presente caso de estudio con el objetivo de crear un sistema que los obtenga de distintas fuentes y los fusione con los del sistema de alquiler de bicicletas para crear modelos de predicción de la demanda en cada una de las estaciones.

En este caso de estudio se ha podido realizar un sistema que emplea distintos modelos de predicción en su funcionamiento y que pueden ser aplicados en el sistema propuesto en este trabajo de tesis. De este modo, se pretende profundizar en la metodología del sistema propuesto a la hora de emplear modelos de predicción de la demanda con datos reales de un sistema de movilidad urbana.

### 4.2.1 Sistema de alquiler de bicicletas de Salamanca: SalenBici

El sistema de alquiler de bicicletas SalenBici **Figura 29** se encuentra situado en Salamanca, una ciudad de tamaño medio con una población de 144.949 habitantes de acuerdo con el último censo de 2016. Este sistema tiene hoy 29 estaciones y 176 bicis a lo largo de la ciudad. Las horas de funcionamiento son: días laborables de 7:00 a 22:00 y fines de semana y días festivos de 10:00 a 22:00.



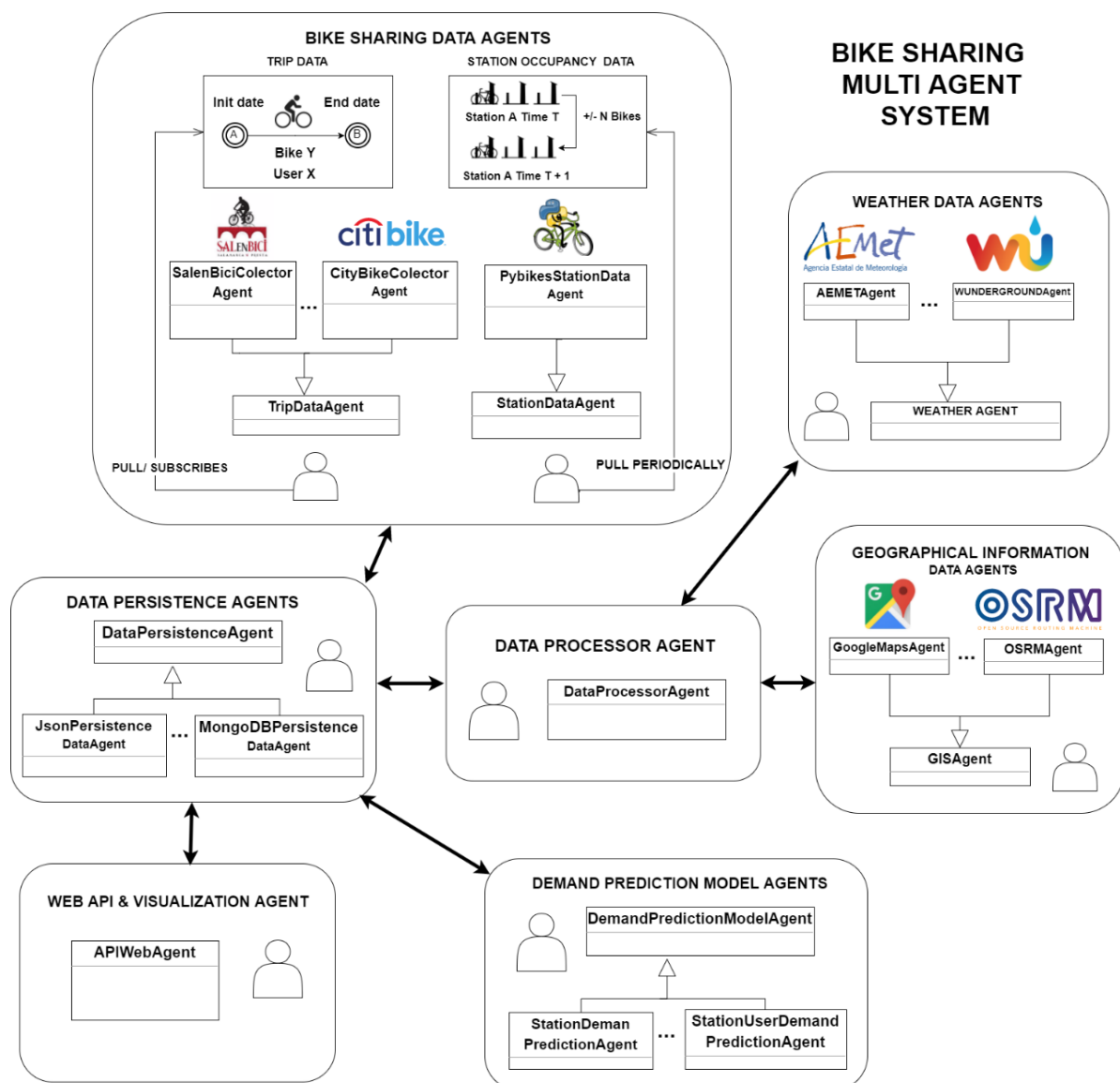
**Figura 29** (a) Uno de los camiones de balanceo de bicicletas de SalenBici. (b) Mapa situación de las estaciones del sistema facilitado por el ayuntamiento de Salamanca

Los operarios del sistema realizan tareas de reubicación y balanceo a primera hora de la mañana, a mediodía y al finalizar el día, por tanto, una información sobre la demanda esperada para mañanas y tardes podría ser especialmente útil para realizar las tareas de ubicación.

### 4.2.2 MAS de predicción de la demanda y visualización de rutas en BSS

Para obtener la información de distintas fuentes heterogéneas de datos se ha diseñado un sistema de obtención de datos, predicción de la demanda y visualización de estos siguiendo un paradigma de Sistema Multi-Agente (Multi Agent System MAS). Para su implementación se han valorado distintas plataformas como SPADE [Gregori et al. 2006], JADE [Bellifemine et al.], PANGAEA [Zato et al. 2012], AIOMAS [Aiomas] y osBrain [OsBrain]. Esta última ha sido seleccionada como la idónea debido principalmente a su facilidad de uso. Además, es una

plataforma implementada en Python, como SPADE y AIOMAS, lo que permite emplear de forma sencilla *frameworks* como Scikit Learn y su desarrollo es el más activo frente a estas dos plataformas. En él cada agente del sistema tendrá una tarea específica que debe desempeñar y podrá comunicarse (o no) con el resto de los agentes del sistema para llevarla a cabo. El sistema está dividido en 4 grupos de agentes: Bike Sharing Data Agents, Weather Data Agents, Geographic Data Agents, Data Processor Agents, Demand prediction Agents y Web API Agents que se muestran en la **Figura 30**.



**Figura 30** Diagrama general del sistema Multi-Agente propuesto

- **Bike Sharing Data Agents:** encargados de la obtención de los datos del sistema de alquiler de bicicletas. Los datos que pueden obtener estos agentes son principalmente de dos tipos:
  - **Información a nivel de estación:** información relativa al número de espacios libres en cada estación, así como al número de bicis disponibles en cada momento. Esta información se suele proporcionar a los usuarios finales para que conozcan el estado de las estaciones cercanas y si hay bicicletas libres para coger o espacios libres para depositar la que llevan. Se proveen mediante una API en tiempo real de un proyecto llamado PyBikes [Esquerda; Lluís Esquerda] que unifica la información de múltiples BSS a lo largo del mundo.
  - **Información de viajes:** este tipo de dato no suele ser proporcionado públicamente por los BSS, sino que mensualmente realizan una publicación de los viajes realizados anonimizando a los usuarios del sistema.
- **Weather Data Agents:** agentes encargados de la obtención de los datos relacionados con el tiempo climatológico que sean requeridos por el agente encargado de pre-procesar los datos (*Data Processor Agent*).
- **Geographic Information Data Agents:** estos agentes obtienen información relativa a la situación geográfica de las estaciones, la altitud de cada una de ellas y las distancias existentes entre cada una de ellas. Suministra información para la visualización geográfica del sistema.
- **Data Processor Agents:** encargados del procesado de la información proveniente del BSS y de su fusión con el resto de las fuentes de información, como datos relativos al calendario laboral y festividades e información meteorológica proporcionada por los *Weather Data Agents*.
- **Demand Prediction Agents:** agentes de predicción de la demanda en cada estación. Tienen como principal tarea, por un lado, generar los modelos de predicción a partir de los datos históricos ya pre-procesados proporcionados por el Processor Agent y, por otro, generar la predicción para una determinada estación a una determinada hora. Estos agentes se ejecutan periódicamente para producir nuevos modelos incorporando los últimos datos recibidos del BSS. Estos modelos se generan siguiendo la metodología propuesta en este trabajo de tesis y se emplean los siguientes modelos de regresión ya que han ofrecido buenos resultados en estudios previos como [Regue and Recker 2014] [Malani et al.] [Prakash Nekkanti 2017] [Wang et al. 2016]. Los algoritmos empleados son Random Forest Regressor [Breiman 2001], Gradient Boosting Regressor [Friedman and Friedman 2000b] and Extra Tree Regressor, disponibles en el *framework* Scikit-learn [Pedregosa FABIANPEDREGOSA et al. 2011].
- **WEBAPI Agent:** este agente tiene la tarea de ofrecer acceso a datos del BSS a través de una API que pueda ser consumida por una aplicación web. La aplicación web proporciona la siguiente información:

- **Visualización de datos históricos:** esta sección mostrará información acerca de los datos históricos permitiendo al usuario realizar un filtrado sobre ellos.
- **Predicción y estado actual del sistema:** esta sección muestra el estado actual del sistema y las predicciones de demanda para cada estación.

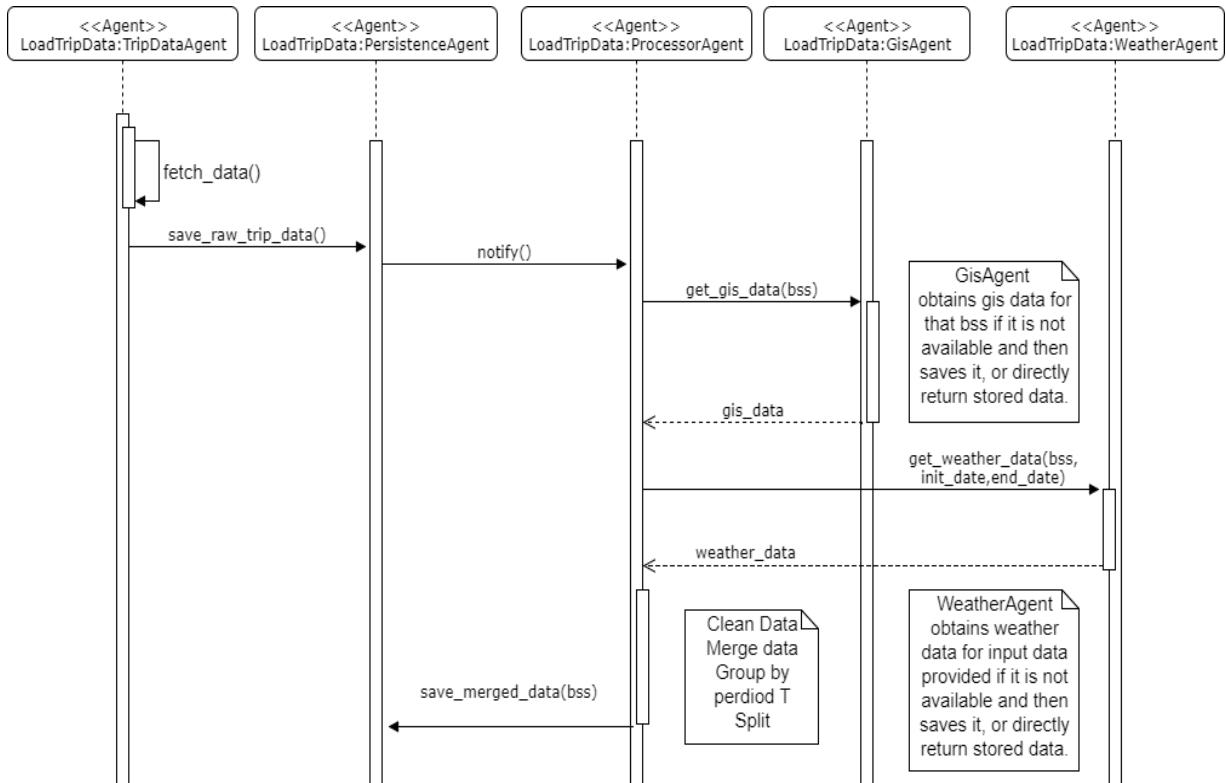
### 4.2.3 Proceso de obtención de datos

La información obtenida del sistema de alquiler de bicicletas SalenBici consiste en todos los viajes efectuados en el sistema desde una estación origen a una estación destino, por lo tanto, la información en este caso es del tipo mencionado anteriormente como *trip data* o viajes realizados de una estación a otra.

La información posee los siguientes campos: (1) Time Start: marca de tiempo de inicio del viaje, (2) Time End: marca de tiempo del final de viaje, (3) Bicycle ID: identificador único de la bicicleta, (4) Origin Station: estación origen, (5) End Station: estación destino, (6) Origin Dock: candado de origen, (7) End Dock: candado destino, (8) User ID: identificador único de usuario.

Esta información ha sido proporcionada por la compañía SalenBici en forma de ficheros CSV para cada año que han sido cargados por el sistema Multi-Agente para su procesamiento y posterior uso. El agente *Trip Agent* realiza la carga de todos los datos de la fuente especificada, en este caso, ficheros CSV, posteriormente el *Processor Agent* es notificado por el *Persistence Agent* ya que nuevos datos en crudo han sido agregados al sistema. El *Processor Agent* comprueba los datos obtenidos y solicita la información geográfica y meteorológica a los agentes GIS y *Weather Agent*, respectivamente. Estos agentes, cuando se solicita, comprueban la disponibilidad de los datos con el *Persistence Agent* antes de devolver una respuesta; si los datos están disponibles, se devuelven inmediatamente, en caso contrario, se recopila toda la información, se guarda y se devuelve una vez almacenada. En la **Figura 31** se muestra un diagrama de secuencia del proceso de obtención de datos y la comunicación entre los distintos agentes (la comunicación entre el *Persistence Agent* y *GIS Agent* se ha omitido para clarificar el proceso).

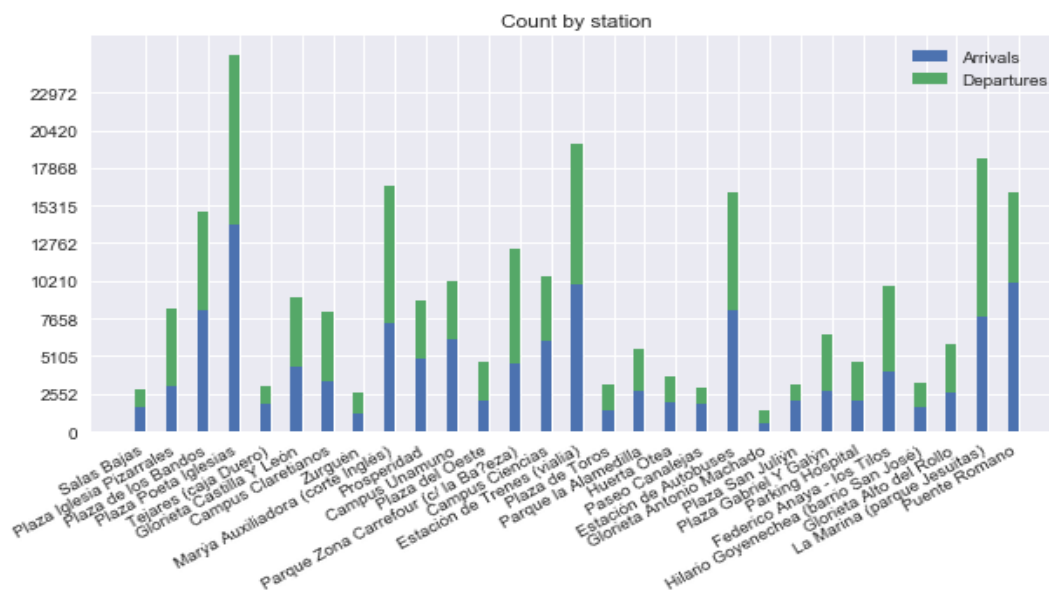
Después de este proceso, los datos resultantes se clasifican en llegadas y salidas de cada estación con los datos relativos al calendario (año, mes, día de la semana, día del mes, semana del año, hora, estación —invierno, primavera, verano u otoño—), fin de semana, día festivo y los relativos a la información climatológica (temperatura mínima, máxima y media en grados Celsius, la velocidad del viento media en km/h, la presión atmosférica mínima y máxima en milibares y por último la cantidad de lluvia del día). Como variable dependiente se especifica la cantidad total de llegadas y salidas agrupadas por un periodo de tiempo T.



**Figura 31** Diagrama de secuencia que representa la carga de datos *trip data* en el sistema multi-agente

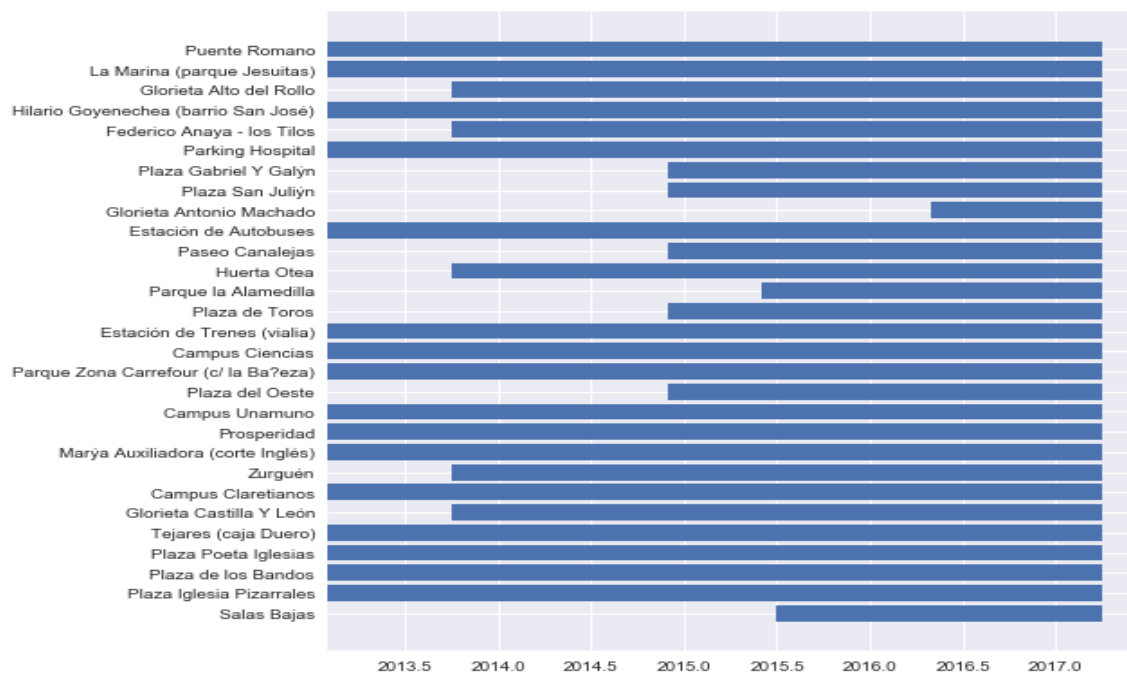
Antes de desplegar el *Predictor Agent* en el sistema multi-agente, se ha realizado un análisis exploratorio previo de los datos. Este análisis ha determinado qué modelos incluir en el *Predictor Agent*. La **Figura 32** muestra la cantidad total de viajes cargados en el sistema, divididos en llegadas y salidas para cada estación en el sistema. Los datos pertenecen al periodo desde enero de 2013 hasta marzo de 2017 con un total de 1520 días y es posible percibir una diferencia clara en el uso de las distintas estaciones del sistema. Existe un elevado número de estaciones donde el número medio de viajes realizados es menos de 2 al día (salidas o llegadas) y solo una estación, llamada “Plaza Poeta Iglesias”, cerca de la Plaza Mayor de Salamanca (uno de los lugares más turísticos de la ciudad) que tiene una media superior a 15 viajes por día (contando tanto llegadas como salidas).





**Figura 32** Número total de eventos (salidas o llegadas) realizados en BSS SalenBici desde enero de 2013 hasta marzo de 2017

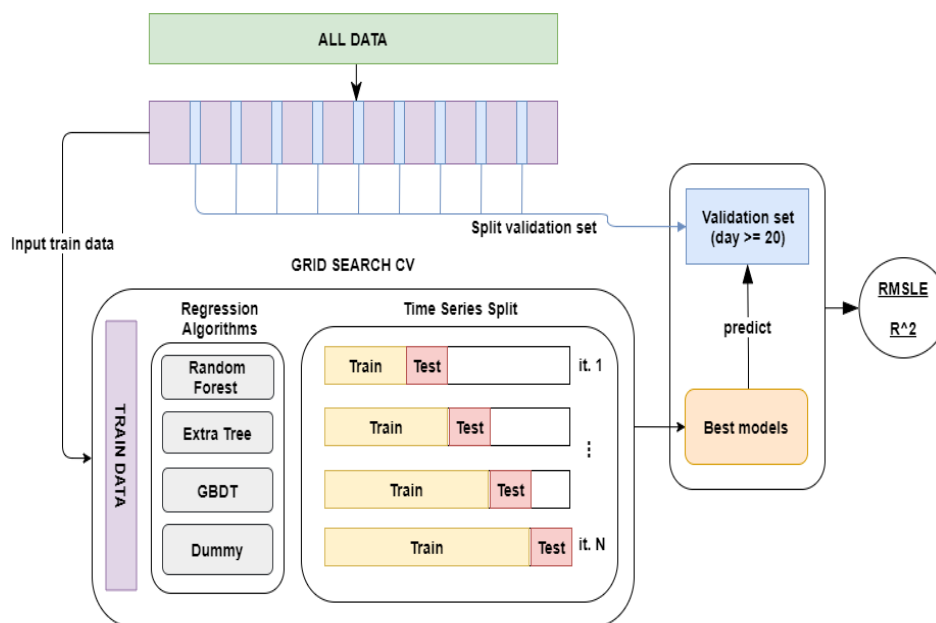
Además, la **Figura 33** muestra el periodo de actividad de cada una de las estaciones en el sistema. No todas las estaciones han estado funcionando desde el principio (enero de 2013) y es importante tener esto en cuenta a la hora de analizar el número total de viajes mostrados en la **Figura 32**.



**Figura 33** Periodos de tiempo de funcionamiento de cada estación en el BSS SalenBici

#### 4.2.4 Selección de modelos de predicción

En esta sección se detalla cómo se ha separado el conjunto de datos del BSS y la metodología que se ha seguido para seleccionar los modelos de predicción que se incluirán en el Predictor Agent. Se han separado inicialmente los datos en un conjunto de entrenamiento y otro de validación. La **Figura 34** muestra de forma esquematizada cómo se han empleado los datos disponibles para realizar el entrenamiento, selección y validación de los modelos empleados. En la parte superior de dicha figura, en verde, se representa el conjunto de datos total. De este conjunto se extrae inicialmente un conjunto de validación, al igual que se hace en la competición de Kaggle [Kaggle INC 2014], que estará formado por los días 20 en adelante de cada mes, representado en la figura de color azul. Por otro lado, el conjunto de datos restante (días del 1 al 19 de cada mes) se utilizará como datos de entrenamiento, representado en color violeta en la figura mencionada. Estos datos serán una de las entradas que tendrá la técnica de búsqueda de hiperparámetros empleada: *GridSearchCV* [Bergstra et al.]. Esta técnica tendrá como entrada: (1) los algoritmos de regresión empleados con su correspondiente *grid* de parámetros, (2) una función de *scoring*, para evaluar los modelos entrenados, en este caso RMSLE y R<sup>2</sup>, y, por último (3) un método de *cross validation*, en este caso *TimeSeriesSplit*, que consiste en un método especialmente indicado para series temporales. Este se aproxima más al funcionamiento habitual del sistema en producción, ya hace posible entrenar con datos del pasado y validar con datos del futuro progresivamente. Se puede observar un esquema de su funcionamiento en la parte inferior de la **Figura 34**, incluido en el *GridSearchCV*.



**Figura 34** Esquema de selección de hiperparámetros y entrenamiento aplicado al caso de estudio de BSS

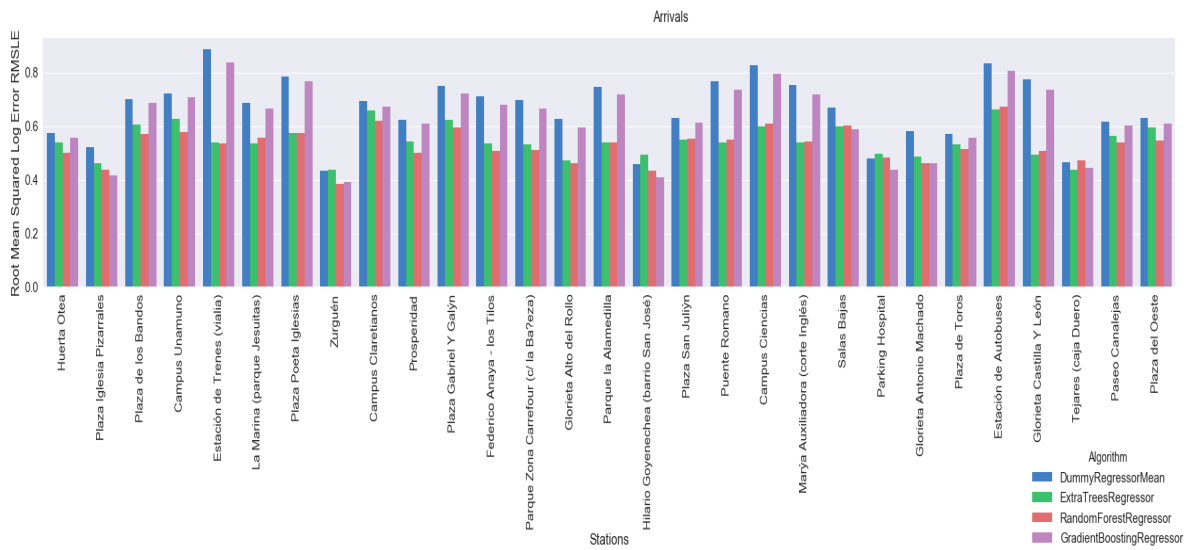
El método *GridSearch* realizará todas las combinaciones posibles para cada algoritmo con las *grids* de parámetros facilitados, empleando el método de *cross validation* proporcionado (Time Series Split) y evaluando los modelos entrenados con las funciones de *scoring* provistas. Como salida de esta se obtendrán los modelos para cada algoritmo con mejores resultados y estos evaluarán con el conjunto de validación separado inicialmente (parte derecha de la **Figura 34**). Se ha añadido un *Dummy Regressor* junto a los modelos empleados y se ha establecido como su estrategia de predicción la de predecir continuamente la media. Los algoritmos de regresión que se han entrenado utilizando *GridSearchCV* y sus *grids* de parámetros han sido:

- Extra Tree Regressor: criterion: [mae, mse], number estimators: [10, 100, 1000], max features: [auto, sqrt, log2]
- Random Forest Regressor: criterion: [mae, mse], number estimators: [10, 100, 1000], max features: [auto, sqrt, log2]
- Gradient Boosting Regressor: learning rate: [0.1, 0.01, 0.001], subsample: [1.0, 0.9, 0.8], max depth: [3, 5, 7], min samples leaf: [1, 3, 5]

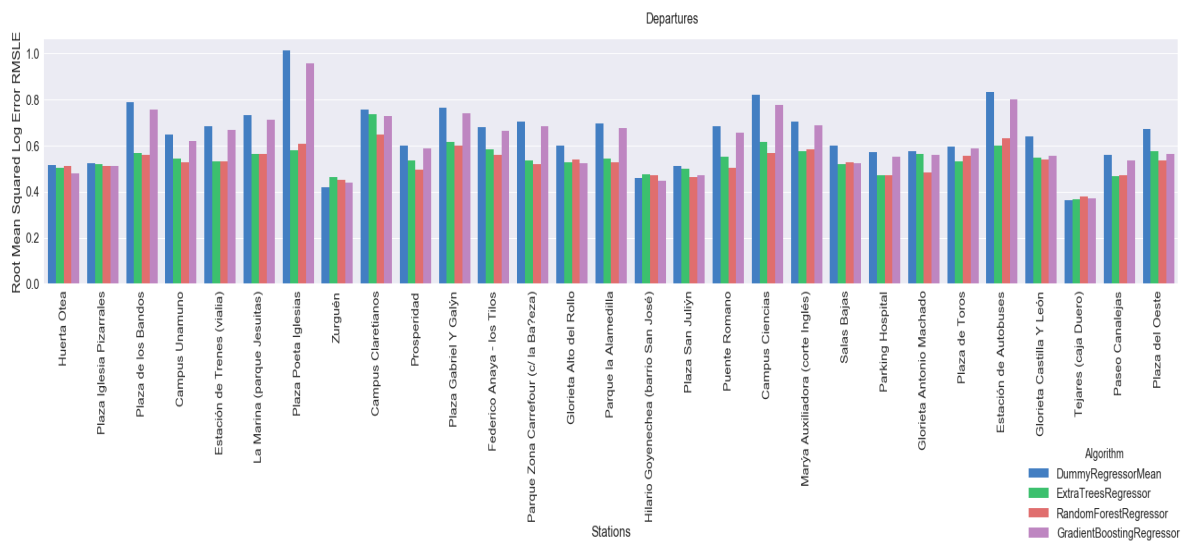
Una vez seleccionado el modelo, se validan los resultados del mejor modelo seleccionado con el conjunto de datos de validación separado previamente. Se ha calculado adicionalmente el coeficiente de determinación  $R^2$  de los modelos seleccionados para cada estación. Se han entrenado de forma separada modelos para calcular las llegadas y las salidas de cada estación.

#### 4.2.5 Discusión de resultados obtenidos

A continuación, se muestran los resultados del RMSLE en el conjunto de validación. Se detallan los resultados de los mejores modelos seleccionados por el *GridSearchCV* para cada estación. Se presentan en la **Figura 35** los resultados para las llegadas y en la **Figura 36** los resultados para las salidas. En las gráficas se puede observar cómo, en general, los regresores obtienen mejores resultados que el *baseline* establecido. Destacan el *Random Forest Regressor* y el *Extra Tree Regressor* con los mejores resultados.



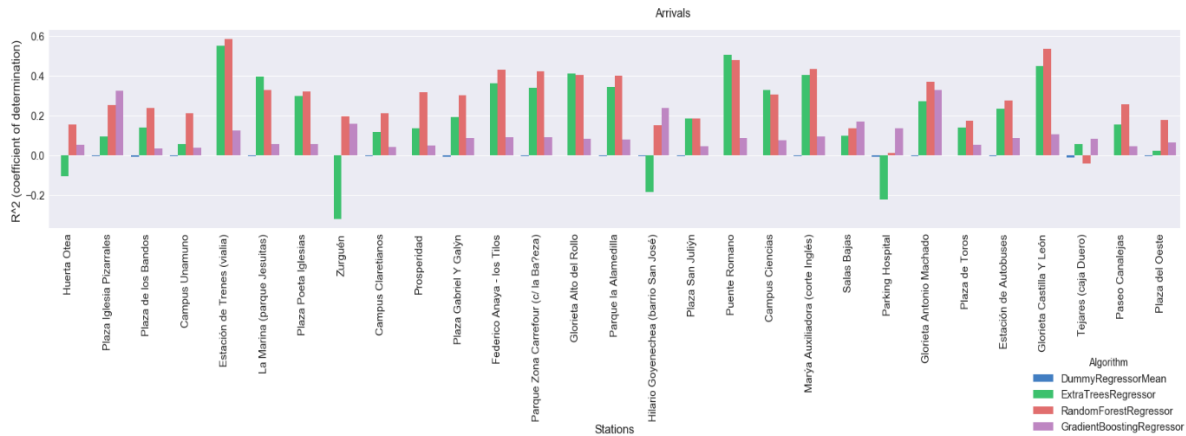
**Figura 35** RMSLE obtenido para cada algoritmo de cada estación en los datos de llegadas



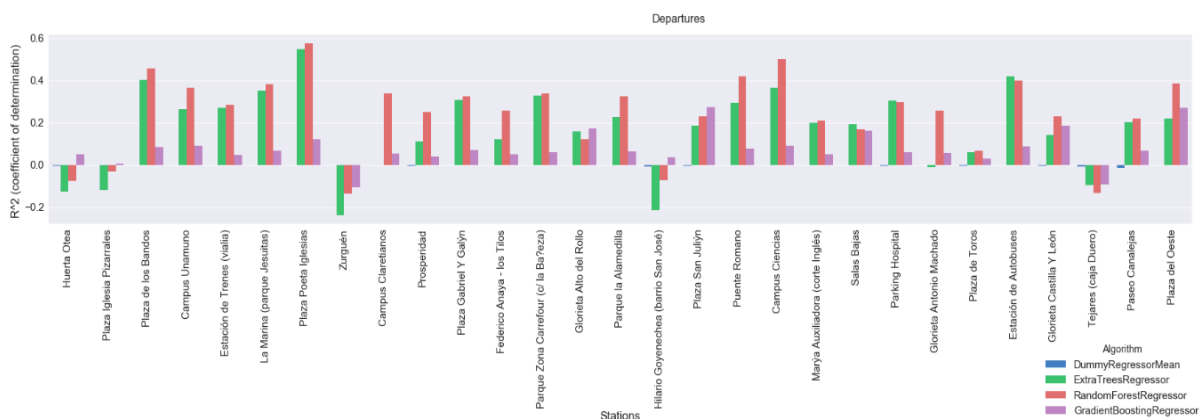
**Figura 36** RMSLE obtenido para cada algoritmo de cada estación en los datos de salidas

Adicionalmente, se ha calculado el coeficiente de terminación  $R^2$  para cada uno de los algoritmos con el fin de emplearlo para seleccionar el modelo que se incluirá en el *Predictor Agent*. Se muestran los resultados del coeficiente de determinación para cada uno de los algoritmos en cada una de las estaciones, tanto para llegadas en la **Figura 37** como para salidas

en la **Figura 38**. Gráficamente se puede ver que el método con  $R^2$  mayor es Radom Forest Regresor no obstante se analizará si esta diferencia se considera significativa.

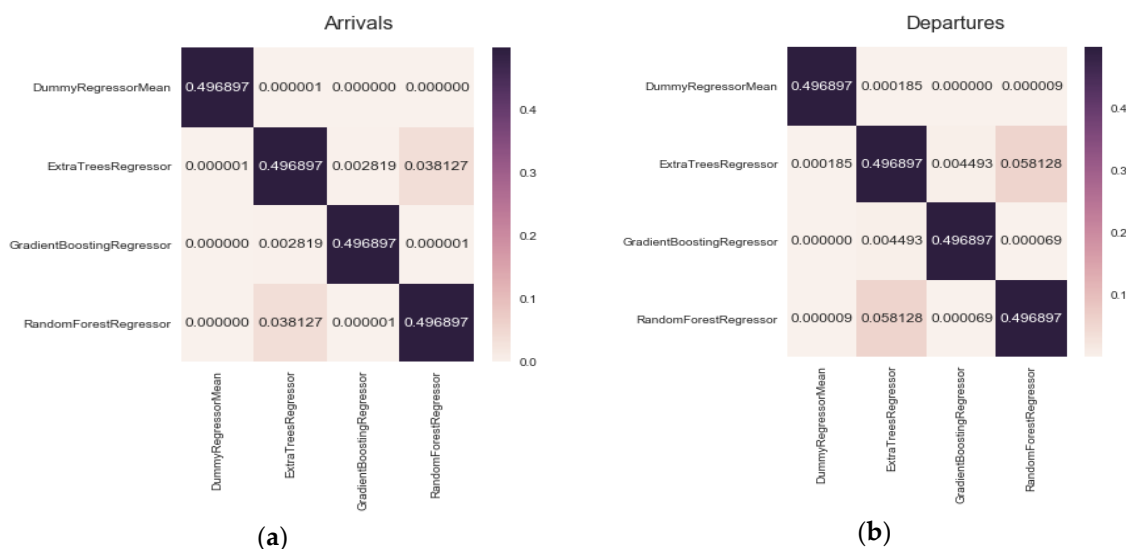


**Figura 37** Coeficiente de determinación de los modelos en las llegadas de cada estación



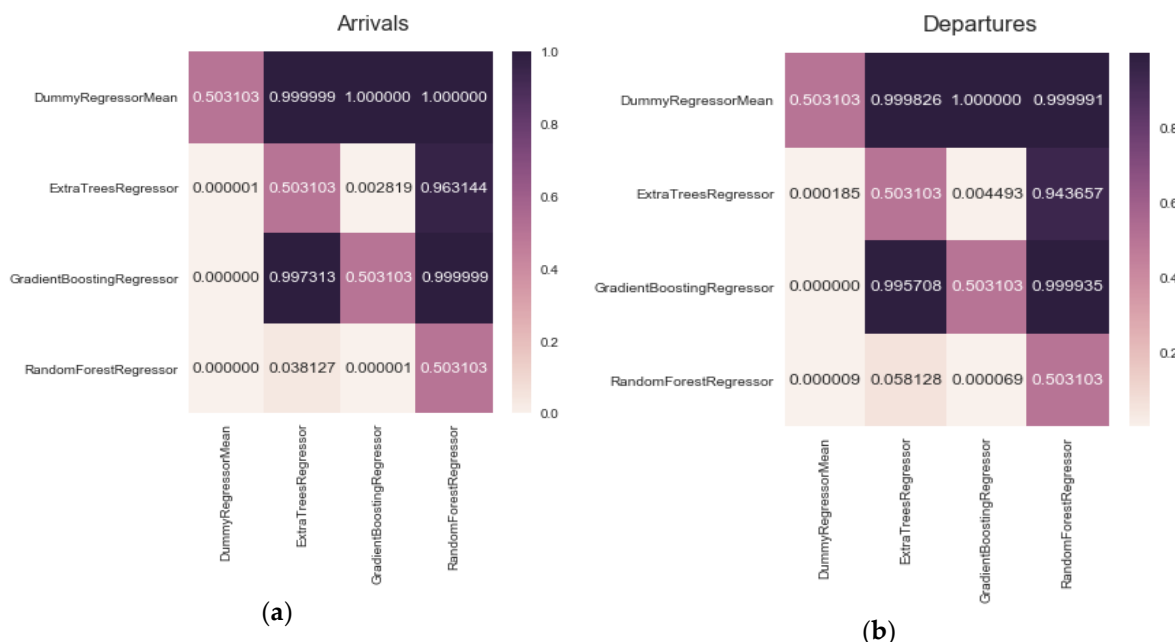
**Figura 38** Coeficiente de determinación de los modelos en las salidas de cada estación

Para seleccionar el mejor algoritmo para todas las estaciones del sistema, se ha empleado el test estadístico U de Mann-Whitney. En primer lugar, se ha realizado el test para comprobar si uno de los algoritmos empleados funciona de forma distinta al resto de ellos. El contraste de hipótesis aplicado es el siguiente:  $H_0$  considera la mediana de ambos métodos iguales mientras que  $H_1$  considera la mediana de ambos métodos diferentes. En ambos casos, como se puede observar en la **Figura 39**, para los modelos de salidas y llegadas, el algoritmo *Random Forest Regressor* posee una mediana diferente al resto, por lo que se establecería que existe una diferencia estadística significativa. El p-valor, obtenido para la pareja *Random Forest Regressor* y *Extra Tree*, sobrepasa ligeramente el nivel de significación del test de 0,05 por lo que, en este caso, no se puede afirmar que haya una diferencia estadística significativa, no obstante, al calcular la mediana para ambos métodos, se puede observar que la mediana para *Random Forest Regressor* es mayor frente a *ExtraTreesRegressor*.



**Figura 39** Test U de Mann-Whitney (H0 igualdad) para (a) llegadas (b) salidas.

Una vez llevado a cabo este test, es necesario realizarlo de nuevo para comprobar si dado el caso de que se detectaran diferencias estadísticas significativas se procedería a determinar si la mediana es menor o mayor, concretamente, la  $H_1$  definida ha sido que la mediana del clasificador de la fila es mayor que la mediana de la mediana del clasificador de la columna. Como se puede ver en la **Figura 40**, se cumple que *Random Forest Regressor* posee una mediana mayor que el resto de los algoritmos para todas las estaciones.

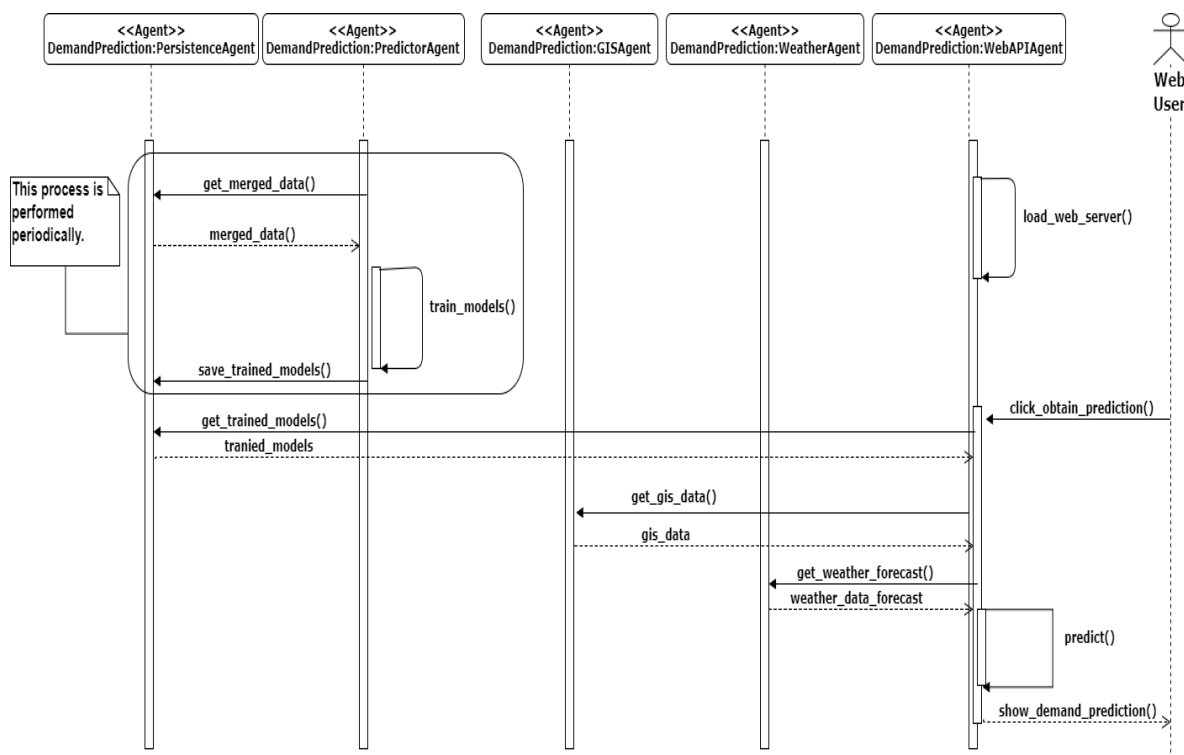


**Figura 40** Test U de Mann-Whitney (H0 menor que) para (a) llegadas (b) salidas.

Después de ver estos resultados, se ha seleccionado el modelo *Random Forest Regressor* para ser añadido al *Predictor Agent* y generará los modelos de forma periódica con la información disponible y los guardará para su posterior uso, sirviendo predicciones que le serán solicitadas por el *WebAPI Agent*.

En la **Figura 41** se muestra un diagrama de secuencia que describe las predicciones de la demanda. En primer lugar, el *Predictor Agent* entrena y almacena el modelo de predicción para cada estación; este proceso lo realiza el agente de forma periódica. Más tarde estos modelos se emplean para generar las predicciones solicitadas por el *WebAPI Agent* con el fin suministrarlas, a su vez, mediante una API REST.

Finalmente, el *WebAPI Agent* sirve una aplicación web donde los usuarios pueden solicitar las predicciones para la estación seleccionada. Para generar la predicción, el agente obtendrá los datos de entrada correspondientes tanto a la información horaria como a la información meteorológica, solicitando dicha información al *Weather Agent*. La interfaz que presenta este agente se detalla en la siguiente sección.

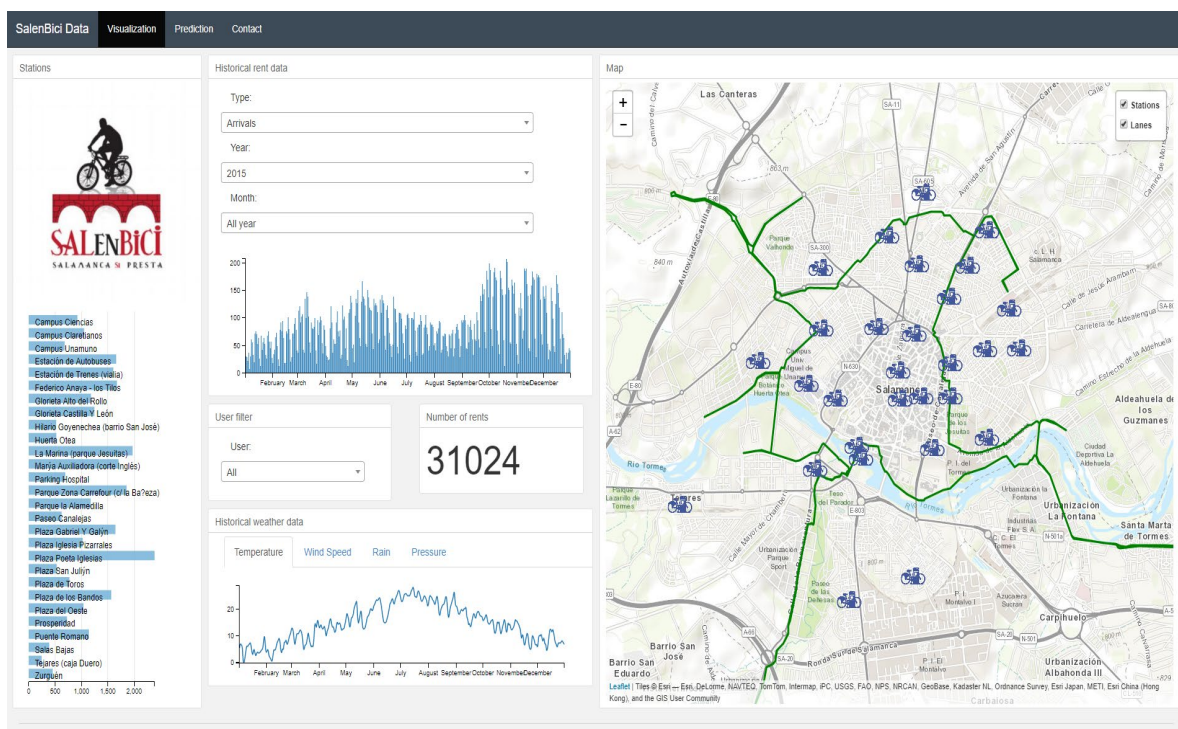


**Figura 41** Diagrama de secuencia del proceso de generación de predicciones de la demanda.

## 4.2.6 Herramienta de visualización y predicción

El WebAPI Agent ofrece una API REST para que terceras aplicaciones puedan obtener los datos procesados en el MAS y solicitar predicciones de la demanda para las estaciones del sistema. Además, sirve una aplicación web en la que el usuario puede realizar tareas de visualización y de predicción. La aplicación web posee dos secciones. La **Figura 42** muestra la vista inicial de la parte de visualización; en ella es posible ver los viajes por año y mes que se han realizado en el sistema.

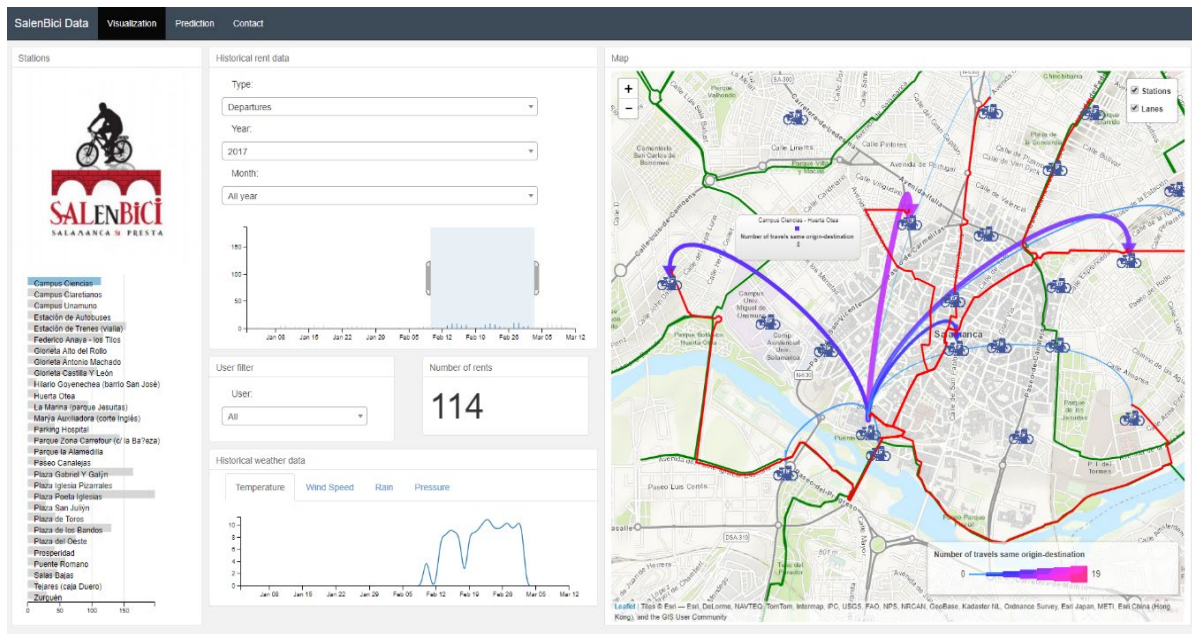
La parte de visualización cuenta con varias secciones en las que se puede seleccionar la estación, el periodo de tiempo (año, mes, etc.), el tipo de trayecto (llegadas o salidas) y el usuario. Estas secciones actúan como filtros y permiten focalizar la visualización en una serie de datos determinados.



**Figura 42** Aplicación Web servida por el WebAPI Agent. Sección visualización de datos históricos

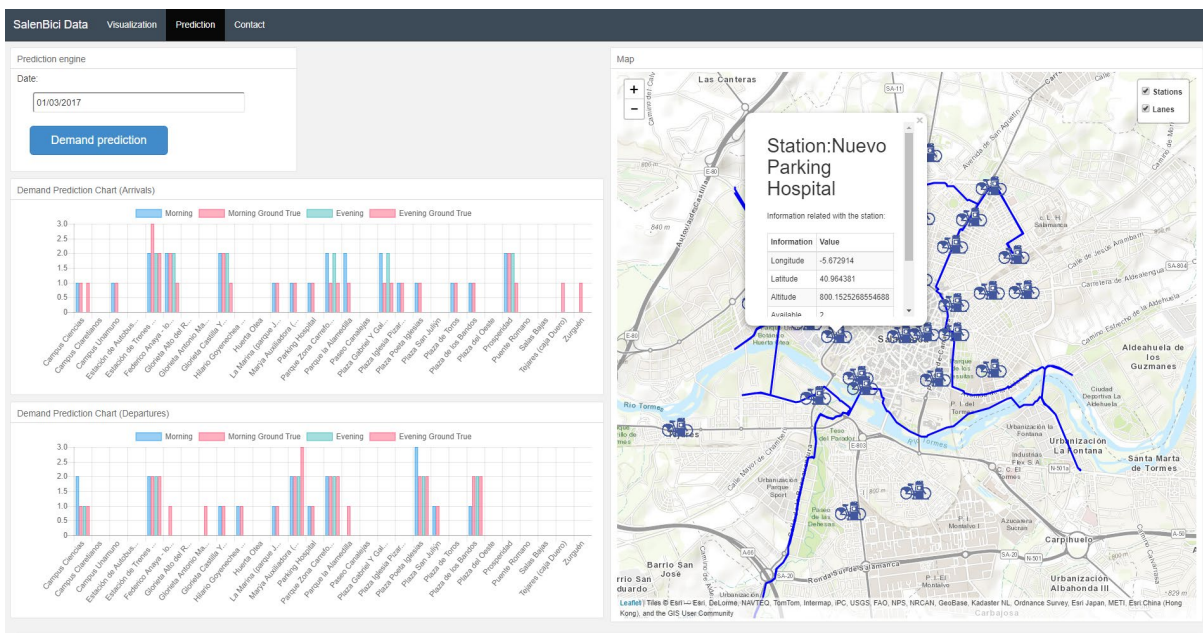
En el mapa, como se puede ver en la **Figura 43**, se representan las salidas desde la estación seleccionada, en la franja de tiempo seleccionada en la gráfica de la izquierda. La cantidad de viajes de la estación seleccionada al resto se representa en una escala de colores y el grosor de la flecha es proporcional al número de viajes efectuados entre ese origen y destino.





**Figura 43** Sección de visualización. Selección de estaciones individuales y periodo de tiempo y visualización de cantidad de viajes a cada estación.

Estas visualizaciones permiten al operador del BSS analizar los históricos del sistema y encontrar rutas habituales entre estaciones. En la **Figura 44** se muestra la sección de predicción de la demanda. En ella se emplearán los modelos seleccionados para predecir la demanda en cada estación.



**Figura 44** Sección de predicción de la demanda. Demanda para cada estación.

Es posible seleccionar tanto una fecha pasada como una fecha futura. Si se trata de una fecha pasada para la que el sistema ya posee datos, se mostrarán, adicionalmente a la predicción realizada por el modelo, los datos reales que se dieron en cada estación dicha fecha. Estos datos serán de utilidad para los operadores del BSS que realicen la reubicación de las bicicletas.

#### 4.2.7 Conclusiones

A lo largo de este caso de estudio se ha desarrollado y evaluado un sistema Multi-Agente orientado a *Bike Sharing Systems* con datos provenientes del sistema SalenBici de la ciudad de Salamanca. El modelado del sistema como un sistema Multi-Agente ha sido especialmente útil para recopilar datos de distintas fuentes y desacoplar cada una de las partes del sistema. La arquitectura propuesta permitiría la obtención de datos de otros sistemas BSS de forma sencilla.

Del mismo modo, el uso de agentes para las tareas de predicción ha permitido emplear información recopilada por otros agentes y ofrecer servicios a aplicaciones de terceros. El modelado de un Predictor Agent de forma aislada hace posible la implementación de sistemas de predicción de forma desacoplada de otros sistemas, permitiendo la reutilización de código para otros casos de uso en los que sea necesario implementar, entrenar y utilizar modelos de predicción de la demanda.

En este caso de estudio se han evaluado distintos algoritmos del sistema propuesto en esta tesis y se han aplicado a un caso de estudio con datos reales acerca de la demanda de un BSS. Se ha tenido en cuenta su rendimiento, precisión y su tiempo de entrenamiento y se han realizado test estadísticos para determinar el uso de ellos en la implementación final del sistema (debido al tiempo necesario para entrenar todos los modelos siguiendo la metodología propuesta). En este caso el algoritmo que mejores resultados ha obtenido frente al resto es *Random Forest Regressor*.

Adicionalmente, se ha diseñado e implementado un agente que proporciona una aplicación web como herramienta de visualización de datos históricos y otra de predicción de la demanda en las estaciones (empleando los modelos generados).

Respecto a la parte de visualización, el operario puede centrarse en las partes de los datos históricos de su interés gracias a los filtros implementados y la visualización realizada. Esta herramienta permite obtener conocimiento acerca del comportamiento de las estaciones del sistema y cómo se comportan sus usuarios.

La parte de la aplicación web relativa a la predicción puede ser empleada por los operarios para las tareas de reubicación de bicicletas a lo largo de la ciudad.

Finalmente, las predicciones obtenidas pueden ser empleadas en futuros trabajos de optimización de rutas de reubicación de la flota de bicicletas, mejorando las estrategias actuales.

### 4.3 Optimización de rutas de recogida con demandas estocásticas combinando metodología Simheuristics y modelos de predicción

En este caso de estudio se aborda la inclusión de la incertidumbre en la optimización de rutas de recolección de residuos empleando la metodología Simheuristics con una modificación propuesta en este trabajo de tesis.

Para evaluar si empleando modelos de predicción en la metodología Simheuristics [Juan et al. 2011] se mejoran los resultados de optimización frente a emplear el valor esperado de la distribución de desechos generados, se ha planteado un caso de estudio teórico en el que se realiza una simulación de datos de recogida para instancias VRP clásicas de la literatura. Algunas de estas instancias han sido previamente empleadas por Juan et al. en su trabajo para validar su metodología.

Este tipo de instancias es posible encontrarlas en [COIN-OR] y representan distintos problemas CVRP teóricos con sus correspondientes parámetros. Estos son: el número de *customers* con sus respectivas demandas, vehículos disponibles y matriz de costes. Habitualmente, estas instancias vienen acompañadas de la mejor solución obtenida hasta el momento con su respectivo coste.

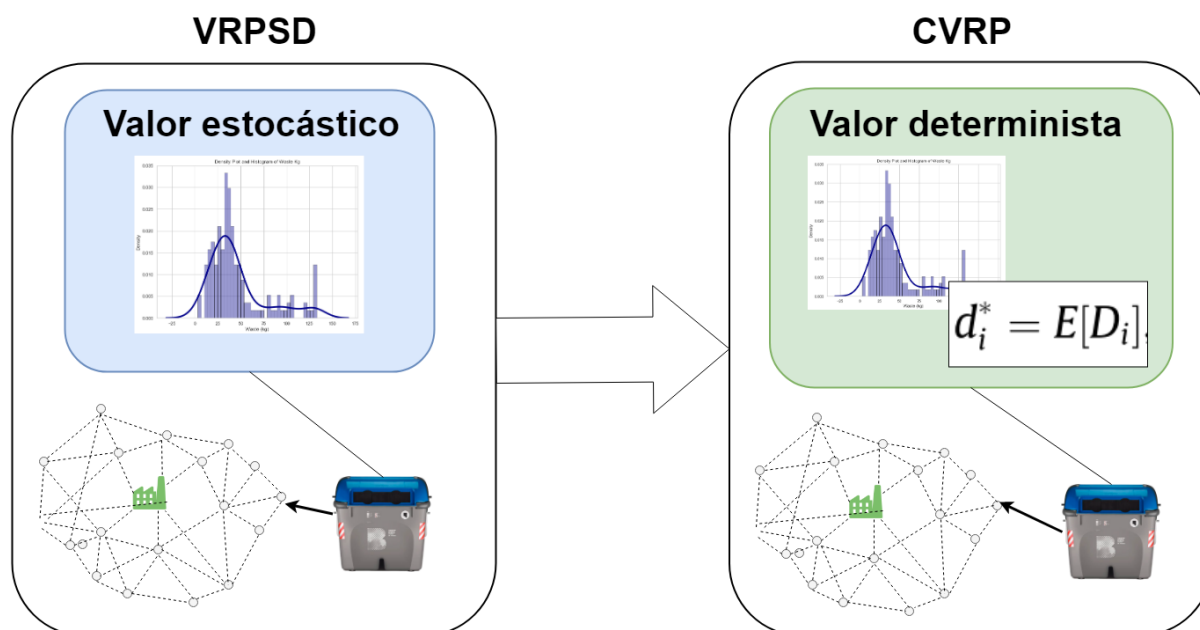
Dado que estas instancias son problemas CVRP y no VRPSD, Juan et al. toman las demandas de cada uno de estos problemas como el valor medio de una distribución Log-Normal convirtiéndolas en estocásticas y, posteriormente, realizan las simulaciones definidas en su metodología con dichas distribuciones.

A continuación, se describe la modificación que propone este trabajo en la metodología Simheuristics.

#### 4.3.1 Combinación de Simheuristics con modelos de predicción

En el presente trabajo de tesis se propone realizar una modificación en la metodología Simheuristics previamente descrita en el estado del arte. Esta modificación está especialmente focalizada en un punto del proceso de la metodología, en el proceso de simplificación que realizan los autores a la hora de pasar de un problema estocástico (VRPSD) a un problema determinista CVRP.

En este punto los autores proponen simplificar el problema, empleando el valor esperado de la distribución (teórica o empírica) que sigue cada contenedor; de esta forma el problema VRPSD pasa a ser un problema CVRP como se describe en la **Figura 45**.



**Figura 45** Proceso de simplificación seguido en la metodología Simheuristics

Cada uno de los contenedores toma, por tanto, el valor esperado descrito por la ecuación (15):

$$d_i^* = E[D_i] \quad (15)$$

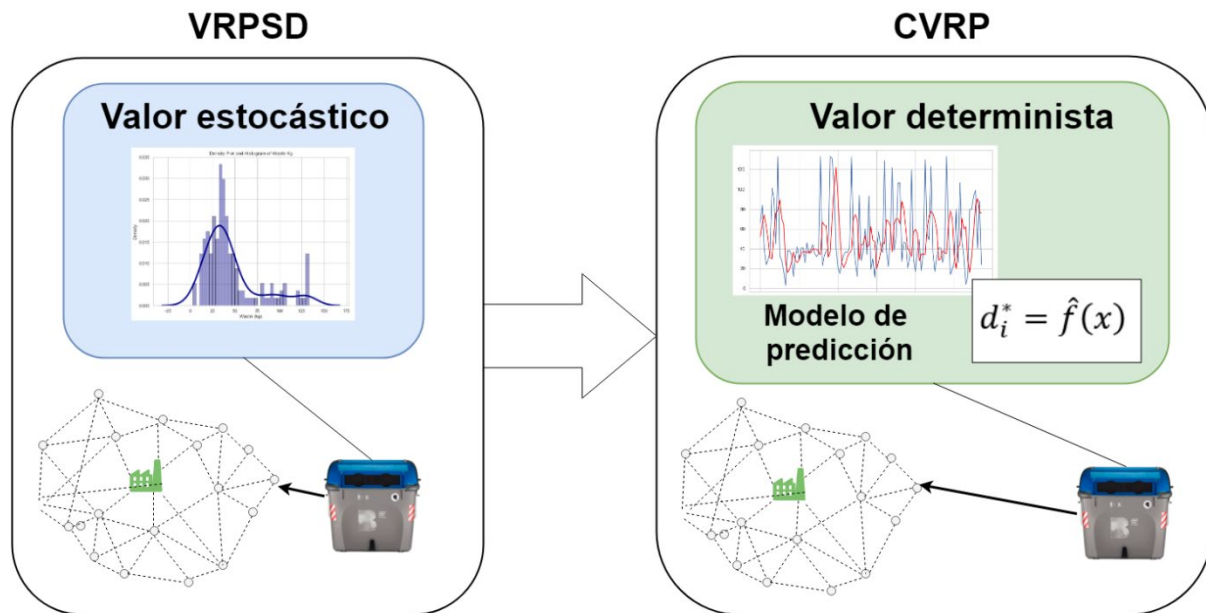
Donde  $d_i^*$  es la demanda estocástica de cada contenedor y sigue una distribución estadística definida, ya sea teórica o empírica, y  $E[D_i]$  simboliza el valor esperado de dicha distribución.

La modificación que se plantea se fundamenta en que en la producción de la demanda (de desechos en este caso, y en otros tipos de demanda de problemas similares a este) pueden existir patrones que guíen el comportamiento de dicha demanda y que dichos patrones podrían llegar a ser identificados por modelos de predicción. De esta forma, el valor proporcionado por el predictor de la demanda de cada uno de los contenedores podría ofrecer un valor más cercano al futuro valor que se encontrará en la recogida. Por tanto, si el funcionamiento de los predictores presenta un error inferior al producido por el valor esperado, el cálculo de soluciones para el caso CVRP determinista se encontrará más cercano al caso estocástico. Esto resultará en soluciones para problemas deterministas más cercanas a escenarios estocásticos, lo que implicaría una menor probabilidad de fallo en la ruta.

En este caso, los valores establecidos para cada contenedor se producen siguiendo la siguiente expresión (ecuación (16)):

$$d_i^* = \hat{f}(x) \quad (16)$$

Donde  $\hat{f}(x)$  representa el predictor que ha obtenido mejores resultados tras el proceso de entrenamiento y  $x$  las variables de entrada para dicho predictor para el día de recogida en cuestión. Una ilustración del proceso se puede observar en la **Figura 46**.



**Figura 46** Proceso de simplificación propuesto incluyendo modelos de predicción para la metodología Simheuristics

En este caso de estudio se realiza una comparación inicial entre los modelos de predicción y el valor esperado del último mes de recogida para distintos días de recogida de un sistema de recogida de desechos. De esta forma es posible comprobar el error producido por cada uno de los enfoques a la hora de estimar la cantidad de demanda total real de los contenedores. Posteriormente, una vez realizada la simplificación de un SVRP a CVRP, se continúa con el proceso descrito por la metodología Simheuristics. Para ello se realizarán las simulaciones siguiendo una distribución Log-Normal y un escenario de varianza 0.25. Al igual que en el trabajo de Juan et al., la media y la varianza para las simulaciones vienen descritas por las ecuaciones (17) y (18):

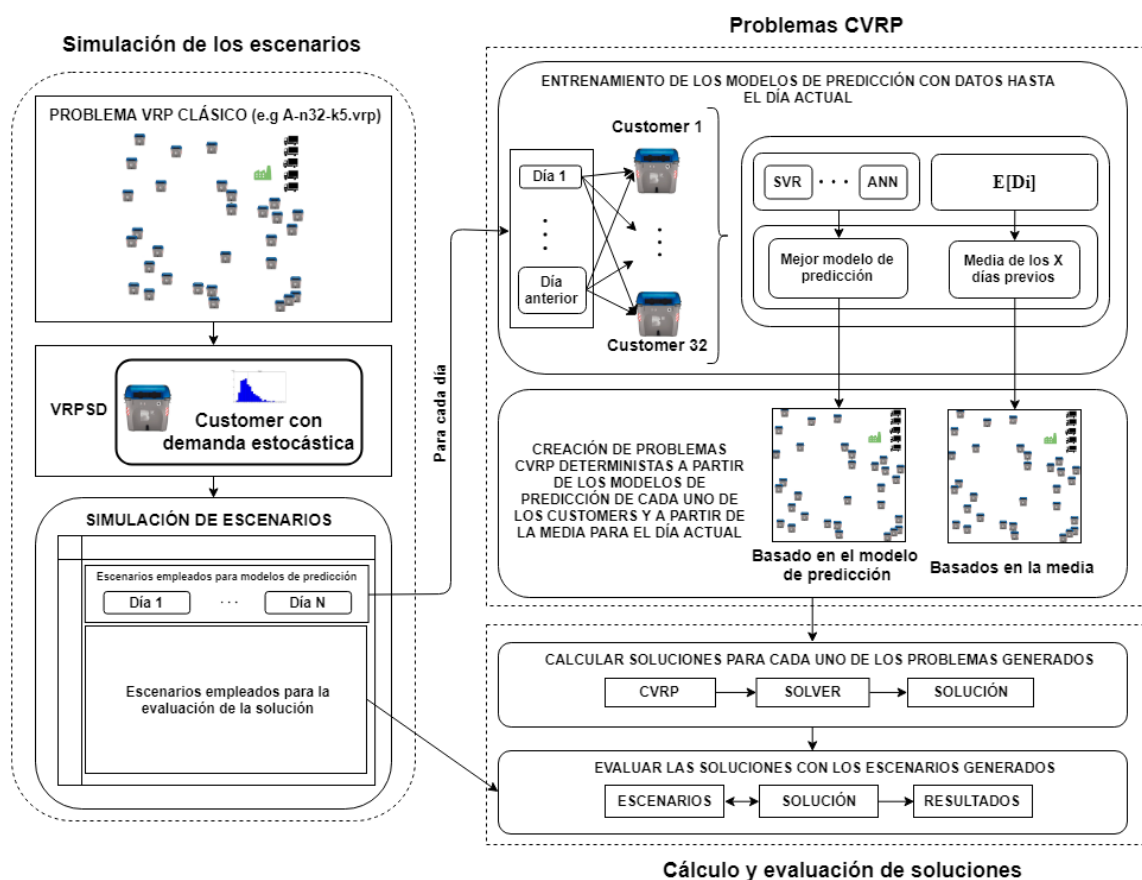
$$\mu_i = \ln(E[D_i]) - \frac{1}{2} \cdot \ln\left(1 + \frac{Var[D_i]}{E[D_i]^2}\right) \quad (17)$$

$$\sigma = \sqrt{\ln\left(1 + \frac{Var[D_i]}{E[D_i]^2}\right)} \quad (18)$$

Una vez realizadas las simulaciones, se comparan las soluciones obtenidas para cada uno de los dos enfoques (empleando el valor esperado y empleando el mejor modelo de predicción) y se comparan sus índices de factibilidad y costes medios de simulación obtenidos

### 4.3.2 Caso de estudio

Una vez descrita la modificación introducida en la metodología, se plantea un caso de estudio teórico en el que se simulen datos relativos a un problema CVRP como el que se da en los sistemas de recolección de desechos. Cabe destacar que la modificación propuesta puede ser empleada en cualquier problema CVRP independientemente del caso de estudio.



**Figura 47** Esquema principal del diseño del caso de estudio

En la **Figura 47** se muestra un diagrama general del caso de estudio propuesto con las fases del proceso. En la parte izquierda se puede observar el proceso de simulación de los datos y también de los escenarios para la posterior evaluación de soluciones, seguidamente en la parte superior derecha se entrenan los modelos de predicción con los datos generados y también se obtienen predictores basados en la media de las últimas recolecciones. Una vez obtenidos los modelos, se generan las demandas de cada uno de los *customers* para generar así 2 problemas

CVRP, uno basado en el uso de la media y otro basado en el uso del valor predicho por el regresor. Seguidamente se realiza un proceso de cálculo de soluciones para los problemas planteados y finalmente, se realiza una evaluación de las soluciones frente al conjunto de escenarios generados inicialmente. En posteriores secciones se detallan cada una de las partes del proceso en profundidad y finalmente se presentan los resultados obtenidos tras dicho proceso.

#### 4.3.2.1 Simulación de los escenarios

Se describen a continuación los parámetros (**Tabla 4**) y el proceso llevado a cabo para la generación de los escenarios para el caso de estudio. Esta sección corresponde con la parte izquierda del diagrama presentado anteriormente. Se han generado escenarios relativos a las demandas (la cantidad de desechos generados en cada contenedor) para cada uno de los días del año comprendidos entre *initDate* y *endDate*. Como base se han tomado instancias CVRP clásicas de la literatura (*InstanceCVRP*) como, por ejemplo, la instancia A-n32-k5.vrp, que representa un problema CVRP de 32 clientes y 5 vehículos con unos costes y demandas determinados. Esta instancia se empleará para ejemplificar el proceso seguido en el caso de estudio.

**Tabla 4** Parámetros para la generación de los escenarios para instancia A-n32-k5.vrp

Parámetro	Definición	Valor
<i>InstanceCVRP</i>	Instancia del problema CVRP	A-n32-k5.vrp
<i>numScenarios</i>	Número de escenarios que se generarán	10000
<i>collectionDays</i>	Periodo de días entre cada una de las recogidas de desechos	3 días
<i>initDate</i>	Día de inicio del periodo simulado	01/01/2017
<i>endDate</i>	Día de fin del periodo simulado	01/01/2019
<i>calendar</i>	Calendario laboral del país tomado como referencia para festivos	España
<i>variance</i>	Varianza de la distribución empleada para la simulación de los escenarios	0.25
<i>percentageHolidays</i>	Porcentaje de incremento sobre la media tomada para cada <i>customer</i> en un día festivo	0.8
<i>seed</i>	Semilla para la generación de números aleatorios	7



Las distribuciones empleadas para generar los datos son las empleadas por Juan et al. en su trabajo [Juan et al. 2011], descritas anteriormente, es decir, se toma la demanda de cada uno de los clientes de la instancia CVRP como media de la distribución empleada para generar los valores de demanda de los escenarios.

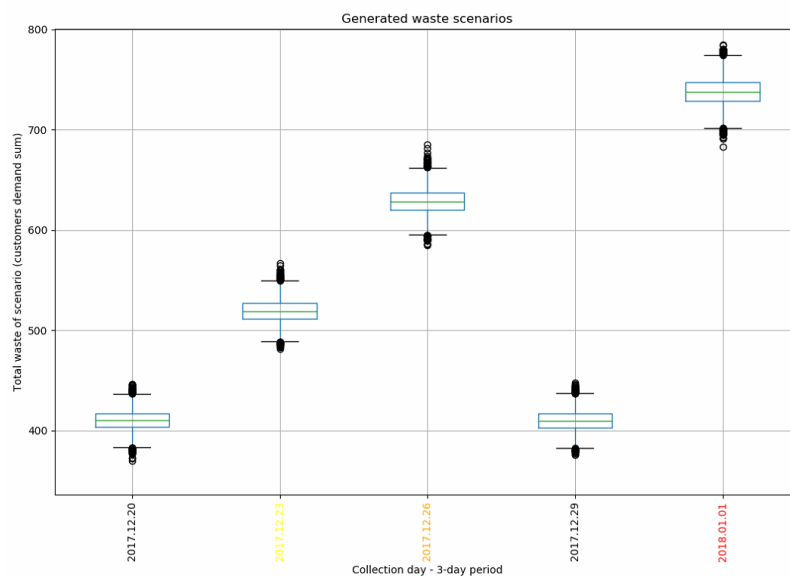
Para el caso de estudio se han establecido dos distribuciones para la generación de los valores de demanda: la primera, que corresponde con los días de lunes a viernes no festivos, y otra para el resto de festivos y fines de semana, descritas por las siguientes ecuaciones (19) y (20)

$$d_{iworkdays} = \ln(E[D_i]) - \frac{1}{2} \cdot \ln\left(1 + \frac{Var[D_i]}{E[D_i]^2}\right) \quad (19)$$

$$d_{iholydays} = d_{iworkdays} + (d_{iworkdays} \cdot percentageHolidays) \quad (20)$$

Con esto se ha establecido como patrón arbitrario que los fines de semana y festivos tomen una distribución cuya media es un porcentaje más alta que el resto (*percentageHolidays*). Con ello se pretende introducir un patrón en la generación de los datos que se espera que sea detectado por los algoritmos de predicción empleados.

Seguidamente, los datos son agrupados en recogidas, simulando el comportamiento que ocurriría en un sistema de recolección de basuras convencional (parámetro *collectionDays*). Para asegurar la reproducibilidad del experimento, se ha establecido una semilla común de números aleatorios en las bibliotecas empleadas (*seed*).



**Figura 48** Ejemplo de los datos generados para 5 recogidas cada 3 días.



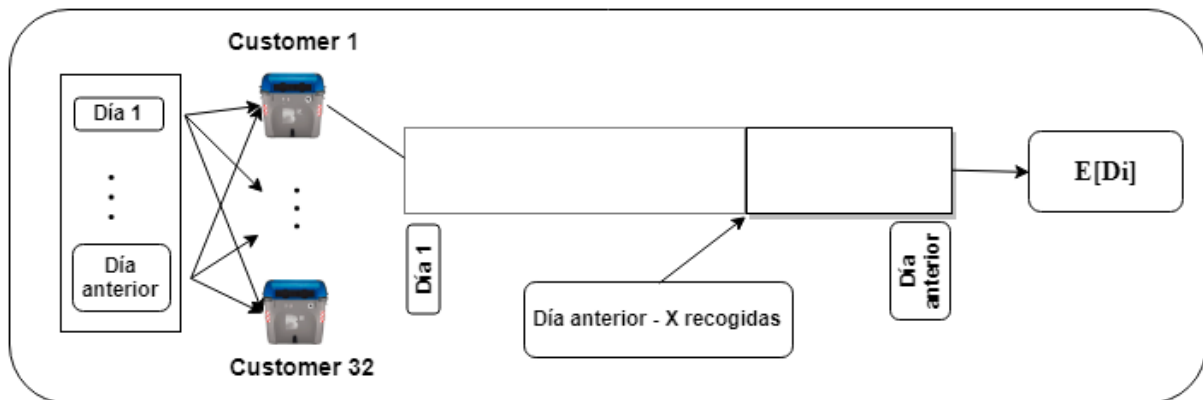
En la **Figura 48** se presenta un extracto de los datos generados para 5 días correspondientes a recogidas de desechos. En la leyenda de cada una de las recogidas se puede ver en color negro aquellos días de recogida en los que los días previos no incluían días festivos, mientras que en colores amarillo, naranja y rojo se ven aquellas recogidas que incluyen un porcentaje más alto de días festivos en los días previos a la recogida. Es posible apreciar cómo las demandas totales de los escenarios generados son mayores en aquellas recogidas que incluyen más días festivos.

### 4.3.2.2 Generación de los problemas deterministas CVRP

Esta fase del caso de estudio corresponde con la parte superior derecha de la **Figura 47**. Se realiza el proceso de simplificación de VRPSD a CVRP mediante dos enfoques distintos: empleando, por un lado, el valor medio de las últimas  $X$  recogidas y, por otro, modelos de predicción con los datos históricos hasta ese momento. Los datos empleados para la media y para el entrenamiento de los modelos de predicción corresponden a un único conjunto de escenarios para los días especificados anteriormente. A continuación, se describen los dos procesos empleados para obtener los valores deterministas de los problemas CVRP.

#### 4.3.2.2.1 Media

Para la media se tomará la información disponible de una ventana temporal de  $X$  recogidas. Este proceso se realizará para cada uno de los *customers* involucrados en el problema. En la **Figura 49** se presenta un diagrama que muestra el proceso descrito.



**Figura 49** Datos empleados de cada contenedor para obtener la media para el proceso de simplificación

El valor calculado para cada *customer* será el empleado para la predicción de la demanda para el problema del día que se esté tratando en ese momento.

### 4.3.2.2 Modelos de predicción

Por otro lado, se utiliza la información de todos los días de recolección para entrenar distintos algoritmos de predicción con el objetivo de obtener un pronóstico certero para el día que se esté tratando. En la **Tabla 5** se muestran las características seleccionadas para entrenar los regresores. En la tabla  $t$  representa la recogida actual y se ha considerado un total de 7 valores rezago (*lag features*), no incluidos en la tabla por brevedad.

**Tabla 5** Características de entrada para los regresores

Atributo	Descripción	Tipo
$demand(t - 1)$	Demanda del último día de recogida (kg)	Continua
...	...	...
$demand(t - 7)$	Demanda de siete recogidas antes (kg)	Continua
$holiday\_percentage$	Porcentaje de días festivos hasta el día actual [0 - 1]	Continua
$holiday\_percentage(t - 1)$	Porcentaje de días festivos hasta el día de recogida anterior [0 - 1]	Continua
...	...	...
$holiday\_percentage(t - 7)$	Porcentaje de días festivos hasta el día de 7 recogidas antes [0 - 1]	Continua
$week\_day$	Día de la semana de la presente recogida [0-6]	Categórica
$month$	Mes del año [1-12]	Categórica

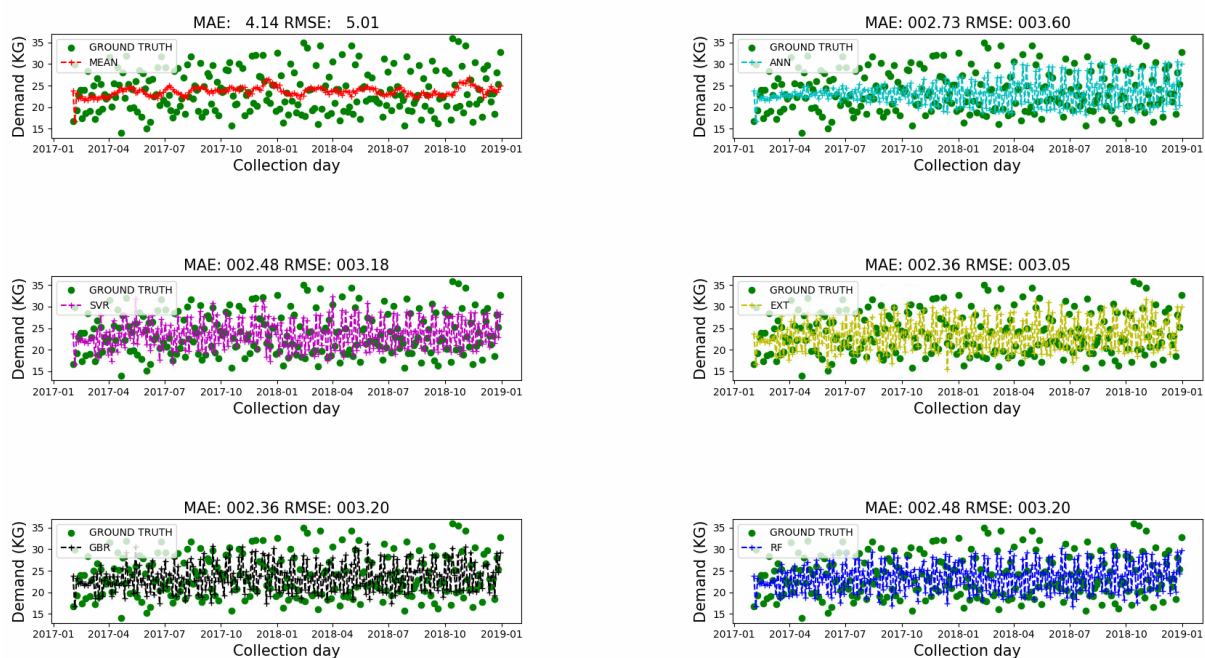
Por otro lado, en la **Tabla 6** se presentan los algoritmos de regresión empleados para generar modelos de predicción de la demanda. Se indica el *grid* de parámetros empleado para la selección de hiperparámetros empleando *GridSearch* junto con el esquema de *cross validation* de *TimeSeriesSplit*.

**Tabla 6** Parámetros para la generación de los escenarios

Algoritmo	Grid de parámetros
<i>Support Vector Regressor</i>	{ "C": [1e0, 1e1, 1e2, 1e3], "gamma": np.logspace(-2, 2, 5) }
<i>GradientBoostingRegressor</i>	{ learning rate: [0.1, 0.01, 0.001], subsample: [1.0, 0.9, 0.8], max depth: [3, 5, 7], min samples leaf: [1, 3, 5] }
<i>ExtraTreeRegressor</i>	{ number estimators: [10, 100, 1000], max features: [auto, sqrt, log2] }
<i>RandomForestRegressor</i>	{ number estimators: [10, 100, 1000], max features: [auto, sqrt, log2] }
<i>ANN</i>	{ optimizer: [sgd, adam], epochs [10, 20, 30], activations = ['relu', 'tanh'] }

El proceso de entrenamiento y selección de modelos se ha llevado a cabo siguiendo el esquema descrito en el sistema propuesto. Se ha seleccionado el modelo que menor Mean Absolute Error ha presentado tras este proceso y se ha empleado para generar la predicción del siguiente día de recogida para cada uno de los *customer* y los días involucrados en la simulación.

En la **Figura 50** se muestran las predicciones para uno de los contenedores de la instancia A-n32-k5.vrp para cada uno de los días de recogida. También se muestran los valores de los escenarios frente a las predicciones de la media y del regresor que mejor rendimiento ha desempeñado en el proceso anterior.



**Figura 50** Ejemplo de resultados de los regresores para un contenedor del problema A-n32-k5.vrp

Por tanto, después de este proceso se han obtenido tantas instancias CVRP como días de recolección se han planteado en el estudio, tanto para el enfoque de la media como para el enfoque de los modelos de predicción. Estos problemas CVRP se resolverán más adelante y, posteriormente, las soluciones obtenidas serán evaluadas con el resto de los escenarios previamente generados.

#### 4.3.2.3 Cálculo de soluciones de problemas CVRP y evaluación frente a escenarios

Tras la generación de las instancias CVRP deterministas para cada una de las recogidas del periodo establecido en el caso de estudio, se han calculado las soluciones a estas instancias

empleando los siguientes parámetros mostrados en la **Tabla 7**. Para la implementación de esta parte se ha utilizado la biblioteca Google OR-Tools [Google 2019].

**Tabla 7** Parámetros empleados en el solver

Parámetro	Descripción	Valor
<i>firsSolutionStrategy</i>	Método que usará el <i>solver</i> para encontrar la primera solución	PARALLEL_CHEAPEST_INSERTION
<i>LocalSearchMetaheuristic</i>	Estrategia de búsqueda local seguida	GUIDED_LOCAL_SEARCH
<i>timeLimit</i>	Límite de tiempo de búsqueda de soluciones	15 minutos

Estos parámetros se utilizarán en la resolución de los problemas CVRP previamente generados. Para cada uno de estos problemas, se generará un conjunto de problemas teniendo distintos valores de  $k$  entre 0.95 y 1.0. Tras este proceso, se obtendrán las soluciones (conjunto de rutas, vehículos y costes asociados) para cada problema CVRP establecido por el enfoque de la media y por el enfoque de los regresores.

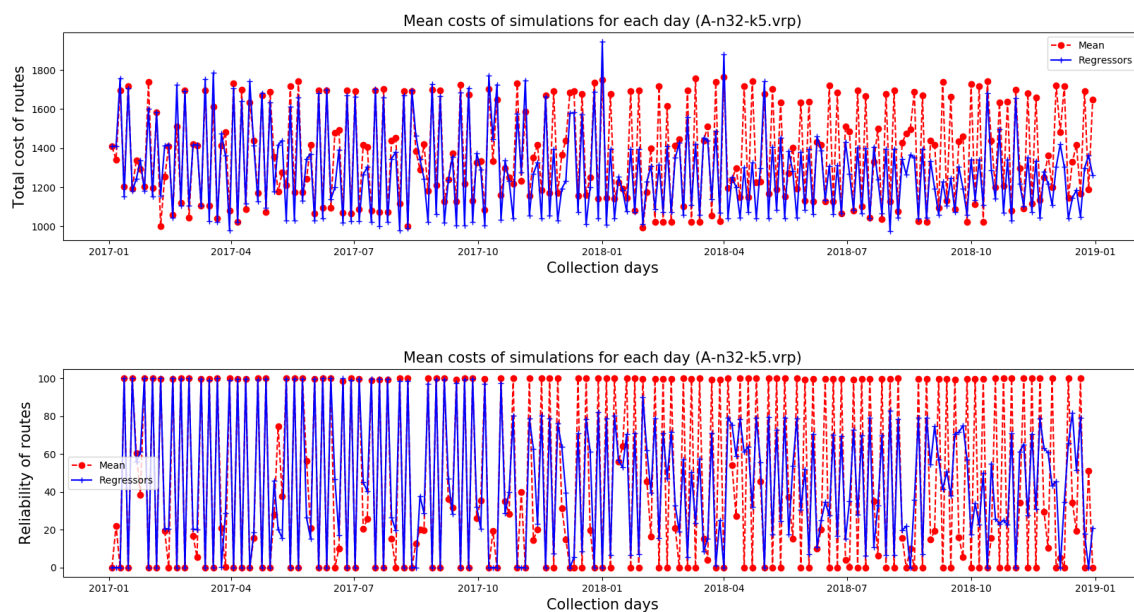
Posteriormente, se evalúan las soluciones frente a los escenarios generados anteriormente. Esta evaluación consiste en aplicar la solución al escenario generado, pudiéndose realizar las rutas o, por el contrario, produciéndose excesos de demanda en determinados puntos y, por tanto, generándose *recourse actions* que incrementarán el coste de la solución.

Tras esta evaluación, se selecciona la solución con un coste medio de simulación más bajo, es decir, el coste medio de todas evaluaciones con los distintos escenarios generados. Tras el proceso de evaluación de soluciones se obtiene para cada una su coste medio de simulación, el índice  $k$  safety stock usado en cada caso y un porcentaje de factibilidad de la ruta frente de los escenarios, es decir, un valor porcentual que expresa la cantidad de problemas de la simulación que se han podido llevar a cabo sin realizar *recourse actions*.

A continuación, se muestra una gráfica con los resultados de los costes medios de simulación de las soluciones calculadas para los problemas generados, tanto con el enfoque anterior de la media, como con los modelos de predicción. En la **Figura 51** se muestra una gráfica con los costes totales para cada uno de los días de recogida en el periodo especificado en el caso de estudio para el problema A-n32-k5. Se presentan los costes medios de las 10000 simulaciones para cada día de recogida junto con los valores de factibilidad de las rutas.

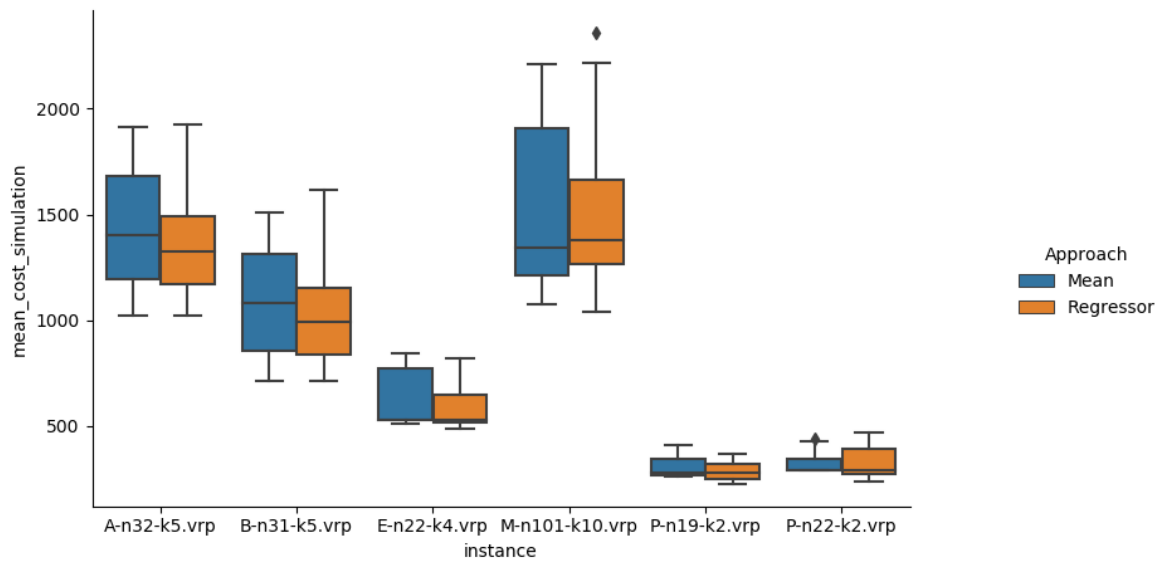
En la gráfica superior se observa cómo en las primeras recogidas el funcionamiento es el mismo ya que aún los regresores no disponían de mucha información para realizar la inferencia. Sin embargo, a medida que pasa el tiempo los regresores son capaces de estimar mejor la demanda

y producir soluciones con un coste de simulación más bajo. En la **Figura 51** el Coste de simulación medio (CSM) para las soluciones de cada día de recolección (arriba) y la factibilidad de las soluciones en la simulación (abajo). vemos que los regresores presentan un índice de factibilidad más robusto que la media, ya que ésta propone soluciones con un índice de factibilidad o bien muy alto (debido a una solución con un coste alto), o bien nulo (debido a un ajuste muy pobre a los escenarios). Es posible observar que la factibilidad en el caso de la media o bien es un prácticamente un 100% porque se plantea una solución con un coste muy alto o bien un 0% porque se ha planteado una solución con un coste muy bajo. En el caso de los regresores ofrecen un porcentaje medio con un coste de simulación más bajo.



**Figura 51** Coste de simulación medio (CSM) para las soluciones de cada día de recolección (arriba) y la factibilidad de las soluciones en la simulación (abajo).

El proceso completo descrito en el caso de estudio se ha aplicado de diferentes instancias CVRP ampliamente utilizados en la literatura como las estudiadas por Juan et al. A continuación, se muestran los resultados de los costes de simulación medios de todo el periodo de las simulaciones para cada uno de ellos en la **Figura 52**. Se ha realizado el test estadístico U de Mann Whitney para cada instancia estudiada. En las instancias A-n32-k5, B-n31-k5, E-n22-k4 y P-n19-k2 el test ha determinado que existen diferencias estadísticas significativas para afirmar que los resultados obtenidos de los regresores son menores que los de la media. Por otro lado, para las instancias P-n22-k2 y M-n101-k10 no se han encontrado diferencias estadísticas significativas.



**Figura 52** Boxplots del Coste Medio de Simulación para cada uno de los métodos e instancias CVRP estudiadas

### 4.3.3 Conclusiones y líneas futuras

Tras la realización del caso de estudio y la evaluación cada una de las partes incluidas en éste, se extraen distintas conclusiones relativas a los resultados de los modelos de predicción, a la modificación de simplificación propuesta en este trabajo y, por último, a los resultados en coste obtenidos en la optimización de rutas.

En primer lugar, se han entrenado modelos de predicción empleando diferentes algoritmos del estado del arte para la predicción de la demanda en cada uno de los contenedores. Se han comparado los valores obtenidos de dichos modelos con los obtenidos por la media de las últimas recogidas, ofreciendo un mejor rendimiento como cabía esperar. Es importante destacar que los modelos generados podrían capturar otros patrones distintos en los datos que los introducidos en el caso de estudio, como por ejemplo tendencias en determinados meses o semanas del año o picos alrededor de determinadas fechas especiales del año.

En segundo lugar, los resultados obtenidos en el caso de estudio, vemos que las soluciones proporcionadas por los modelos de predicción se encuentran más cercanas al caso estocástico, permitiendo obtener mejores costes de simulación, desembocando esto en soluciones más robustas.

Cabe mencionar que dicha modificación solo es efectiva en los casos en los que los modelos de predicción realicen un buen desempeño en comparación a obtener el valor medio de la distribución que sigue la variable estocástica. Además, resaltar que esta propuesta, frente a la

empleada por Juan et.al., incorpora más dimensiones al problema (todas aquellas relativas a una serie temporal y las variables adicionales que incluyan).

Se observa que los resultados obtenidos en los modelos de predicción están directamente relacionados con la calidad de la solución obtenida posteriormente, por lo tanto, un buen ajuste del modelo facilitará la obtención de mejores resultados ya que el escenario estocástico estará más próximo al determinista obtenido.

Después de llevar a cabo el caso de estudio, se propone emplear esta misma modificación en otros casos de estudio en los que se aplique la metodología Simheuristics ya que el enfoque propuesto puede trasladarse a otros dominios de aplicación más allá de los *Smart Waste Collection Systems*.





---

## CAPÍTULO 5

---



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

## CONCLUSIONES

---



## Capítulo 5 Conclusiones

Este capítulo resume las conclusiones obtenidas tras los 3 casos de estudio desarrollados a lo largo todo el trabajo de tesis doctoral. Al mismo tiempo, señala las líneas de trabajo futuras a seguir tras el trabajo desarrollado.

A lo largo de este capítulo se aglutinan las conclusiones generales extraídas tras las investigaciones realizadas a lo largo del presente trabajo de tesis. Del mismo modo se recogen las contribuciones que se han aportado con este trabajo y las líneas de investigación a seguir en futuras investigaciones.

### 5.1 Contribuciones del trabajo de tesis

En esta parte se resumen las contribuciones principales de este trabajo de tesis. Como principales contribuciones en el estudio y revisión del estado del arte se pueden destacar:

- Se ha llevado a cabo una revisión del estado del arte de las redes de sensores inalámbricas que pueden ser empleadas en la toma de datos de sistemas de logística inteligente. Se han estudiado los diferentes tipos de redes y en concreto las denominadas LPWAN y, dentro de éstas, se han revisado y comparado sus características, tanto a nivel de eficiencia energética como de esquemas de utilización.
- Se ha efectuado un estudio de distintos algoritmos empleados en el modelado predictivo de la demanda. Se ha llevado a cabo una revisión de distintas técnicas que han sido empleadas en la literatura para generar modelos de predicción de la demanda en distintos dominios.
- Asimismo, una se ha elaborado una revisión de la formulación de los problemas VRP utilizados ampliamente en los sistemas de logística, seguido de un estudio de su taxonomía, sus métodos de resolución y, finalmente, se ha profundizado en un tipo especial de estos problemas que aborda la inclusión de la incertidumbre en su resolución. Se han revisado métodos de resolución y nuevas metodologías que abordan este tipo de problemas.

Seguidamente, tras la revisión de los temas principales de la tesis realizada, en este trabajo, se ha presentado el diseño de un sistema de logística inteligente con las siguientes características:

- Incorporación de redes inalámbricas de sensores junto con sensores de bajo consumo para la obtención de datos de forma remota.
- Empleo de los datos obtenidos para diseñar modelos de predicción de la demanda que modelan la cantidad de desecho que va a ser generado en cada contenedor.
- Introduce el estudio de la incertidumbre en los métodos de optimización de rutas y emplea los mencionados modelos de predicción para alcanzar mejores soluciones.

Para la evaluación y validación de este diseño, se han elaborado diferentes casos de estudio individualizados que han permitido centrar la investigación en cada una de las partes que componen el sistema. Además, este enfoque de trabajo está totalmente en consonancia con la metodología *action research* que promueve la realización de casos de estudio que mediante la acción y la obtención de resultados refinan la solución propuesta de forma iterativa. Esta metodología de trabajo permite evaluar las distintas partes del sistema y extraer conclusiones valiosas para el diseño global del sistema propuesto y en conclusiones concretas acerca de cada caso de estudio.

En primer lugar, se planteó la realización de un caso de estudio que abordase la obtención de datos mediante una red de sensores. Los sensores implicados deben poseer una serie de características enfocadas a su desempeño en el sistema, por lo que en este trabajo se presenta el diseño de un sistema de bajo consumo junto unido a una optimización de rutas empleando datos obtenidos por dicho sensor. Se concluyó que:

- Tras el estudio de las distintas tecnologías se concluye que el uso de sensores con un microprocesador de ultra bajo consumo energético, unido a una red inalámbrica LoRaWAN y siguiendo un esquema colaborativo es una de las mejores opciones a tomar en este tipo de sistemas.
- La optimización de rutas siguiendo un enfoque dinámico de cálculo de rutas frente a métodos convencionales de rutas estáticas proporciona de ahorros en combustible, tiempo, y mano de obra.
- El coste de despliegue y sensorización de contenedores sigue siendo relativamente alto pero su tendencia a disminuir unido al uso de redes LPWAN como la propuesta reduce notablemente la inversión en infraestructura necesaria, siendo posible dar cobertura a grandes extensiones de terreno únicamente empleando un número muy limitado de *gateways*.
- La utilización de un enfoque colaborativo en el despliegue de este tipo de redes favorece el desarrollo de soluciones IoT y puede suponer una infraestructura para futuras soluciones en ámbitos rurales como urbanos.

En segundo lugar, gracias al proyecto MOVIURBAN [MOVIURBAN] se pudo tener acceso a datos históricos de un sistema de logística como es un sistema de alquiler de bicicletas o *Bike Sharing System*. Se planteó realizar un caso de estudio con dichos datos para evaluar el uso de las técnicas de predicción de la demanda en dicho ámbito para más adelante, si era posible acceder a los datos de desechos emplear el conocimiento y conclusiones extraídas para dicho caso de estudio. En este tipo de sistemas, al igual que muchos otros de logística inteligente, es necesario predecir que demanda de un determinado bien será necesario recoger o depositar y en función de esto realizar una optimización de rutas. En este caso de estudio se planteó un sistema Multi-Agente para realizar las tareas de obtención, procesamiento de datos de diferentes fuentes de datos y la generación de modelos de predicción. Se concluyó que:

- La utilización de modelos de predicción puede ser especialmente útil para la posterior optimización de las rutas de los vehículos de rebalanceo de bicicletas en el sistema.
- La utilización de un sistema Multi-Agente puede ser muy útil a la hora de modelar un sistema con distintos subsistemas de datos muy heterogéneos.
- El desarrollo de una herramienta de visualización podría ser útil para los operarios a la hora de encontrar patrones en el uso del sistema por los usuarios y el comportamiento de estos.

Por último, se planteó el desarrollo de un sistema de optimización que tuviese en cuenta la incertidumbre presente en los sistemas VRP y que hiciese uso de los modelos de predicción previamente estudiados en el caso de estudio anterior para mejorar la resolución de este tipo de problemas. Tras la revisión del estado del arte se propuso realizar una modificación en la metodología Simheuristics (que aborda la resolución de problemas SVRPD). Esta modificación se basa en la inclusión de modelos de predicción de la demanda en la simplificación del problema que realizan en dicha metodología. Se mencionan a continuación las conclusiones generales del trabajo realizado:

- La propuesta de Simheuristics ofrece grandes ventajas frente al resto de métodos de resolución vistos en la literatura para el caso de VRP estocástico. En lugar de afrontar el problema como un problema de optimización estocástica o robusta, combina las simulaciones con los métodos de cálculo de problemas VRP deterministas. Gracias a ello, es posible plantear problemas a problemas estocásticos empleando técnicas para problemas deterministas.
- Se ha presentado como propuesta de modificación de esta metodología la utilización de modelos de predicción en el paso en el que se obtiene el problema VRP determinista sobre el que se realizarán las simulaciones. Tras evaluar la modificación propuesta para esta metodología, se concluye que puede presentar mejoras frente a la original siempre que se los modelos de predicción realicen un buen desempeño.
- Si los modelos de predicción de la demanda generados presentan un buen rendimiento y consiguen predecir de forma certera la demanda para cada contenedor, el proceso de

simplificación se encontrará más cercano al caso determinista por lo que se obtendrán mejores soluciones que si únicamente se emplea el valor esperado de la distribución de probabilidad de la demanda.

- La metodología Simheuristics proporciona una gran herramienta de toma de decisiones a la hora de manejar la incertidumbre asociada a la generación de desechos.
- Cabe destacar que es necesario estudiar la modificación introducida en esta metodología en otros dominios en los que la demanda presente una variabilidad más alta.

Finalmente, a lo largo de esta investigación se ha asistido a cursos workshops y conferencias que han fomentado el intercambio de conocimiento con otros estudiantes e investigadores. Además, se han realizado tareas de difusión del trabajo realizado, mediante la elaboración de artículos para revista científicas con índices de impacto al mismo tiempo que se ha acudido a congresos de ámbito internacional donde se han realizado ponencias y discusiones con otros investigadores.

## 5.2 Líneas de trabajo futuras

En esta sección se resumen las principales líneas futuras de investigación tras el trabajo realizado en las presentes tesis.

Relativo al primer caso de estudio y a la investigación en redes de sensores, es necesario realizar futuros estudios con gran parte de la flota de contenedores dotados de sensores para comprobar la evolución de generación de desechos en los contenedores. Del mismo modo, es necesario realizar test de cobertura en los lugares aislados y con poca línea de visión para evaluar y validar el estudio teórico de cobertura presentado.

Respecto al segundo tema de estudio y relativo a los modelos de predicción y como emplear sus resultados, es necesario realizar casos de estudio que involucren la reubicación de bicicletas empleando los datos obtenidos por los predictores. Además, sería interesante contemplar la incertidumbre en este tipo de problemas.

Finalmente, relativo al cálculo de rutas incluyendo incertidumbre y la modificación de la metodología Simheuristics propuesta, es necesario realizar estudios en otros dominios en los que se disponga de más información que pueda ser empleada para la generación de modelos de predicción de la de demanda de forma certera. Asimismo, sería posible la predicción de otros parámetros distintos a la demanda de los clientes, es decir, incluir incertidumbre en otros parámetros del problema como pueden ser los arcos entre clientes (en este caso relacionado con el tráfico) y realizar la optimización de rutas teniendo en cuenta estos parámetros y las predicciones realizadas para estos.

---

## CAPÍTULO 6

---



**VNiVERSiDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

## RESEARCH OVERVIEW

---





## Capítulo 6 Research overview

This chapter is an English overview about all thesis work carried out, from the introduction, objectives and methodologies to the obtained conclusions after results of case studies.

### 6.1 Introduction

In the last few years, the number of cities including new systems based on the Internet of Things (IoT) has increased in order to obtain new data of the city as well as offer new services and optimize the energetic consumption based on these data. These cities are moving forward a Smart City paradigm [Silva et al. 2018] of which the main purpose is making cities more sustainable and better places to live in..

Among all the applications emerged for Smart Cities there are applications aimed to monitor the flow of people in cities [Alvarez-Campana et al. 2017], monitor parking [Bagula et al. 2015], get information about accessible places [Mora et al. 2017], monitor lighting in domotics [Vinagre et al. 2016] and street lighting [De Paz et al. 2016] together with the use of Smart Grids, management of water, waste and health services, intelligent logistics, etc.

They can be used in both big cities and rural areas, where it is increasingly common the use of applications related to precision crop management, Smart Farming [Villarrubia et al. 2017b] and intelligent control of cattle [Musat et al. 2018]. Meanwhile, the rise of new sensors along with the IoT concept and the daily use of objects connected to the Internet allows the development of this kind of projects. These sensors or nodes are clustered in networks known as Wireless Sensor Networks (WSN) [Buratti et al. 2009]. These wireless networks can be deployed in rural and urban environments, and they provide the basic infrastructure to obtain data in applications for Smart Cities. Researches on the use of this group of sensors and the WSN to obtain data together with efficient energetic consumption are essential nowadays. A large number of applications employed in Smart Cities are focused on improving the energetic efficiency of these sensors to provide them with a long-life battery able to be working for several years. Among the applications related to Smart Cities, those involved in intelligent logistics are remarkable: the distribution of assets management, waste collection management systems [Gutierrez et al. 2015], bike sharing systems [Pal and Zhang 2017][Lin and Chou

2012], etc. These applications are designed to optimize available resources while offering citizens an appropriate service. Like the rest of logistics applications, those focused on Smart Waste Collection include three essential elements to provide these systems with intelligence and efficiency: efficiency of the data collecting system, modelling and demand forecasting based on the collected data, and route optimization system.

Firstly, the system must be equipped with efficient sensors of which maintenance does not present any problem when implementing them. Furthermore, appropriate WSNs are required to allow the development of low-consumption sensors. Secondly, the demand of these systems should be modelled. This demand refers to the number of assets that will be required to be collected or delivered depending on the specific logistics system. Formally, these systems are often modelled as Vehicle Routing Problems (VRP) [Eksioglu et al. 2009]. These problems include a fleet of vehicles aimed to meet the demand of some customers, subject to constraints and using the lowest possible number of resources. Knowing and modelling these problems demand is essential for optimizing routes later. Finally, methods usually employed in literature and available commercial software do not consider the use of the uncertainty in the solving methods for these problems. There is a growing line of research with an increasing number of works found in the literature that aims to reflect the existing uncertainty in the variable employed in the VRP. Among the various solving methods including the uncertainty to the problem, there are methods of stochastic optimization [Erkin and Koole 2017], robust optimization [Solano-Charris 2015] and new methodologies [Juan et al. 2015a] as an alternative and more flexible approach against the first two.

Consequently, there are three lines that require researching: (1) new ways of getting data with sensors that allow higher efficiency and battery life, (2) demand forecasting systems able to model demand and finally (3) optimization systems that use processed information and consider the uncertainty present in actual collecting problems.

### **6.1.1 Hypothesis and objectives**

Nowadays it is possible to find waste collection systems using sensors to measure their filling level, excluding weight measures, and offering an energy efficiency in terms of battery which could be improved with new communication technologies. In addition, it is required to research forecasting models able to forecast accurately the demand created by clients in vehicle routing problems. Finally, a significant number of works currently found in literature do not consider elements related to uncertainty in Vehicle Routing Problems VRP, so the aim of this paper is including these elements into the solving methods of such problems and advance throughout these lines of researches.

In consequence, the following hypothesis is raised as the basis of this doctoral thesis work:

*“The design of a Smart Waste Collection System based on low-cost and low energetic consumption wireless sensor networks, which includes demand forecasting systems and considers uncertainty in solving methods, generates savings in energy, costs, stuff and time against conventional logistics systems.”*

To validate this hypothesis a number of general purposes should be addressed. They are described below:

- Analysing the data capture for intelligent logistics systems studying wireless networks of sensors appropriate for long distances and sensors with high energetic efficiency while researching on the development of sensors that efficiently get data and produce a low energetic consumption.
- Studying the behaviour of the demand for logistics problems related to VRPs and design models able to forecast future demands accurately from the collected data and additional weather and time data.
- Considering uncertainty in Vehicle Routing Problems VRP parameters and including these problems solving methods that provide robust solutions considering stochasticity of these problems' variables in the actual world.
- Design a system that combines the foregoing and provides saving in costs, energy, and workforce against a conventional approach. The development of a case study in a real environment will be essential to obtain results and validate the hypothesis of the mentioned work.

In order to achieve these general aims, specific purposes must be defined to specify the tasks to be developed during the current thesis. These specific purposes are explained below:

- Study of the state of the art of wireless sensor networks for data captured in intelligent logistics systems.
- Study of the state of the art of low-power wireless sensor networks (LPWAN) and their application in smart logistics systems.
- Study of algorithms for predictive demand modelling in smart logistics systems.
- Study of the state of the art of VRP problems and their variants.
- Study of the state of the art of VRP problems with uncertainty.
- Design of a system which includes low cost and low power consumption sensors for data capture.
- Design of a system which includes prediction models allowing to model demand in intelligent logistics systems.
- Design of a vehicle routing optimization system which includes uncertainty in its solutions.

- Design and development of a case study that uses low power sensors and wireless sensor networks for long distances.
- Design and development of a case study focused on a demand prediction system.
- Design and development of a case study that includes systems mentioned above: data capture, prediction systems, uncertainty, and solving methods for Vehicle Routing Problems.

Succeeding in achieving these aims will allow to obtain and evaluate the results of every case study and confirm the hypothesis raised in this doctoral thesis work.

### 6.1.2 Motivation

The IoT rise has caused the inclusion of sensors in a great number of Smart Cities fields. This has resulted in creating WSN where nodes play a fundamental role in data capture and performance in the case of actuators. Battery life has imminently emerged as a constraint in the deployment of these sensors. Cases of use such as deploying sensors in containers, isolated weather stations or precision farming imply the inability to access to a power supply, so battery life must be as long as possible to reduce the maintenance costs of these systems. Consequently, new LPWAN networks [Sinha et al. 2017] have recently emerged to bring a solution to this constraint and allow the use of sensors equipped with batteries able to be working for several years. Developing sensors that use this kind of networks is essential to create systems that will later employ the sensors themselves in its working, such as intelligent waste collecting systems.

Directly relevant to this is the information provided by collected data from these sensors and the possibilities of use they offer to improve the collecting systems. Having historical data of waste produced in each container (called demand in VRP problems) together with other variables such as weather or annual calendar allows creating models able to predict with reasonable certainty. These models allow to anticipate possible container fillings and perform more efficient and intelligent collecting considering this information.

Finally, the information usually available in a problem is stochastic. However, this information is often considered as deterministic in this kind of problems in literature. Works in lines of research [Berhan et al.] intending to include stochasticity in variables of this problem have increased over the last few years. These variables are demand (in this case, amount of waste generated), the matrix of costs (for example, cost per journey from a node to another considering traffic) and clients (is it necessary to go to a specific container or not?). Demand is one of the main stochastic variables in waste collecting problems: the amount of waste that the truck will find in the container. Obviously, demand will be lower if all the containers of a system are equipped with sensors as recent information is available and the variability could be caused by other reasons such as to measure mistakes or waste outside the containers. Nevertheless, in the vast majority of cases, collecting systems are equipped with conventional

containers. In these cases, only historical data are available so that it is possible to predict containers demand based on created models and consider uncertainty for routes estimations.

Among different solving methods found in the literature to approach stochastic VRP problems, there are two main methods: stochastic optimization [Powell and Topaloglu 2003] and robust optimization [Bertsimas and Sim 2004]. The main problem of these methods are either they adopt a feasible but too conservative solution (robust optimization) or their methods for estimating solutions imply a rise in terms of computational costs and do not guarantee that the solution meets all the constraints (stochastic optimization). [Maggioni et al.] work provides a good comparison between both approaches. Alternatively, to these two solving methods, there are other methodologies in the literature [Juan et al. 2011] [Juan et al. 2015a] that address this problem from another perspective. They employ simulations to determine the feasibility of a solution created a priori with data and deterministic algorithms as well as its behaviour in a great number of scenarios generated through the probability distribution of each client to be visited. This allows using solution methods with heuristics widely studied in literature while evaluating the behaviour of the solutions under uncertainty.

This approach makes possible to provide the decision maker with tools to analyse different solutions and their behaviour considering uncertainty in the demands of VRP problem.

### 6.1.3 Methodology

Throughout the development of this thesis the methodology used is the one known as *Action Research* [Reason and Bradbury 2008]. This methodology has been widely employed in information systems and it is focused on action and change. This methodology raises and identifies the problem based on a hypothesis and starting from a number of concepts defined in a quantitative model of reality. After that, information about the concepts discussed along the hypothesis is collected and organised to be analysed and design a proposal to solve the problem or verify the stated hypothesis. Subsequently, three specific case studies are employed to implement the proposed solution. Finally, conclusions from the results derived from the research work are described.

One of the key points of this methodology consists in establishing a group of activities aimed to bring to fruition the goals initially set and the confirmation or refutation of the stated hypothesis. These activities consist of the following:

- Defining the existing problem: description of the problem and its variables to set the goals and the working hypothesis of the research.
- Reviewing state of the art: this section is aimed to review previous works and essential concepts related to the working hypothesis to establish a strong starting point for the research work.

- Analysing and designing the solution proposal.
- Designing case studies to validate every component of the proposed solution.
- Performing each case study and collecting the results.
- Studying, validating and discussing the results from the proposed case studies.
- Drawing conclusions from proposed case studies and their results.
- Disseminating and publishing the results in congresses and journals.

Since these activities are performed iteratively throughout the research process, this process can be considered as an iterative and incremental process like many software engineering methodologies.

Applying this research methodology results in a double benefit: on the one hand, applying the proposed solution in a real data environment provides results that are directly transferable to the market ready for use. On the other hand, new relevant knowledge is produced, and it can be transferred to other environments different from the one of this thesis.

While developing this thesis, scientific papers have been elaborated, and they include different case studies performed along the research process. These papers have been published in scientific journals with impact factor in the area of action of this doctoral thesis.

Scheduled tasks have been performed over time and milestones have been established to track them and measure their progress as well as take corrective measures if required.

Additionally, other activities are done during the research process —such as training courses related to the thesis thematic, attendance to seminars and conferences, contacting national and foreign researchers who have contributed ideas and new approaches— have enriched the conducted research. These other activities also include research stay in a foreign centre to broaden horizons and closer ties with other researchers who are currently working on similar research fields and whose perspectives have enriched the doctoral thesis work.

## 6.2 Background

This chapter comprises the state of the art of the technologies involved in the three essential parts of the proposed system, a Smart Waste Collection System: the data collection with Wireless Sensor Networks (WSN), the use of prediction models for demand forecasting from data previously obtained and, finally, the routes optimization for waste collection.

The first section of this chapter reviews the different kind of WSNs and sensor nodes employed in the literature as well as their main features. An analysis of different networks according to their energy consumption is also described. After this, a review of the main LPWANs used nowadays is presented, and each one of these is addressed for the proposed system.

The following section presents a review of the state of the art of different prediction models for demand forecasting in this sort of problems, together with algorithms which have had great results in this task.

Finally, the vehicle routing problem VRP is defined, the taxonomy of the problem is presented and the different variants and those which consider uncertainty in its definition are described. In addition, several solving methods including uncertainty are reviewed including the one selected for this work.

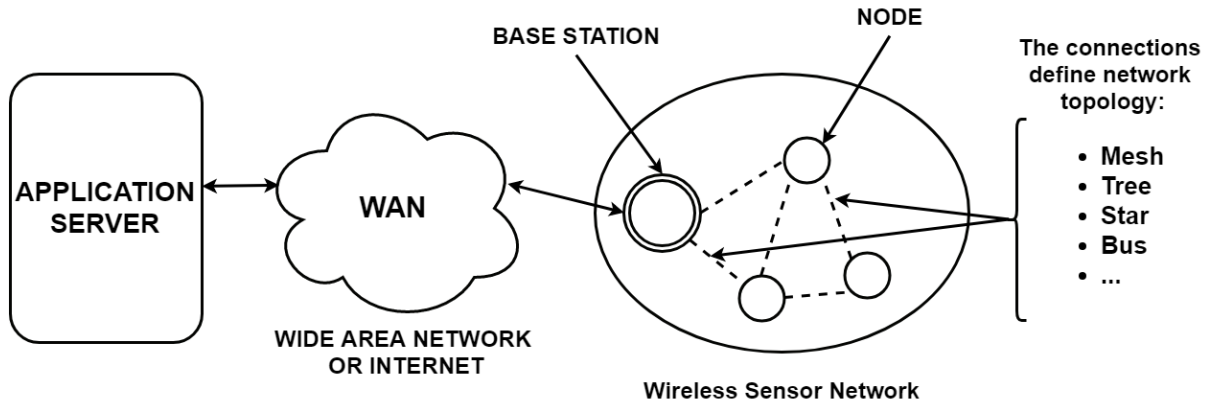
### 6.2.1 Data collection WSNs

#### 6.2.1.1 Wireless Sensor Networks

Wireless Sensor Networks have become very popular nowadays in the scientific environment due to the useful infrastructure that they provide for developing monitoring and control applications. A wireless sensor network (WSN) consists of a large number of devices known as nodes which are deployed to observe or measure certain phenomena or detect specific events and then act if the node has the capability of doing it [Mohamed et al. 2017]. The basic components of a WSN are:

- **Node:** It is a device which is able to obtain certain data through a sensor or a set of sensors and could be equipped with actuators for performing some specific actions in the environment. It usually has a limited computing capacity, and it communicates wirelessly with other nodes or a base station often called gateway or sink.
- **Base station:** It is responsible for capturing the transmitted data by the nodes and forward them or even provide access to third applications. It may have storage and processing capabilities, which are nowadays increasing due to the rise of Edge

Computing [Roman et al. 2018]. It also may act as a gateway and provide access and control to nodes.



**Figure 1** Basic configuration scheme of a Wireless Sensor Network

**Figure 1** describes a basic configuration scheme of a wireless sensor network. The communication between the node and the base station or other nodes is performed via a wireless link. This communication could be performed directly between the node and the base station or via intermediary nodes which forward the data transmitted. The kind of communication between sensors and the base station defines the network topology, thus, several networks could be found such as mesh, bus, tree, star among others.

The WSNs usually employ Wireless Personal Area Networks (WPAN) or Low Power Wide Area Networks (LPWAN) standards for the communication between nodes and base stations. There is no single solution for deploying WSN and the proper network communication standard selection depends completely on the communication requirements and resource constraints of the specific final application [Pule et al. 2017a].

### 6.2.1.2 Low Power Wide Area Networks

Included in the group of WSNs, the use of LPWAN standards is increasing because they focus its operation on two main features compared with other types of WSN: high energy efficiency and long-range communication.

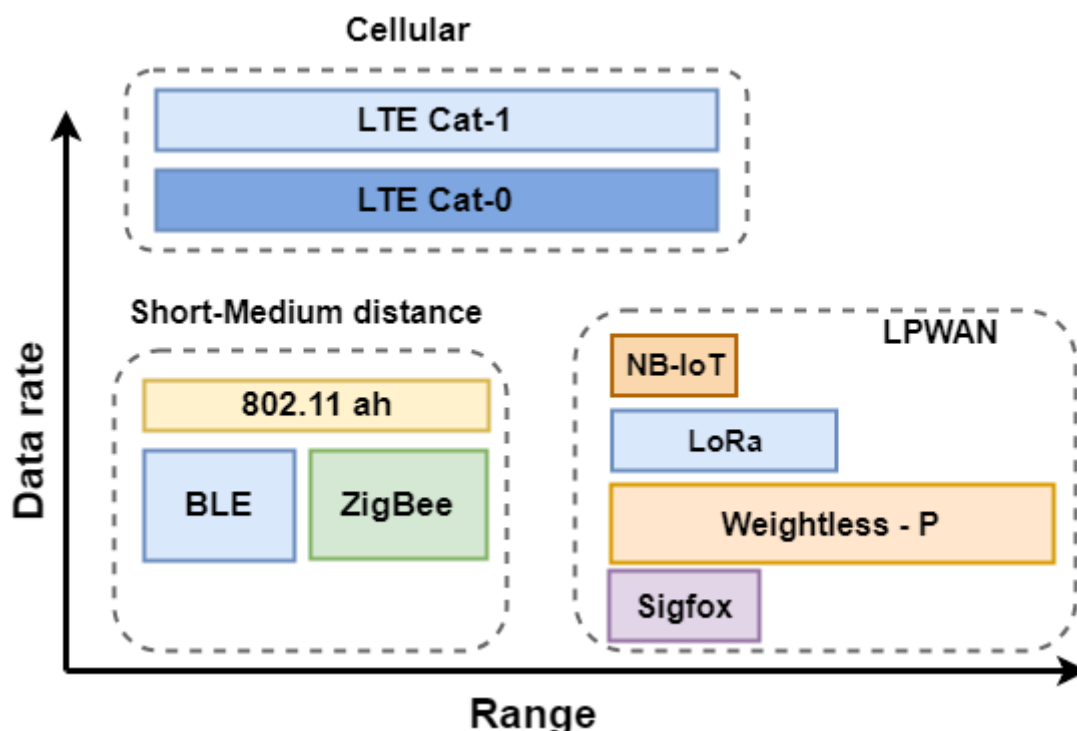
On the one hand, the high energy efficiency enables to develop nodes equipped with batteries allowing a working operation during several. This decreases the maintenance of the final devices and offers the option to place them in isolated locations without an electric energy source.



On the other hand, a high number of these technologies allow long-range connections, giving coverage to wide areas just by using a small deployment of base stations. This coverage factor combined with the energetic efficiency let install nodes in remote areas where there is no access to the electrical source nor GPRS coverage.

However, these technologies are particularly suitable for a type of applications which have specific requirements regarding the amount and frequency of data transfer. **Figure 2** presents a chart where different wireless technologies are placed together with the LPWAN standards taking into account the range and the data transfer in the communication. These technologies are oriented to applications which have a very low data transfer and wide coverage is needed, —in some cases, several kilometres.

Among all the technologies included in LPWAN, the following have to be highlighted: LTE-M, Sigfox, LoRa (including LoRaWAN specification) and NB-IoT [Wang et al. 2017].



**Figure 2** Wireless technologies depending on the data link range and the data transfer

Sigfox [Sigfox 2016] is an Ultra Narrow Band (UNB) communication network [Walker 2004]. As Figure 2 shows, this technology offers a great communication range and a very limited bandwidth. It has a marketing model similar to the one of a traditional enterprise of communication where an operator provides a signal to a big area or country and gets paid for access and transfers. The operator could be the Sigfox company itself (such as in France,

Germany and Spain [39]) or other companies belonging to the Sigfox Partner Network [SigFox 2017] (such as IoTNet in Croatia or VT-IoT in Ireland).

Regarding LoRa [Lora Alliance 2017] [Augustin et al. 2016a], on the one hand, it is a proprietary technology based on chirp spread spectrum (CSS) radio modulation. Semtech acquired this technology in 2012, and it currently produces transceivers which use LoRa. On the other hand, LoRaWAN [Casals et al. 2017] is a media access control layer protocol, employed for managing communication between LPWAN gateways and node devices. It defines system architecture and it is maintained by the LoRa Alliance [Lora Alliance 2017]. LoRaWAN uses the physical layer protocol defined by LoRa.

The LoRaWAN specification is open and, in contrast to Sigfox, a LoRaWAN network can be deployed free of charge for license or data transfer. Multiple platforms based on the LoRaWAN specification are available in the market, like Actility [Yang 2014], Lorient [LORIENT], LoraServer.io [LoraServerIO 2018] and Senet [Senet 2018], among others.

NB-IoT [Wang et al. 2017] is an LPWAN new technology for the Internet of Things developed by the 3<sup>rd</sup> Generation Partnership Project (3GPP). NB-IoT technology employs a licensed network spectrum currently used by different Internet Service Providers (ISPs). The network can be deployed in two different ways: “in-band” or “standalone” deployment. The “in-band” deployment takes place in the spectrum allocated to Long Term Evolution (LTE), using resource blocks within the LTE carrier (even their guard-bands). The other deployment, the “standalone”, implies the use of a dedicated spectrum. NB-IoT uses the Orthogonal Frequency-Division Multiple Access (OFDMA) modulation for the downstream channel and Single Carrier Frequency Division Multiple Access (SC-FDMA) for the upstream channel. Its main advantage, in terms of communication, over other LWPANs is that it is well suited for applications that need to have minimal latency and are required to communicate more frequently [Sinha et al. 2017]. It is currently being deployed throughout Europe [GSMA 2018], but its availability is being reduced to urban areas where several functional tests are being carried out. These provide the network, and the connection is set through classic subscription services.

Besides the technologies previously indicated, there are more technologies that are starting to become available in Europe, such as Random Phase Multiple Access (RPMA) by Ingenu and Weightless of Weightless SIG [Weightless 2017].

**Table 1** shows a comparison between several wireless technologies employed in the design of WSNs. Features such as range, consumption and availability are emphasized because they are essential factors for developing the device. The range of each technology, as well as its availability on the area where the sensor network is intended to be deployed, are especially important in semi-urban or rural areas. Some technologies such as Wi-Fi or cellular cannot be applied when requiring a greater range (Wi-Fi) or when there is no availability (it depends

on the cellular operator). Power consumed by radio modules both in sleep mode and active mode to transfer or receive data from the network is another key factor when selecting the technology to be employed for the sensor network. Increasing the device's battery life is crucial in applications like Smart Waste Collection Systems where maintenance tasks are performed every six months or even longer. According to these requirements, technologies such as Sigfox, LoRa and NB-IoT seem to be well suited for the system proposed in this work.

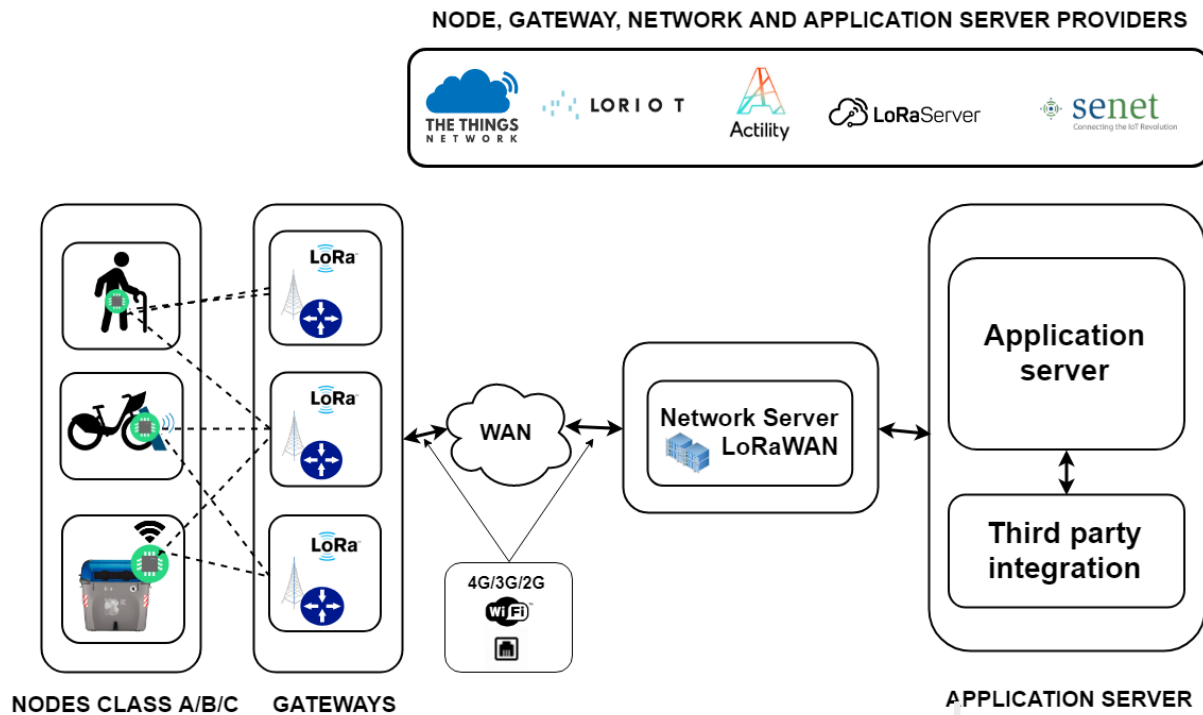
**Table 1** Wireless Sensor Network technologies comparison

Name	Range (km)	Standby consumption	Tx consumption	Modulation	Availability
RPMA	10-100	0.5 $\mu$ A	85 mA	RPMA+DSSS	Specific zones
Weightless P	2	0.7 $\mu$ A	<70 mA	GMSK+QPSK	Worldwide
ZigBee	0,20	3 $\mu$ A	30 mA	BPSK	Worldwide
LoRa	5-15	0.5 $\mu$ A	<90 mA	LoRa	Worldwide
Sigfox	3-10	0.5 $\mu$ A	<70 mA	BPSK	Worldwide
Celular	2-5	10 mA	800 mA	8PSK	Worldwide
WiFi ah	<1	-	<100 mA	QPSK/256QAM	-
NB-IoT	2-5	5 $\mu$ A	<100 mA	QPSK	Specific zones

On the one hand, NB-IoT is nowadays dismissed due to its current availability and the high cost of its modules. On the other hand, Sigfox technology is only available on a subscription basis, which means an extra cost for the developed solution. Consequently, the technology selected for this work is LoRa because of its commercial and technical characteristics.

Once this technology is selected, it is necessary to use a communication protocol over this technology. There are several alternatives in the literature [Ayoub et al. 2018] such as DASH7 [Haystack], Symphony [Symphony Link] and LoRaWAN. In this work LoRaWAN protocol has been selected because it has been employed successfully on previous works such as "An experimental performance evaluation of LoRaWAN over a real environment in Bangkok" by Vatcharatiansakul et al. [Vatcharatiansakul et al. 2017] and "A long-range wide area network-based smart pest monitoring system" by Yu et al. [Yu et al. 2017] among others [Catherwood et al.].

LoRaWAN is a MAC protocol build on the top of LoRa physical layer and it defines the communication protocol and the WSN architecture. This protocol is oriented to WSN where nodes perform very limited size information exchange and it is done every long-time period such as hours or even days. The communication protocol and the network architecture have a direct impact on the amount of energy consumed by final nodes, network capacity, quality of service, security and final applications. This protocol is developed and maintained by LoRa Alliance [Lora Alliance 2017].



**Figure 3** Basic scheme of LoRaWAN architecture

The minimal setup for a LoRaWAN network specified by version 1.0 of the protocol are nodes or final devices, base stations or gateways and network servers:

- **Node:** A low consumption device which communicates to gateways employing a LoRa radio link.
- **Gateway:** It is an intermediary device which is in charge of forwarding packets from nodes to the network server through a network interface backhaul IP like Ethernet WiFi or 3G/4G. A LoRaWAN deployment could include several gateways which could receive the same packet from the same node, and network server settles the duplication issue.
- **Network server:** Device in charge of decoding the information from packets, identifying duplicate packets sent by nodes and generating the response packets which must be sent to the nodes.

As opposed to traditional cellular networks, nodes are not bound to a single gateway for access to the network. The gateway's task is forwarding received packets from nodes to the network server and including some information inside packets about link quality between node and gateway. Thus, a node is bound to a network server which is in charge of detecting duplicate packets, choosing the proper gateway for sending a response if necessary. The gateways act transparently for final nodes. **Figure 3** displays a scheme of LoRaWAN architecture together with all the elements previously mentioned [Petäjärvi et al. 2017].

As previously described in the **Figure 3**, these networks present a “star of stars” topology where nodes communicate to gateways and these send data received to a network server. After this, network servers send information to the application servers which enable the communication between the WSN and third-party applications.

Together with this decision, another important step to be considered was the selection of the LoRaWAN network server (and all the software involved). There were several options to deploy the LoRaWAN architecture which can be summarized in three options:

- (1) **Free and open source solution:** A free and open source option like LoraServer.io [LoraServerIO 2018] which provides the software to implement the required features to be LoRaWAN compliant, specifically, the LoRa gateway bridge (packet forwarder), the LoRaWAN network server and the LoRaWAN application server. The maintenance and the hosting of the network server and some other application services are the responsibility of the user.
- (2) **Collaborative and free option:** It is like The Thing Network (TTN), [The Things Network 2016] which provides a self-hosted option, but it also provides a free and collaborative platform as a service (PaaS). That is, the platform is free, but every gateway using it can receive data from any node belonging to a TTN application. In addition, it includes integrations with third-party applications and communication protocols.
- (3) **Commercial option:** A final option is choosing a complete solution offered by companies like Actility [Yang 2014], Lorient [LORIENT] or Senet [Senet 2018], among others. In some cases, they provide turnkey solutions for the infrastructure, data storing and even data analytics.

**Table 2** presents options for the implementation of the LoRaWAN network server (and all the software involved) along with some of their features.

**Table 2** Several available options for a LoRaWAN network server

Name	Hosting	Open Source	Price plan
LoraServer.io	Self-hosted	yes	free
TTN	Self-hosted / 3rd party	yes	free / paid
Actility	3rd party	no	free (limited)/paid
Loriot	3rd party	no	free (limited)/paid
Senet	3rd party	no	free (limited)/paid

For this work, the option selected is TTN [The Things Network 2016] as LoRaWAN server network because of its collaborative deployment and free hosting option. This allows the use of the deployed network by all of the TTN community and provides a service that is integrated into third-party applications at the same time. Consequently, the deployment of a LoRaWAN network benefits the entire area where the system is implemented. It provides the region with an infrastructure to develop applications based on this kind of technology and therefore, improves the technological development of the area.

The system proposed in this work is decoupled from the LoRaWAN network server and application server so that any change in this element does not affect the other parts of the system, enabling changing this part of the system if required.

### 6.2.1.3 Application of wireless sensor networks (LPWAN)

Throughout this section, several of the main applications of wireless sensor networks in the literature, specifically LPWAN, are presented. These applications share the same interaction patterns which can be:

- **Event detection:** Detection and classification of events that trigger a subsequent action in the system making use of the WSN.
- **Periodic data collection:** Many of the applications consist of periodic data collection from sensors with a relatively low frequency that depends on the constraints of the network technology employed or the data to be collected.
- **Tracking:** Together with periodic data collection and event detection, a feature regarding tracking could be included if the end node is equipped with a geopositioning system or simply using the network itself to locate it. This will depend on the required location accuracy.

Next, a description of some of the main use cases employing this kind of LPWAN found in the literature is presented.

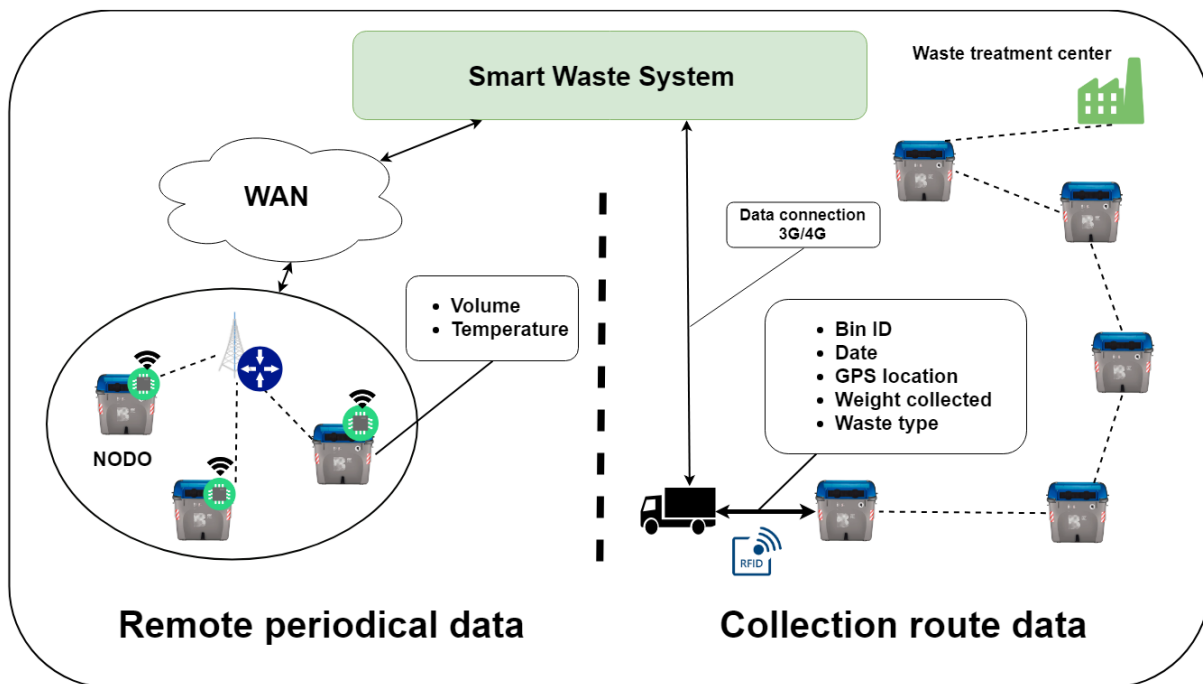
- **Health and monitoring:** The use of LPWAN networks for remote monitoring health and well-being of both patients and elder people has been widely studied in recent years

[Petäjärvi et al.]. Several works are available in the literature such as the one performed by Chen et al. [Chen et al. 2017] in which they develop a device to detect heart attacks and send an alert using LoRa.

- **Environmental monitoring:** Some works have been developed in this area such as the work done by Pule et al. [Pule et al. 2017b]. They describe how to employ different LPWAN for water quality monitoring. Other examples are the work done by Sujuan Liu et al [Sujuan Liu et al. 2016], which is focused on the air quality monitoring, or the work performed by Park et al. [Park et al.], focused on fire detection on forests.
- **Cattle monitoring and precision agriculture:** The use of LPWAN networks in the field of livestock and intelligent agriculture has recently become very popular and the literature presents many works about this: [Villarrubia et al. 2017a] [Barriuso et al. 2018] related to the control of livestock and the collection of soil data for creating smart systems that, for instance, detect and notify the imminent birth of a cow or accurately control the amount of water needed for irrigation.
- **Bike Sharing Systems:** Researches on sensor networks applied to collect data from bicycle loan systems in cities [Lundh 2017] regarding where bicycles are picked and delivered or its geographical position are also available in the literature.
- **Waste collection systems:** As well as in previous use cases, several works in the literature explore the application of WSNs to waste collection for obtaining useful data. The following section explains how these data are obtained in waste collection systems.

#### 6.2.1.4 Data collection in Smart Waste Collection Systems

Several methods to obtain information about the bin filling in a Smart Waste Collection System can be found in the current literature [Hannan et al. 2015]. Some of them are proposed by M.A Hannan et al. [Arebey et al. 2010], who employ RFID technology for bin identification in the system. The fleet of trucks is equipped with RFID readers, a GPS, a GPRS system and a built-in weighting system in the hook used to lift bins [Chowdhury and Chowdhury 2007]. During waste collection, each bin is firstly identified by its RFID tag. After that, the truck stores the current GPS location. Next, the waste amount of the container is weighted and, finally, all this information is sent to a central server. In works like the presented by [Karadimas et al. 2016], the bin itself obtains the container weight, whereas in other cases the weight is obtained by the operator: the bin is lifted by the hook and the operator weights the bin in order to obtain the amount of waste. Then, the bin is emptied, and the operator weights the container again in order to obtain the tare weight of the container. This kind of systems are usually installed in the current waste collection systems, in both rural and urban areas. The main advantage of this sort of systems is its low cost in terms of deployment because the cost is focused on the truck fleet rather than investing in the bin fleet. The major disadvantage of this kind of systems is that they are unable to obtain measurements of containers in real-time and on a regular basis, since it is only possible to obtain the measurements once the collection is being carried out.



**Figure 4** Scheme of kind of data collection in Smart Waste Collection System

This enables to obtain historical data about the data collection for each bin, but it does not allow to know in detail the state of the bins before collecting waste in them. Furthermore, it does not provide information about the daily waste generation. Right side of the **Figure 4** shows this approach of data collection in Smart Waste Collection Systems.

As an answer to this approach, authors such as Longhi et al. [Longhi et al. 2012] and Ramson et al. [Ramson and Moni 2017a], among others, propose the use of WSNs in order to obtain real-time data from bins (temperature, volume, weight, etc.) remotely (**Figure 4** left side). The current rise of small sensors and low-power consumption protocols enable the creation of new nodes which could be integrated into the bin for capturing data of weight or approximate volume or even eventual data of an incident, like an excess of temperature or a change in the last position of the bin, periodically. This kind of system starts to be integrated into the current Smart Waste Collection Systems. This is done in a sort of pilot projects with a limited number of bins and sensors which are progressively included in the whole fleet of bins. This allows using these data for better optimization of the collection routes and saving resources like workforce, fuel and time. However, many systems have a mixed fleet of bins (with and without sensors) and this thesis work aims to integrate the two kinds of data for a later demand prediction of each bin.



## 6.2.2 Demand Prediction

This section is focused on reviewing some of the state-of-art machine learning algorithms employed for demand forecasting using historical data in different case studies of the literature.

Next, several regression techniques are briefly described, and these will be used in the proposed case studies of this thesis work.

### 6.2.2.1 Random Forest Regressors

Random Forest is a machine learning algorithm which has its origin in decision trees and consists of a set of these, hence part of its name, the word Forest. This technique was developed by Leo Breiman in 2001 [Breiman 2001].

Therefore, it is a model that falls within the category of methods known as ensemble methods, which are a type of learning algorithms that in turn employ multiple automatic learning algorithms to achieve better results in terms of prediction than the results that may be achieved using only one of those algorithms.

This technique is based on decision trees, which can be used for both classification and regression tasks (classification or regression trees). Therefore, this type of ensemble can be used for either of these two tasks.

This technique has been widely used in the literature in very different fields obtaining good results. Several works from different fields can be found in the literature. For example, engineering in predicting water demand [Herrera et al. 2010] or electric power [Dudek 2011], e-commerce in predicting demand for the purchase of articles [Joshi et al. 2018], environmental and geospatial sciences [Joshi et al. 2018], financial studies [Kumar and Thenmozhi 2014], medicine [Kane et al. 2014], etc.

An extensive description of how they work in depth can be found in works such as [John Lu 2010] [Biau and Scornet 2015] [Verikas et al. 2011]

### 6.2.2.2 Extra Tree Regressors

This technique, proposed by [Geurts et al. 2006], goes one step further in the randomness inclusion process and adds an additional step to the process performed by the Random Forest algorithm. In this technique, the trees are trained in the same way as in the Random Forest algorithm with the exception that the process used to make the cut is done randomly, that is, instead of calculating the best characteristic/cut combination, for each characteristic that is

being considered, a random value is selected for the cut. This random value is selected from an empirical range obtained from the training set in question.

The main advantage of this technique over Random Forest is that it has a better computational performance.

### 6.2.2.3 Gradient Boosting Regressor

This technique is based on what is known as Boosting. This is another approach aimed at improving predictions made by a weak predictor, i.e. this technique can be applied to any automatic learning technique (such as, for example, a decision tree) for either regression or classification problems.

Boosting is an ensemble method similar to Bagging but, in this case, prediction models are not created independently, they are created sequentially instead. In this technique, subsequent prediction models learn from the mistakes made by the previous ones. By doing so, the samples of the training set have an unequal probability of appearing in subsequent models and those with the greatest error will appear more often. Consequently, the samples are not selected based on Bootstrapping—as in Bagging—but they are chosen according to the error.

The main idea is to filter samples from the training set, leaving those that the weak prediction model can handle and then create new models that focus on the rest of the samples that were difficult to handle by the previous models.

One of the first algorithms using this type of technique was Adaptive Boosting or AdaBoost [Freund and Schapire 1997]. In this algorithm, weak prediction models are decision trees with a single subdivision. Its operation is based on assigning different weights to the training samples, assigning more to those that are more difficult to classify and less on those that have been handled well, the new models created are added sequentially and focus their training on more difficult patterns. Predictions are made by voting and weighted by the weight of each of their precisions.

Later, Friedman et al. [Friedman and Friedman 2000a] developed the algorithm known as Gradient Boosting which consists of a statistical framework where Boosting is treated as a numerical optimization problem. Its objective is to optimize a model cost function by adding weak prediction models using a procedure such as Gradient Descent. This kind of algorithms are described as additive models by stages because a new weak prediction model is added in each step and the existing previous models in the model remain unchanged. Gradient Boosting involves three basic elements:

- A cost function which must be optimized: Depending on the task (regression or classification) the function can vary (Squared Error, LogLoss, etc). The great advantage of this framework is that it is generic enough to be used with different cost functions.
- A weak prediction base model for making predictions: Decision trees are usually used.
- An additive model that allows weak models to be added in order to minimize the cost function: Each model is added one by one and the ones previously included do not vary. A gradient lowering process is performed to optimize the cost function when adding a new model. Typically, gradient descent is used to minimize parameters such as coefficients in a regression equation or weights in a neural network, after calculating the cost, the weights are updated to minimize that cost. In this case, there are weak prediction models (in this case decision trees) instead of parameters. After calculating the cost, to perform the gradient descent process, adding a tree to the model that reduces the cost is required. This is done by parameterizing the tree and consequently modifying the parameters of that tree and moving it in the right direction (reducing the residual cost). This approach is called Functional Gradient Descent with functions. This addition process stops when a certain number of trees have been added or once the cost reaches an acceptable level or does not improve compared to a set of validation data.

From this basic framework, improvements can be incorporated into the creation of trees (limiting the number of trees, their depth, number of final nodes, etc.), the process of updating weights (acting on the learning rate), the use of random samples like in Random Forest, regulation processes, etc.

These techniques have been widely used in various works in the literature, for example, the prediction of electricity demand [Nassif 2016], prediction of travel times [Zhang and Haghani 2015] and prediction of waste generation [Johnson et al. 2017] among many others.

#### 6.2.2.4 Support Vector Regressors

This technique is totally different from the previous ones and is closely related to the ones known as Support Vector Machines. In fact, it is the application of this technique to a regression problem.

Support Vector Machine (SVM) [Cortes and Vapnik 1995] is a supervised learning algorithm developed by Vladimir Vapnik and used mainly for classification tasks. Its operation is based on the construction of a hyperplane or set of these in a high-dimensionality space which separates the classes from the training set. A good separation of these classes is achieved by the hyperplane that presents the greater distance to the samples of the training set of any class since if the distance is greater, the error of generalization of the classifier will be smaller. On the other hand, in data sets that are not linearly separable in their dimensional space, the

application of what is known as the “kernel trick” [Boser et al. 1992] is required. Doing so, these data can be mapped to a much larger dimensional space where this separation is possible.

The Support Vector Regression (SVR) technique [Drucker et al. 1997] uses the same principles as SVM for the problem of classification with some differences. The idea is to select the regressor hyperplane that best fits the training dataset. In this case, there are no classes that should be separated. However, the idea is based on considering a margin of tolerance  $\varepsilon$ , so that all samples of the training set are expected to be present in that band or "tube" around the hyperplane aimed to be calculated. That is, the distance from the hyperplane is less than  $\varepsilon$ . When defining the hyperplane, only the samples of the training set that are more than  $\varepsilon$  away from the hyperplane will be taken into account, and they will be considered the support vectors in this case.

A broad description of this technique can be found in the paper [Smola et al. 2004].

### 6.2.2.5 Artificial Neural Networks

In the field of artificial neural networks, it is possible to find from the so-called perceptron, the simplest model consisting of a single neuron (perceptron), to complex structures with multiple neurons grouped in layers forming different architectures.

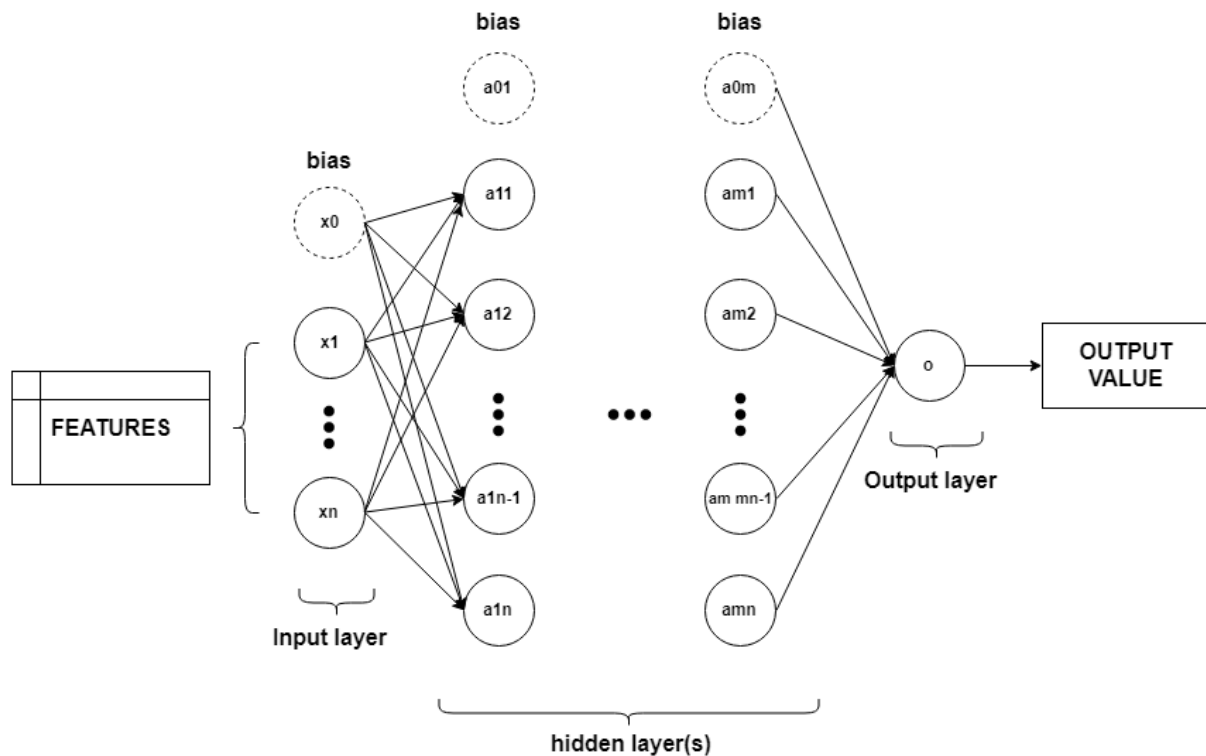
These techniques are used for both regression and classification tasks, and their power comes from their ability to learn the representation of training data and how that data relates to their output variable.

According to the universal approximation theorem [Kolmogorov 1957] [Cybenko 1989], a neural network with a single hidden layer and a sufficient number of neurons in it can approximate any nonlinear function  $f(\cdot): R^m \rightarrow R^o$  where  $m$  is the number of dimensions for the input and  $o$  the number of dimensions for the output.

As the simplest architecture used in literature, it is possible to find the Multi-Layer Perceptron (MLP). It is possible to see a representation of this type of neural network. The layer on the left, known as the input layer, consists of a set of neurons  $\{x_i | x_1, x_2, x_3, \dots, x_m\}$  representing the input characteristics and  $x_0$  the constant term known as bias.

Each neuron in each hidden layer transforms the values of the previous layer using a weighted sum of the form  $w_0x_0, w_1x_1, w_2x_2, \dots, w_mx_n$  followed by a nonlinear activation function  $g(\cdot): R \rightarrow R$  (such as sigmoid or hyperbolic tangent among others). There may be multiple hidden layers until you reach a final layer known as the output layer.

This last layer produces the output values of the network, so it may have one or more output neurons. This type of networks is known as feedforward networks because the connections are always to the next layer.



**Figure 5** Simple diagram of the architecture of a multilayer perceptron (MLP)

There are other neural network architectures such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), among many others, which include different types of connections with properties oriented to different problems. This type of techniques has a very expensive training process, but in recent years, the field is known as Deep Learning, these techniques have had much boom due to increased processing capacity and new neural network architectures with more and more layers (deeper) and more complex (architectures with different types of connections and neurons).

These techniques have been widely used in the literature in works such as [Shahabi et al. 2012], [Adamović et al. 2017] [Abbasi and El Hanandeh 2016] [Kannangara et al. 2018] for solid waste prediction.

### 6.2.3 CVRP, uncertainty and Simheuristics

This section deals with a state-of-the-art review of the problem called the Vehicle Routing Problem (VRP). The optimization of waste collection routes in Smart Waste Collection Systems is presented as a problem of this type, where the main objective is to minimize collection costs by looking for the least costly routes for the fleet of available vehicles and containers to be collected with their respective demands (amount of waste).

Throughout this section, the formal definition of the VRP problem is presented as well as the solve methods, the different variants that exist on the classic problem and, within these, those that contemplate the uncertainty in the problem. Finally, VRP problem solving approaches which incorporate uncertainty such as stochastic optimization, robust optimization or other new alternatives such as Simheuristics methodology are detailed.

#### 6.2.3.1 VRP formal definition

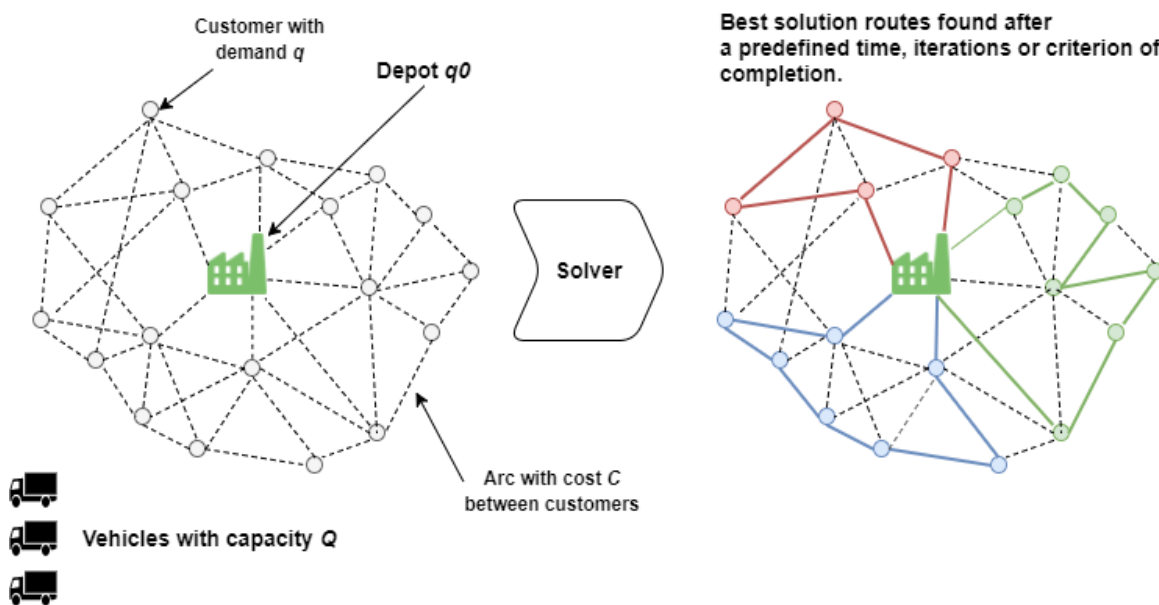
This kind of problem is known in the literature as VRP (Vehicle Routing Problem) [Dantzig and Ramser 1959] [Braekers et al. 2016]. It has many applications such as routes for courier companies, transport for people, the supply of goods, etc. In addition, this problem features many practical applications, but with a high complexity [Lenstra and Kan 1981] since it is a problem NP-Hard. This is because the space of solutions grows exponentially when the number of nodes increases, and an exact algorithm cannot find an optimal solution for large instances in reasonable finite computation time. This is the reason why the development of approximate solving methods such as heuristic ones or metaheuristic ones have proliferated.

The name VRP is used for a whole category of problems aimed at finding a number of routes for a set of vehicles located in one of the nodes called depot. These vehicles must meet the demand of a geographically distributed set of nodes and the main objective is finding the routes that satisfy those demands while minimizing the total cost of all the involved routes. These routes must start and end in the same depot. The most classic version of the VRP problem is the one known as CVRP (Capacitated Vehicle Routing Problem) in which there are capacity constraints for fleet vehicles, i.e. the entire fleet has a uniform capacity and there is only one depot.

**Figure 6** shows the approach of a classic CVRP with a homogeneous fleet. This is formally defined with the following description: given  $G(V, A)$ , an undirected graph where  $V$  indicates a set of  $n + 1$  vertices.

$A$  is a set of arcs, defined as  $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ .  $V$  is a set ( $V = \{0, 1, 2, \dots, n\}$ ) that defines the number of customers (in this case, containers). Each  $(v_i, v_j)$  of set  $A$  has an associated cost (not negative) ( $C_{ij}$ ). Each customer of the set  $\{1, \dots, n\}$  has a demand (in this

case, the quantity of waste) ( $q_i$ ) that must be collected and transported to the depot ( $q_0$ ). The demand of the depot is always 0, so  $q_0 = 0$ . A set ( $m$ ) of vehicles with the same capacity ( $Q$ ) (if the fleet has different capacities, the problem would be another kind of VRP problem) must be employed to collect waste of  $n$  customers (containers). The  $m$  vehicles must start and end their routes in the depot. A route is defined as a lower cost cycle of the graph ( $G$ ) that passes through the depot and the total demand of the set of vertices must not exceed the total capacity of the vehicle. The purpose of the problem is to reduce the distance, time or cost of the  $m$  vehicles while meeting the following requirements: (1) the depot is the start and end point of every route; (2) every customer is visited only once by just one vehicle; and (3) the total demand of each route does not exceed the capacity ( $Q$ ).



**Figure 6** This figure shows a graphic representation of the input of a Capacitated Vehicle Routing Problem (CVRP) and the output route solutions.

The classic CVRP is stated as follows:

$$\min \sum c_{ij}x_{ij} \tag{1}$$

with  $(i, j) \in V, V = \{0,1,2,3 \dots, n\}, x_{ij} \in \{0,1\}$ , subject to

$$\sum x_{0j} = m \tag{2}$$

$$\sum x_{i0} = m \tag{3}$$

$$\sum_{j=1} x_{ij} = 1 \quad (i = 1, \dots, n) \quad (4)$$

$$\sum_{j=1} x_{ij} = 1 \quad (j = 1, \dots, n) \quad (5)$$

$$m \geq 1 \quad (6)$$

The target function, Equation (1), is the total cost of the solution. The constraints of Equations (2) and (3) indicate that  $m$  is the number of vehicles employed for the solutions and state that all those vehicles must return to the depot. The constraints of Equations (4) and (5) turn each client into intermediate nodes of a route, and the constraint of Equation (6) states that the vehicle fleet employed must be greater than 1.

This formal definition corresponds to the most basic version of a CVRP, since depending on the constraints of the problem this will belong to another category within the multiple variants of this problem. These variants are reviewed in the following section.

### 6.2.3.2 VRP variants

The following are some of the many variants of the CVRP in the literature, the authors Toth and Vigo make a more detailed and extensive review in their book [Vigo and Toth 2014]. This section addresses a general review of the different typologies in order to frame the specific typology that will be tackled in this work.

The main VRP variants which can be found in the literature are:

- **Multiple Depot VRP (MDVRP):** It is an extension of the classic VRP in which the difference lies in the possibility of incorporating several depots. Wren and Holliday [Wren and Holliday 1972] described this problem for the first time.
- **Periodic VRP (PVRP):** It refers to a VRP case in which a number of delivery routes are designed over a given period, as opposed to all other cases in which routes are planned for a single day. Several works such as Christofides et al. [Christofides and Beasley 1984] or Francis et al. [Francis et al. 2008] deal in depth with this problem.
- **Split Delivery VRP (SDVRP):** It is a variant of the VRP in which customer demands can be divided into smaller ones. In this problem, the customer demand could be greater than the capacity of the vehicles; for this reason, customers could be visited more than once by different vehicles. These types of problems have been studied initially by Dror et al. [Dror and Trudeau 1989] and later by Archetti et al. [Archetti and Speranza 2008].



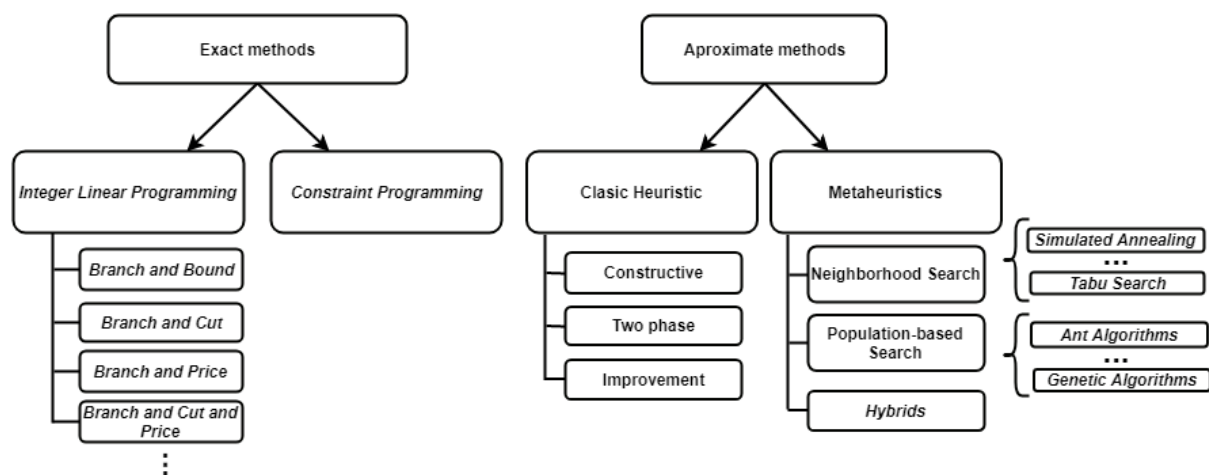
- **VRP with Backhauls (VRPB):** This kind of VRP deals with two types of customers. The ones are known as linehaul customers to which certain good must be delivered from the depot, and the ones known as backhaul customers, whose goods are collected and taken to the depot, that is, the opposite process takes place. In this problem, firstly, the goods are delivered to the clients and then the collections are made. They are extensively studied in the work of Parragh et al. [Parragh et al. 2008]
- **VRP with Pick-Up and Delivering (VRPPD):** This variant of the problem differs from the previous one in that customers can return goods at any time and trucks are considered to have enough space to carry them. Authors such as Gribkovskaia et al. [Gribkovskaia et al. 2008] have worked on solutions to this type of problem.
- **VRP with Time Windows (VRPTW):** This is one of the most widely used variants in literature. In this variant, Solomon et al. in 1987 [Solomon 1987] introduced a window of time in which each of the customers had to be visited. Therefore, vehicles must meet the demand of each customer within their time window. The aim of this variant is to minimise the vehicle fleet, travel time and waiting time. This last time refers to the situation in which a vehicle arrives at a customer before the lower limit of its time window, causing a wait and extra time along the entire route. However, solutions, where vehicles arrive after the upper limit of a customer's window, are not considered feasible. Many authors have studied this type of problem and its variants such as Taillard et al [Taillard et al. 1997], Liberatore et al [Liberatore et al. 2011], Hashimoto et al [Hashimoto et al. 2010] among others.
- **Heterogeneous Fleet VRP (HFVRP):** This variant of the problem includes vehicles with different capacities and with different associated costs. It has been widely studied in the industry as many of the vehicle fleets are not completely homogeneous. Likewise, among this type of problems, there are those that consider an unlimited fleet of vehicles and those that, on the contrary, have a limited fleet of vehicles.
- **Stochastic VRP (SVRP):** It provides the inclusion of uncertainty in different parts of the problem such as whether or not there are customers, the amount of customer demand or the costs associated with arcs (travel time). This variant of the problem is the one that will be studied in the proposed system of this paper.

### 6.2.3.3 Solving methods

As previously mentioned, there are many different types of VRP problems within the family. Depending on their variants, the formulation of each problem may vary as well as their solving methods. This section generally describes the approaches followed in the literature to find solutions to these problems.

The solving methods change depending on whether the objective is to find an optimal solution to the problem or a suboptimal solution in a reasonable time. It is, therefore, possible to make a first division between the solution methods known as exact, which guarantee an optimal solution to the problem by conducting an intensive exploration of the search space, and the approximate methods, in which a good enough solution is sought in a reasonable computational time. The approximate methods are more widely used in the literature due to the complexity of these problems. In instances with a slightly high number of customers, the exact methods are no longer applicable. Among the approximate methods, there are, on the one hand, heuristic methods, which usually obtain a good solution, but they do not guarantee that the solution is the optimal one since they can be stopped at local minimums. On the other hand, there are metaheuristic methods, a series of techniques that do not depend on the specific problem and allow to explore the search space in pursuit of solutions close to the optimal one in an iterative way, allowing to guide the search process of other heuristics.

**Figure 7** shows an outline of the different solving methods used in the literature to solve VRP problems.



**Figure 7** General scheme of solving methods for VRP

### 6.2.3.4 Uncertainty in CVRP

In the previous sections, the typologies of VRP have been described as well as several techniques for the resolution of the deterministic version of this problem. However, these problems in a real application environment present uncertainty in their parameters, so introducing this uncertainty in the solving methods is required.

As seen above, the family of VRP problems that considers the uncertainty in their formulation is the so-called SVRP (Stochastic VRP). It is generally defined as a VRP where one or more of the parameters are stochastic and follow a known probability distribution.

In the literature, this type of problem is in turn subdivided according to where this uncertainty is introduced:

- **VRP Stochastic Clients - VRPSC:** In this case, if the client will be attended or not is not known, in other words, each client presents a deterministic demand and this one may be present or not for elaborating the routes.
- **VRP Stochastic Demands - VRPSD:** This is one of the most frequent versions and it is a problem in which the exact demand amount of each of the clients is unknown, just a probability distribution of demand for each of them is known.
- **Stochastic clients and demands:** This version of the problem considers a combination of the previous two versions.
- **VRP Stochastic Travel Times - VRPSTT:** This version includes uncertainty related to traffic, climate and skills of drivers in the VRP problem resulting in stochastic values of travel costs from one place to another.

Another important factor of these problems is the moment when information is revealed, that is, when it becomes deterministic. Depending on this, the solving methods will vary as changes could be made to the routes calculated based on the new information obtained. In this work, the VRPSD problem is considered the starting point and the information is assumed to be revealed when a client is visited.

The kind of solving methods is also specific to this problem's family. Among the different types of solving methods that have been developed to deal with these problems, the main methods found in the literature are:

- **Stochastic programming with recourse:** This approach was developed separately by Dantzig [Dantzig 1955] and by Beale in 1955. The key idea of this approach is dividing the problem into two distinct stages between which the information is revealed. The most basic case of this approach comprises only two stages. In the first one, a solution able to address the consequences resulting from the information later revealed is

implemented. The second stage deals with the so-called recourse actions or "corrective actions" that are executed to adapt the routes after obtaining the deterministic information. It is important to point out that corrective actions (second stage) depend on the decision taken in the first stage and the information later revealed. Therefore, this type of techniques seeks to find solutions for the first stage which optimize the expected value of all decisions.

- **Dynamic programming:** It was proposed by Bellman in 1958 [Bellman 1958]. This approach is developed to effectively manage sequential decision problems. These methods are based on the temporal subdivision of the problem into stages and attempt to exploit the principle of optimality. These approaches are good for problems with a limited number of possible states and actions to be carried out.
- **Chance-constrained programming:** It was proposed by Charnes and Cooper in 1959 [Charnes and Cooper 1959]. The key idea of this approach is to allow some of the constraints to be met only with a certain probability.
- **Robust optimization:** In this approach, uncertainty is represented by the fact that the vector of stochastic parameters must belong to a polyhedral array (without any defined probability). For example, in a VRPSD, having a set of upper and lower limits for each demand along with an upper limit of total demand. These types of approaches are the most conservative and seek a solution that provides the best "worst case". The main disadvantage of these types of approaches is that the solutions they provide are too conservative, although they ensure the feasibility of the route. A comparison between this approach and stochastic programming can be found in the work presented by [Maggioni et al.].

In addition to these classic solving methods, new solving methodologies have recently emerged, such as the one described in the next section.

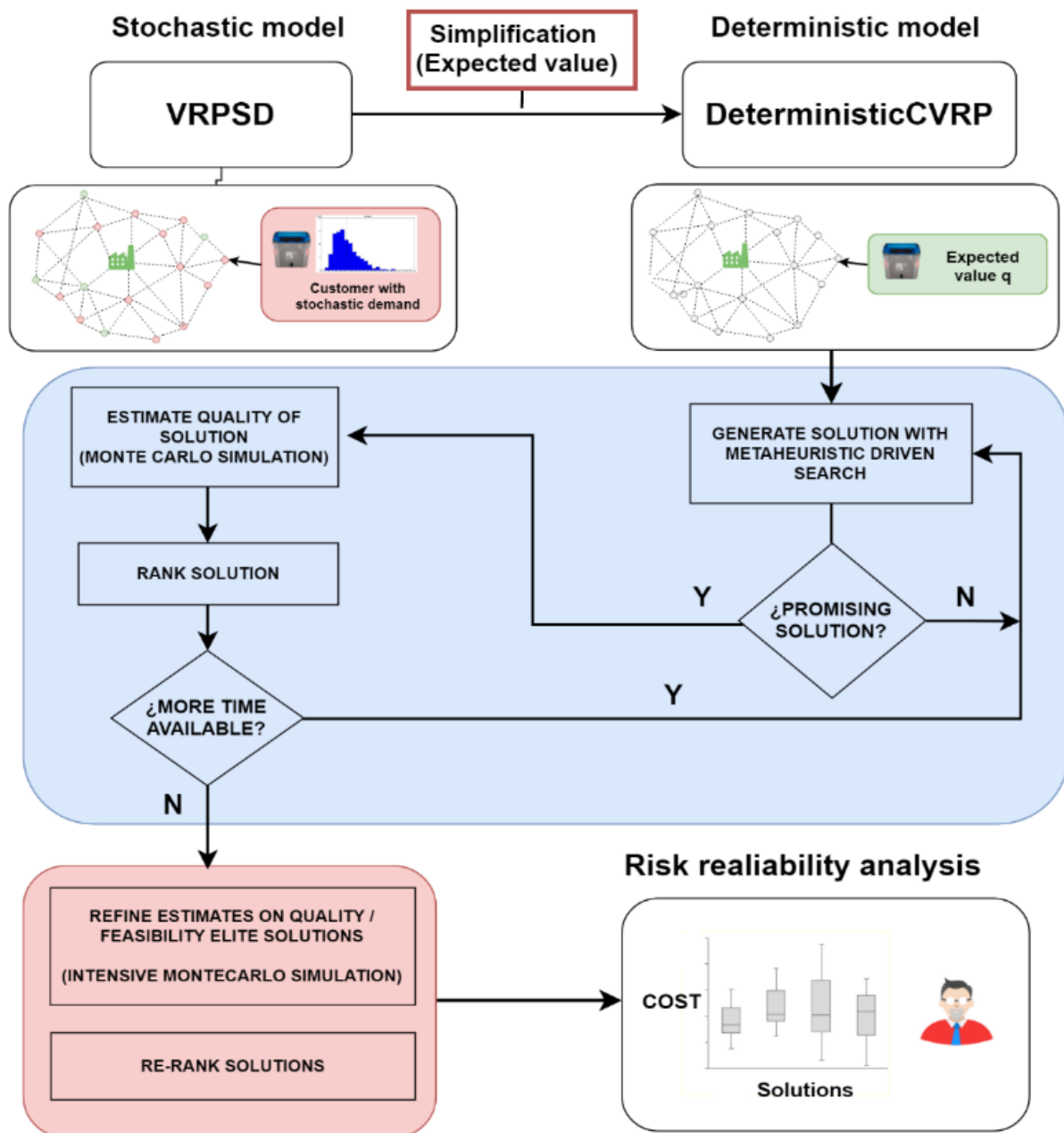
### 6.2.3.5 Simheuristic methodology

In recent years, a methodology proposed by Juan et al. [Juan et al. 2011] has emerged. It is named Simheuristic [Juan et al. 2015b] and it addresses SVRP problems from a different methodology to the one seen previously: it combines optimization and simulation techniques.

This approach assumes that, in scenarios with moderate uncertainty, high-quality solutions valid for a deterministic VRP problem may also be high-quality solutions valid for its stochastic version. Certainly, this does not imply that the best solution for the deterministic problem is the best one for the stochastic problem. The authors state that this methodology should not be applied in scenarios with a very high variability due to the great number of changes that can exist between the deterministic and the real scenario.

This premise in the relationship between solutions on the stochastic plane and the deterministic plane allows the authors to generate a series of high-quality promising solutions for the stochastic version of the problem through a deterministic version of the problem. This deterministic version is achieved using a simplification of the stochastic problem by transforming its stochastic demands into deterministic variables through, for example, the expected value (this work explains in detail how this point of the methodology is proposed to be modified). Next, solutions are searched using an efficient solution method for deterministic problems (algorithms of state of the art previously seen) that iteratively provides high-quality solutions that will be later evaluated. This evaluation consists of the estimation of the obtained solution quality and feasibility using scenario simulation methods to manage the randomness of these variables.

These scenarios requirements can be met either by the deterministic solution or, on the contrary, by applying a recourse action that will increase the cost of the solution and decrease its feasibility percentage concerning all the scenarios evaluated in the simulation. The estimated values of this stage allow creating a ranking of solutions for the stochastic problem. Once the time for searching solutions has finished, a scenario simulation is carried out—this time, with a higher number of replicas—, aiming to elaborate a new ranking of the solutions and provide the decision maker with an overview of those elite solutions and their probabilities so that he can do a risk/reliability analysis of them. This last point is a particularly useful tool that offers some approaches, like the Simheuristic approach, to decision makers enabling them to evaluate which solution they should apply depending on the risk they are willing to assume.



**Figure 8** General scheme of Simheuristics methodology proposed by Juan et al.

**Figure 8** shows a diagram of the methodology proposed by Juan et al. [Juan et al. 2015b] described above, where the simplification process of the stochastic problem that will be modified in the system proposed in this thesis work is highlighted.

## 6.3 Proposed system

This chapter presents the proposed system in this doctoral thesis work, a waste collection system which aims to address the hypothesis of this work. It includes a system for capturing data through wireless sensor networks that could be incorporated into the current collecting systems, a system for predicting the waste production from the data previously captured, and a system for optimizing the collecting routes that include the current uncertainty in waste collecting.

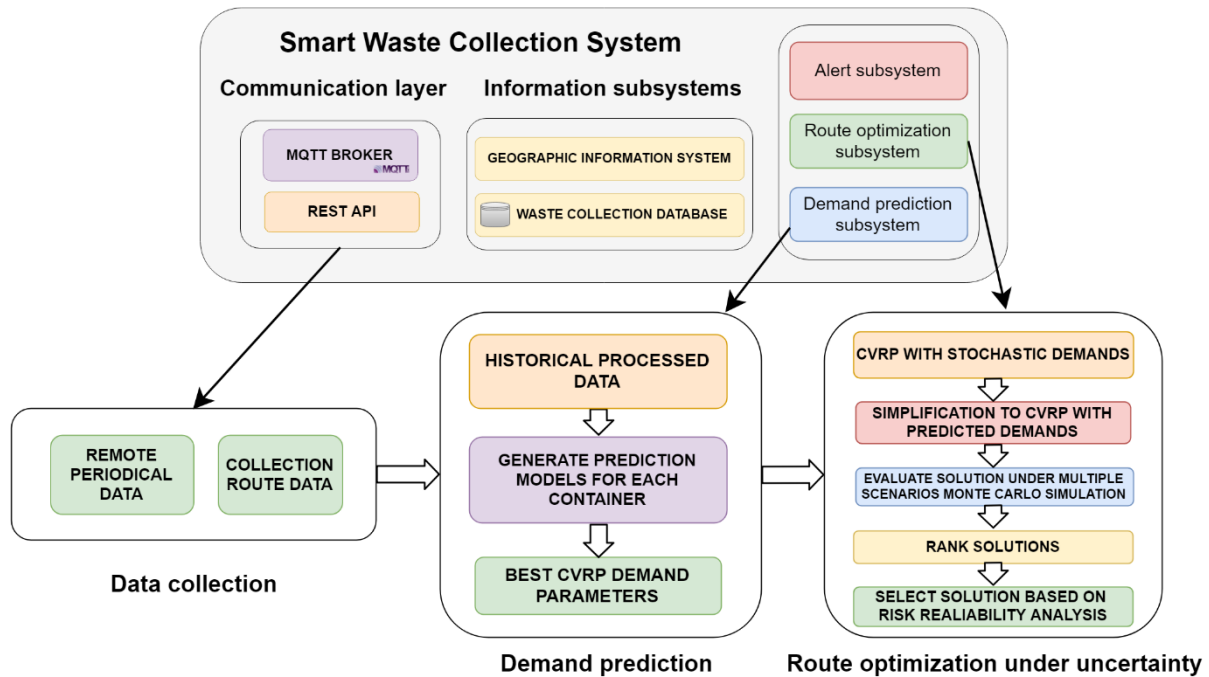
Firstly, the general architecture of the system will be described by identifying the different part of the design. After that, every part of the system will also be described in detail, the working proposal will be explained as well as how the techniques —previously reviewed in the state of the art— are employed.

### 6.3.1 System architecture

The proposed system covers three fundamental areas: obtaining the data by a network of sensors specially designed for a waste collection system, the use of these data to generate predictive models to obtain the most accurate demand values and the subsequent use of these in the optimization of the collection routes including methodologies that handle the uncertainty in the demands. **Figure 9** shows a general schematic diagram of the proposed system. At the top is a diagram a Smart Waste Collection System with different parts is displayed:

- **Communication layer:** For obtaining data from sensors and providing data produced in the system. This includes data generated by the containers of the system and obtained both in a remotely way or during the collection itself.
- **Information subsystems:** Each one develops a specific function in the system. The following ones are highlighted:
  - **Geographic information system:** This system provides the other subsystems with geographic data needed to optimize and create waste collecting routes.
  - **Waste collection database system:** It stores information related to each container, sensor, vehicle and waste collecting route of the system. Historical data obtained by the sensor networks and done collecting routes are also stored.
  - **Alert system:** This system monitors data captured by sensors and generates alerts about improper functioning or extreme values in temperatures of a given container.
  - **Demand prediction system:** This system is responsible for predicting the waste amount of each container in the following collections. Results of this system will be employed by the optimization system to elaborate on collecting routes.

- **Route optimization system:** This one is designed to optimize collecting routes including methodologies that incorporate uncertainty into the demand of each container and using predictive models produced in the demand prediction system.



**Figure 9** General diagram of the proposed system

In the bottom section of the same figure, there is an outline of the three main parts of the proposed system which are the focus of this thesis:

- **Data collection:** Both remotely or during the waste collection itself (bottom left part of **Figure 9**) including these two kinds of data in the system. A WSN is proposed to obtain data from containers together with a low-consumption sensor able to get values of weight, volume and temperature.
- **Prediction of the demand:** Predictive models of the waste amount that each container will present are generated from captured historical data and other sources data (calendar, weather conditions, etc.) to obtain useful predictions for the subsequent route optimization. These are represented on the central bottom part of **Figure 9**.
- **Route optimization under uncertainty:** This part presents a modification of the proposed methodology introduced by Juan et al. [Juan et al. 2011] in order to optimize collection routes by including uncertainty in demand and employing predictive models (bottom right of **Figure 9**).

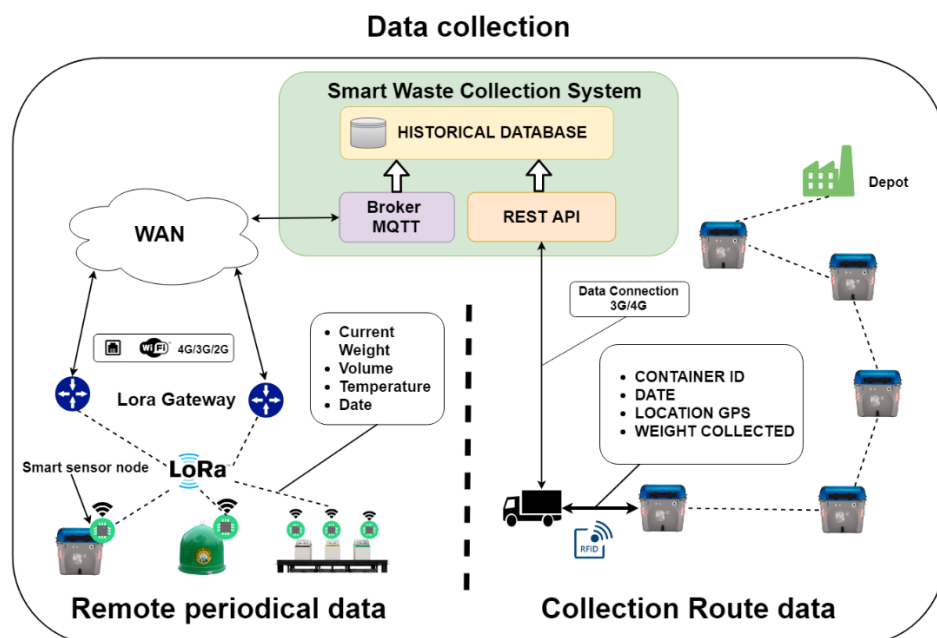


Next, these three parts of the proposal are described in detail: their functioning and how they interact with the other elements of the system.

### 6.3.2 Collection data subsystem

This section presents a collection data subsystem that collects data related to the waste containers and integrates information of different sources. In general, there are two kinds of data obtained from two distinct sources: periodical data about the waste container filling captured remotely, and filling data obtained when collecting waste. **Figure 10** shows a general diagram of the ways to collect data.

On the right side of the, there is a scheme of the collection data method applied when collecting waste. This method is the one incorporated into the conventional waste collection systems. In this process for collecting data, every vehicle identifies the container to be collected using an RFID reader. To get the container weight, the worker weighs it just before placing waste in the vehicle through the load cells located in the hook of the truck. Once the waste is in the truck, the worker weighs the container again, so that he obtains the weight of the container when it is empty. After that, the information related to the container identifier, GPS location, collection data, and weight is sent to the Smart Waste System through a GPRS connection and a REST web services API. These data are later store in the database system of the Smart Waste System, so they can be subsequently used by the other elements of the system.



**Figure 10** General diagram of the collection data system

On the left side of the **Figure 10**, there is a diagram of the wireless sensor network proposed in this work as the means of collecting data remotely and periodically.

Data related to container volume, weight and temperature are periodically collected. However, if unusual temperature values are detected, they are immediately reported. These data are later stored in the system database for future use.

Doing so, the system will have two kinds of container filling data. On the one hand, those that can be available when planning the routes (received from the containers with sensors) and, on the other hand, historical data of the filling level that have been obtained after collecting waste.

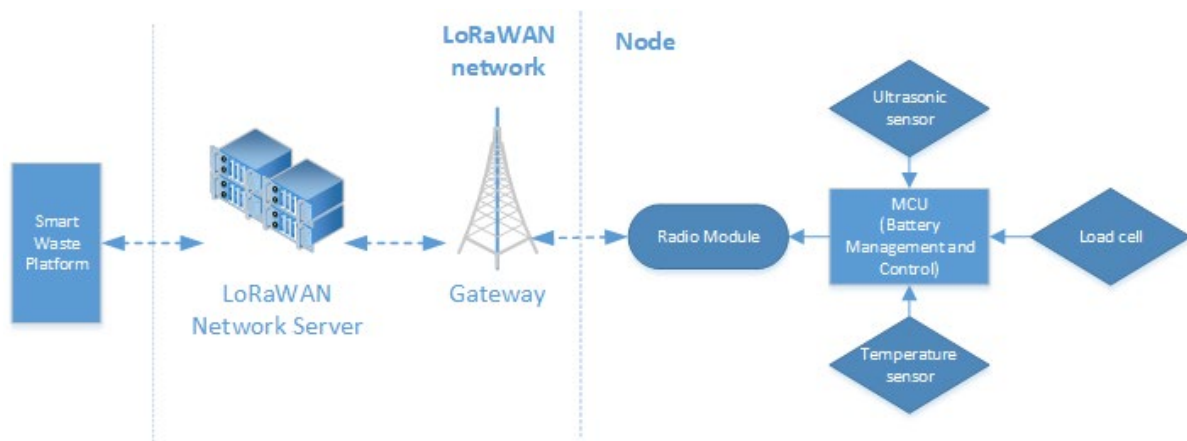
Current Smart Waste Systems implement sensors progressively so that the container fleet is equipped with sensors gradually due to the initial investment required for the deployment. This is the reason why these systems usually consist of a mixed fleet of containers. Such fleet has some containers equipped with sensors and some others without sensors, causing uncertainty in the waste amount of the conventional containers when collecting. Consequently, the proposed system includes data of the two above mentioned sources considering the uncertainty of those containers not equipped with the proposed low-consumption sensor.

Below the proposed wireless sensor network is explained in detail as well as the LoRaWAN sensor design.

### 6.3.2.1 Wireless Sensor Network and LoRaWAN sensor proposed

This wireless sensor network is based on the LoRaWAN protocol, so it comprises three basic components:

- **LoRaWAN sensor:** It is a low-consumption sensor that measures weight, volume and temperature of the container. A sensor with these features has been designed, it is explained in detail further on.
- **Gateways:** They provide coverage to the area where the sensors will be deployed, and they work as a bridge between network servers and nodes.
- **LoRaWAN network server and application server:** The Things Network [The Things Network 2016] has been selected as the platform in the design of the system because it incorporates both the LoRaWAN network server and the application to integrate the MQTT broker of the proposed system. Thereby the system remains disconnected from this technology.

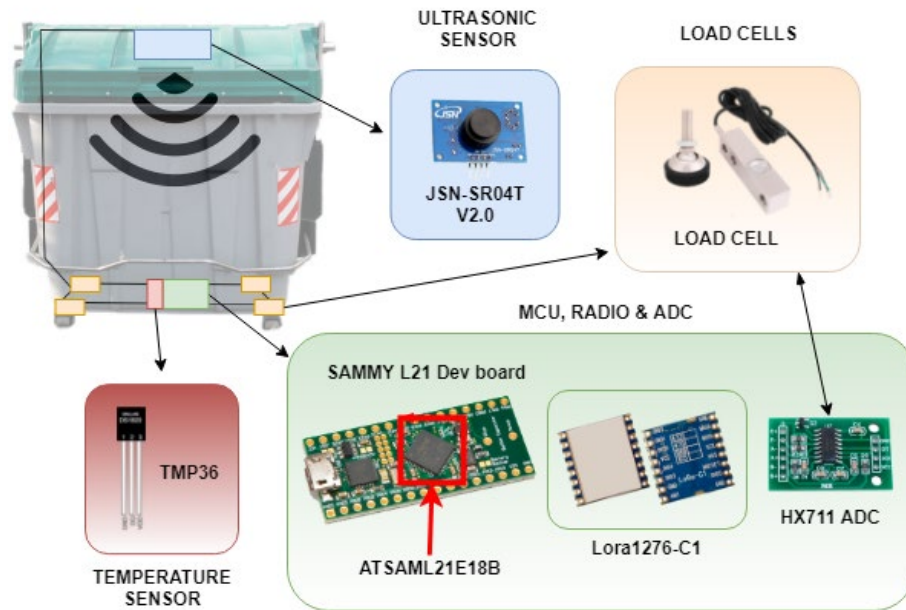


**Figure 11** General communication diagram of the proposed system

**Figure 11** shows the elements in the sensor network previously described. On the right side of the same figure, the function modules of the proposed sensor are described as well as how they interact with the rest of the system.

As the basis for the development, a device has been designed instead of using other kinds of board available in the market like RN2483 LoRa Mote of [Microchip Technology Inc] or Wasp mote of [Libelium]. This option allows to choose every component of the node on a case-by-case basis (from the microcontroller to each sensor) and, consequently, increase the power efficiency of the final device. In addition, this approach enables designing a solution that can be specifically adapted to the requirements of the system related to the data capture by using the more appropriate sensor for each kind of data.

The device has been designed to obtain data regarding three variables: estimated filling volume of the containers, the weight of the waste inside them and their internal temperature. Unlike other works currently present in the literature that only estimates the filling volume [Ramson and Moni 2017b], this work proposes the development of a node able to additionally obtain the weight for its future use in route optimization as well as the temperature for detecting dangerous situations—caused either by high temperature when the container is not empty or the potential combustion that could happen inside it. As can be observed on the right side of the **Figure 11**, the device has different functional modules. On the one hand, an ultrasound sensor, load cells, and a temperature sensor to obtain the data above mentioned and, on the other hand, modules for managing the power consumption and the communications with the network (MCU microcontroller and the LoRa radio module).



**Figure 12** Location of the functional modules of the node and the development boards employed in the prototype

The node uses the container structure and it is located in a specific location designed by the manufacturer in the bottom part. **Figure 12** shows the distribution of the modules used in the prototyping of the container: the bottom section has the board with the different modules and sensors. The load cells are located in every leg of the container and the ultrasound sensor is on the top section of it.

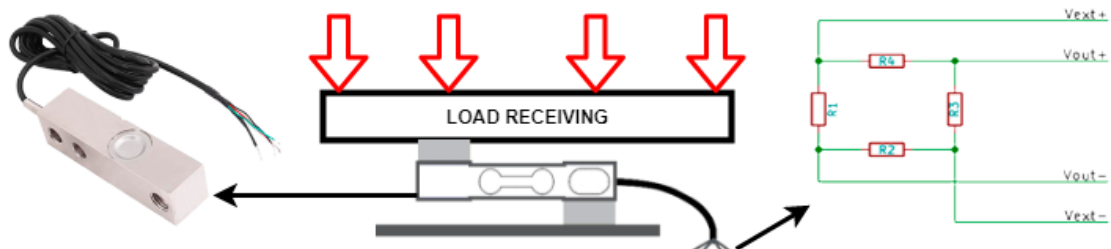
Sensors employed to develop the node are explained in detail below:

- **Ultrasound sensor:** It is the JSN-SR04T V2.0 model [Bakar et al. 2017]. It is a water-resistant sensor with a functioning range from 20cm to approximately 4m, an accuracy of 3mm and operation angle of 15°. Its functioning is based on the stimulation of a pinger and the measurement of the time elapsed between the stimulation and the return of the receiver. When the signal is received, the distance is calculated according to the following equation (7):

$$d = \frac{T * v_s}{2} \quad (7)$$

Where  $d$  is the distance between the device and the first surface achieved by the ultrasound,  $T$  indicates the time elapsed since the ultrasound was emitted until its echo is received and  $v_s$  is the sound speed. This sensor calculates the estimated volume of the container. **Figure 12** shows an image of this module.

- **Load cells:** They are employed for measuring the weight of the container. For this work, four load cells have been used and they have been attached to each leg of the container. This kind of sensors is based on a Wheatstone bridge operating principle. **Figure 13** presents a diagram of the load cells functioning.



**Figure 13** Diagram of a load cell and the associated electrical circuit

When there is a weight variation in the transducer, resistance is modified due to its intrinsic features. Since there is a continuous difference in voltage ( $V_{ext+} - V_{ext-}$ ), the produced voltage difference can be calculated through the equation (8):

$$V_{out} = V_{ext} * \left( \frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) \quad (8)$$

Since changes in these measurements are really small, an analog-digital amplifier (24-bit resolution HX711 model) is required to obtain enough accuracy and range. This module is shown in **Figure 12** together with the microcontroller and the radio module. The amplifier receives the signal from the load cells and converts the small variations into digital measurements that are sent to the microcontroller. The container weight can be obtained by calibrating the obtained voltage difference with a known weight value.

- **Temperature sensor:** an analog sensor model TMP36 was used for measuring temperature. This sensor provides results with an accuracy of one degree Celsius. In this case, the temperature accuracy is not a crucial factor in the development of the node, but its communication, analogical, is vital for the subsequent energy efficiency of the node. **Figure 12** shows the selected development sensor.
- **Microcontroller (MCU):** One of the main aspects of the node design proposed in this work is the selected microcontroller. It is an Atmel SAML21 model. It is a microcontroller with a 32-bit RISC processor belonging to the ARM Cortex-M-0+ ultra-low consumption processor family. This microcontroller consumes 35  $\mu\text{A}/\text{MHz}$  in active mode while it consumes low than 0.5  $\mu\text{A}$  in standby mode when working in a voltage of 3.3V. It can perform several tasks at the same time. In this case, it allows to obtain the values of the different sensors in parallel as well as store them for its future transmission. Choosing the appropriate microcontroller implies the study of rankings like ULPMark [ULPMark]. The selected one for this design is the only one available in

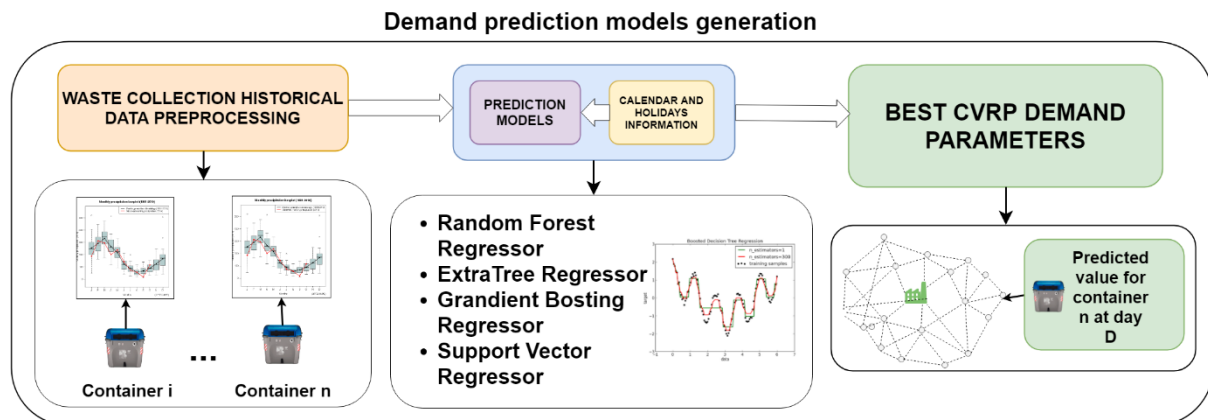
such ranking having an ARM Cortex-0+ architecture. S Sammy-L21[Sammy-L21] board has been initially used as developing board, as can be observed in **Figure 12**. Another key aspect of this microcontroller is its ability to perform what is known as Customized Logic Operations CCL through a unit aimed at this. This unit is really useful as it allows to perform operations with the temperature measures by employing previous measurements. Thereby, the sudden temperature increasing as well as exceeding a given threshold can be assessed to send an alert to the system. This unit is created using logic doors so that it does not depend on a synchronous programme which is executing in the microcontroller. Consequently, this unit can work even if the microcontroller is in sleep mode so that it can be activated if required. This improves the power efficiency of the whole node.

- **Radio module:** Since it is a node in a LoRaWAN network, a LoRa radio module is required to transmit the obtained information via the sensors mentioned above. The module selected for this purpose is the LoRa1276-C1 model, developed by niceRF [Nice RF]. It provides great sound isolation, and it presents a low consumption of 0.2  $\mu\text{A}$  when in sleep mode.

The entire sensor receives electrical power from a reduced battery. The capacity of this battery will vary depending on the requirements of the case study. This work later presents the case study in which the sensor functioning is analysed together with a Gateways network deployed to provide coverage to a waste collection system in the region of Salamanca.

### 6.3.3 Demand prediction subsystem

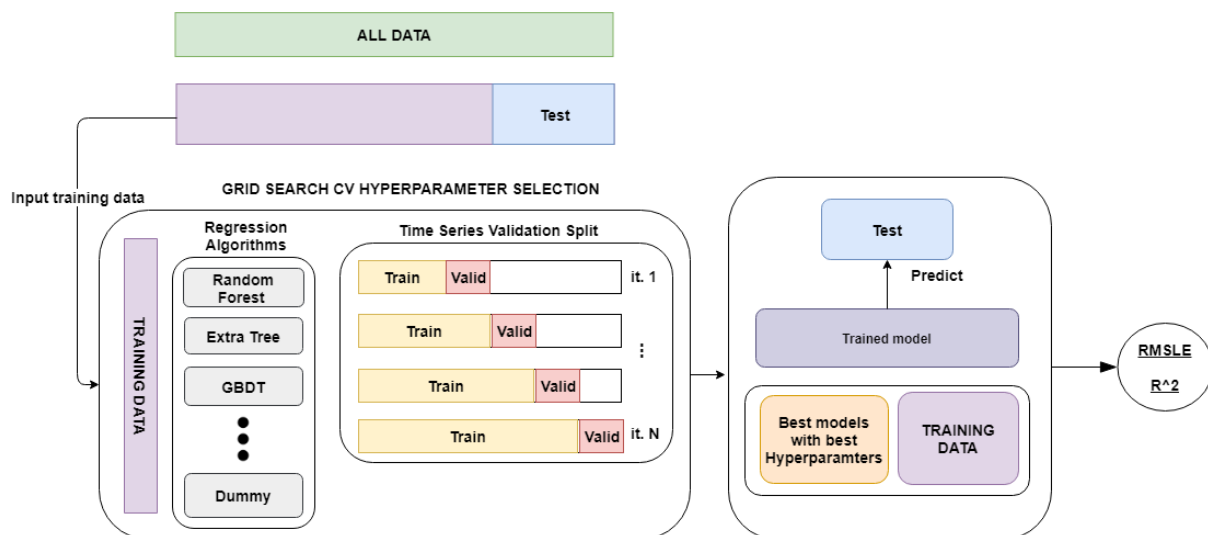
The demand prediction subsystem is the subsystem employed to estimate the demand of each container by using collection historical data previously obtained. It gets the collection of historical data of each container, processes them and trains predictive models for each one of them. Its objective is to employ them to estimate the demand for the containers that have no sensors and consequently neither data.



**Figure 14** Prediction of the demand from collection historical data

**Figure 14** Shows a general diagram of the demand prediction process. Firstly, historical demand data are extracted by performing a pre-processing of the data to eliminate any possible outlier. Next, these data are combined with other collection historical data and the working calendar: day of the month, day of the week, week of the month, month, bank holidays, a bank holiday in previous days and number of days since the last collection.

**Figure 15** shows the methodology employed to create the models, select the hyperparameters and the final models used to predict.



**Figure 15** Diagram of the methodology employed to train the models, and select the hyperparameters and the predictive models

Once the data have been pre-processed, a first separation is made between training data and test data. These training data are then used to select hyperparameters of different algorithms such as Random Forest Regressor, Extra Trees Regressor, Gradient Boosting Regressor, Support Vector Regressor. In addition, a Dummy algorithm is used as the basis to see the functioning of the rest of the algorithms (in this case its prediction is the average demand of the previous month).

These models are generated using a Grid Search scheme for hyperparameters selection. This method allows to select a grid of hyperparameters and it generates as many models as possible combinations between the different hyperparameters were specified in the grid. At the same time, a Cross Validation (CV) process called Time Series Split Cross Validation is performed. This cross-validation method is used for time series, in such a way that the training and validation sets are split progressively since it is not possible to train models employing future data to predict past instances, as it would occur if a conventional k-fold cross-validation scheme is applied. Finally, the error obtained after generating predictions with the initially separate test set (only with posterior data to the selected model) is studied.

#### 6.3.4 Optimization collection routes under uncertainty subsystem

This subsystem is responsible for optimizing the waste collection routes, considering the current state of the Smart Waste Collection System and uncertainty associated with each of the collection containers in the fleet. A VRPSD is proposed, in which, on the one hand, there are containers equipped with sensors of which the uncertainty is minimal during collection and, on the other hand, containers of which only historical data and predictive models are available to estimate the amount of waste for the day of collection.

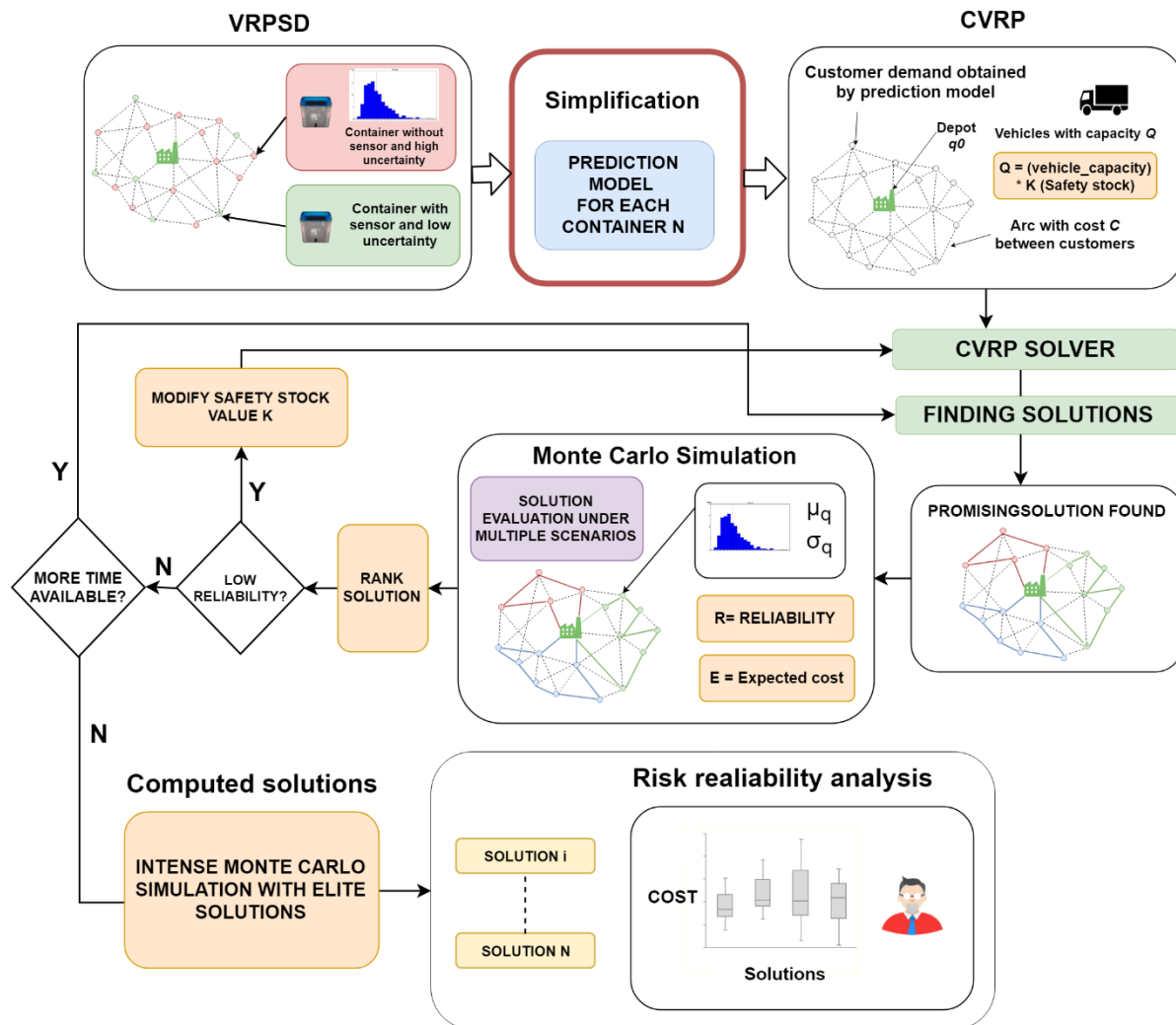
The methodology proposed by Juan et al. [Juan et al. 2011] is used as the basis for the search for optimal collection solutions. In this work, a modification in the methodology is presented in order to improve the performance of this in the optimization of routes.

**Figure 16** details the calculation process followed by this subsystem. In the first place, data relating to the containers from which the collection is to be carried out are obtained; these may include containers with or without sensors (**Figure 16**, top left). Following the Simheuristics methodology, the stochastic problem is simplified to a deterministic CVRP problem (**Figure 16** top central). In this proposal, the predictive models that have obtained the best results during the training are used to obtain the estimated value of the demand of each container for the day of collection, in contrast to the original proposal by Juan et al. in which they use the expected value of the distribution chosen for each container.

This step can be fundamental since, if the models generated fit adequately to the real demand, the deterministic solution would be much closer to the real one and therefore the values



obtained in the simulations will present higher feasibility values. For example, in containers that present a very high demand at certain times of the year, the simplification employing predictive models will be closer to the real one than via the expected value or the average of a distribution obtained for this container.



**Figure 16** Simheuristics methodology diagram including predictive models

Below (Figure 16 top right), a classic CVRP problem to which a safety stock  $K$  variable is added [Juan et al. 2011] [Gruler et al.] is presented. The  $K$  variable is used to reserve space in each truck for possible excess demand in subsequent simulations. The load  $Q$  considered for the truck is displayed in equation (9):

$$Q = Q' * K \tag{9}$$

Being  $Q'$  the real capacity of the truck and  $K$  the percentage of the vehicle to be used for carrying waste, thus, the smaller  $K$  value is, the greater the space reserved and the feasibility of the routes are.

Once the parameters of the CVRP problem have been established, a search for solutions is performed using the most efficient method from state of the art for a CVRP problem (as previously observed, heuristic and metaheuristic methods usually give the best results when the problem is large). As the algorithm implemented in the so-called solver finds promising solutions, these are evaluated by a Monte Carlo simulation which generates multiple scenarios in order to obtain values about the associated costs of each solution and a feasibility index (central part of **Figure 16**).

During the simulation, the space reserved in trucks for excess demand is taken into account, but, even so, there may be scenarios in which the solution fails, and a recourse action must be performed: a return to the depot from the container where the failure occurred and continuation of the route from the last container visited. If any of the routes in the solution fails, the complete solution is considered as not feasible for calculating the feasibility index. At the end of the simulation, a feasibility index is obtained, expressed as a percentage and an array of costs for each of the solutions. A ranking of solutions is elaborated with these data (central left part of **Figure 16**). Depending on the  $K$  selected for the approach of the problem, the levels of feasibility will be lower or higher so, if they do not meet a criterion introduced beforehand,  $K$  may be modified to obtain higher values of feasibility by reserving more space. It is worth mentioning that by reserving more space, costs will increase and sometimes the use of more vehicles will be required to meet the entire demand.

If there is no time available to continue looking for solutions or a stop criterion has been reached, an intensive Monte Carlo simulation is performed with the solutions best positioned in the ranking according to their feasibility and cost. Finally, these solutions are presented with the new feasibility indexes and cost values to the decision maker to evaluate the solution to be implemented (bottom right of **Figure 16**).

## 6.4 Case studies

Throughout this chapter, the case studies performed during the development of the thesis work are described. These are three case studies that focus on each of the parts of the proposed system.

The first case study addresses data collection and the development of a low-power sensor together with a LoRaWAN WSN. It presents a study of coverage of the WSN proposed and explains how to employ the obtained data for route optimization.

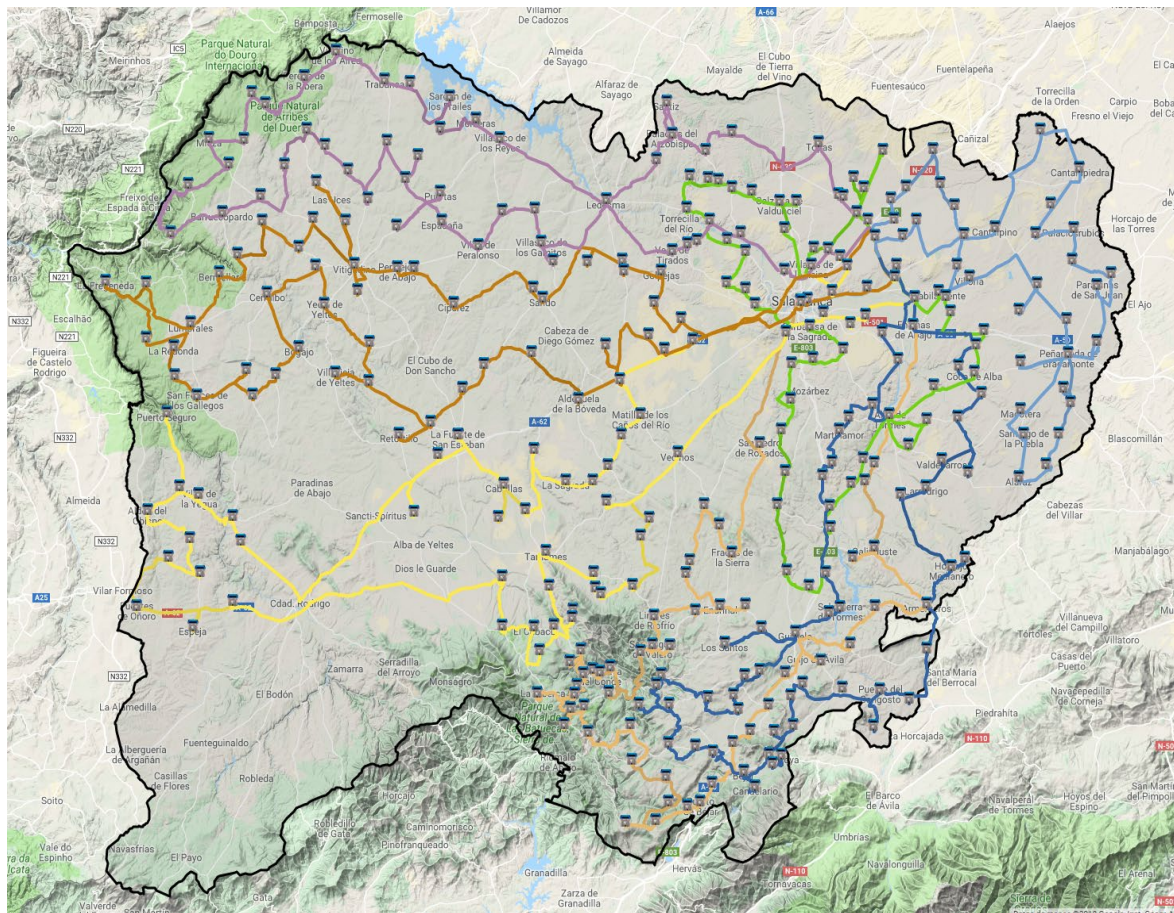
The second case study approaches the study of demand prediction, but, this time, in a different vehicle system: A Bike Sharing System. As predicting the amount of waste from each container is required in Smart Waste Collection Systems, Bike Sharing Systems need to predict the demand that will exist in each station.

The third and last case study focuses on the inclusion of uncertainty in route optimization methods using the data obtained and the predictive models studied in previous case studies. The objective is to incorporate improvements in an optimization methodology for Vehicle Routing Problems in scenarios where the demand of the clients to be visited presents an uncertainty.

### 6.4.1 Data collection by using LoRaWAN WSN

The study is based on a public tender procedure from the year 2016 about the selective collection of waste (paper, packaging, etc) [Diputación de Salamanca]. According to the technical document, the regional administration of Salamanca requires the recollection of selective waste from 329 towns in the region and from their own installations. In addition, they include relevant information about the number of containers in each town and their yearly productions of selective types of waste. In the same document, they specify a minimum fleet of 8 vehicles for the collection.

The main objective of this case study is to analyse the proposed WSN in this work for performing data collection in the context of a real Smart Waste Collection System: (1) performing a study of the power efficiency of the proposed sensor used to obtain the bin related data, (2) designing a deployment of gateways in order to provide the needed coverage to towns involved in the case study, (3) optimizing collection routes taking into account a bin fleet equipped with the sensor developed and assuming a dynamic approach for calculating the collection routes instead of a traditional static approach.



**Figure 17** Towns of the case study and routes performed by the permanent vehicle fleet.

This information was used to formulate the CVRP problems. In the case study, the statements were set as follows: these vehicles collect waste every two weeks, and they always perform the permanent routes presented in **Figure 17**. This results in a cost of around 3050.725 km in journeys every two weeks.

This case study was performed using synthetic data generated according to statistics of selective waste production in towns. This information was derived from the previous documents and the waste plan of the regional administration of Castilla y León (Junta de Castilla y León) [Gobierno and Rodríguez 2001b]. Data related to the region in OpenStreetMaps was loaded in the geographical information subsystem.

The following section presents a discussion about the results related to the developed sensor, the WSN presented for this case study and the process of route optimization applied to this case study.

### 6.4.1.1 Measurement results from the developed sensor

A decisive factor of the proposed system is the nodes' battery lives. One of the main purposes for this was to include the system without increasing the current maintenance requirements of containers, which must be revised between 1 and 4 times/year. This depends on the kind of waste and implies that the device battery must work for at least a year. The amount of battery life consumed by the node was evaluated by adding the power consumed by every built-in component and subsequently, the total consumption of the components together.

Before creating the sensor, theoretical estimations of consumption were performed according to the datasheet of every component and the tools of their manufacture so that power consumption for each action could be calculated: sensor reading, information processing or sending through the network.

Different batteries in the market express the value using two quantities: power and electrical load, measured in Ah. One ampere/hour (Ah) is the electrical load unit that represents a current of one ampere passing through a conductive element for an hour. This allows the power of batteries to be described and consequently, the load that the device will consume for working can be calculated—expressed in mAh by Equation (10):

$$Q = \sum_i i_i \cdot t_i = t_{MCUi} \cdot i_{MCUi} + t_{radio} \cdot i_{radio} + t_{MCUa} \cdot i_{MCUa} + \sum_j t_{sensorj} \cdot i_{sensorj} \quad (10)$$

$Q$  is the charge consumed in (mAh),  $i_i$  is the intensity consumed by the microcontroller in idle mode and  $t_i$  is the time.  $MCUa$  is the intensity of the microcontroller in active mode ( $MCUi$  inactive mode), and  $i_{radio}$  is the intensity consumed by the radio module to perform the shipment. The summation presents consumptions of different  $j$  sensors during the capture of different data.

The ultrasonic sensor generated consumption of 30 mA when working. Due to its properties, the maximum operating time if the container is empty will be 40 ms. The implemented weight system consisted of four load cells connected to a hub and an ADC amplifier which was accurate enough to capture the variation produced by an increase in weight. This amplifier provides the required voltage for the proper functioning of transducers. It only consumes 1.5 mA while performing a measurement for 40 ms. The temperature sensor employed for this case study (TMP36) was an analog temperature sensor. It consumes 4 mA and can perform measurements every 5 ms. The LoRa1276-C1 module—developed by niceRF—manages communications with the Gateway. This device uses as a maximum of 140 mA upon shipment and sends data in less than 10 ms. These consumption numbers were added to the period of

time that the microcontroller was active and fully working in order to obtain the total required power to read the sensors and perform the shipment.

The microcontroller uses 35  $\mu\text{A}/\text{MHz}$  when working. As it works at a frequency of 48MHz, it consumes 1.68 mA. In active mode, it performs two tasks: capturing data and managing shipments. For capturing data, the working time employed in the equation is the longest time spent obtaining measurements by sensors. In this case, the weight sensor takes the longest time to provide data—it provides a measurement in 40 ms. When the MCU is in idle mode, it consumes 200 nA, so it uses a charge of 0.2  $\mu\text{Ah}$  to be in idle mode for 1 hour.

Once the estimations were obtained, we proceeded to build the system and we obtained the battery consumptions. The tests consisted of simulations to define the real consumption rates of the node under different data capture and shipment conditions. This allowed specifying the best rate to transfer data to the CVRP system and maintain a consumption level to keep the battery working for more than a year.

**Table 3** Measurements, shipments and consumption rates

Shipments/day	Measurements/day	Estimated Consumption (mAh per day)	Measured Consumption (mAh per day)
12	12	6.18	6.84
12	24	7.5	7.8
24	24	25.488	26.88
24	48	28.128	32.16
48	48	104.3232	107.04

**Table 3** describes some possible configurations of the nodes in order to estimate the consumption of the device. The first column describes the data shipment per day; these shipments are periodically performed at the same time intervals and provide the needed information for node selection. The second column shows the number of measurements taken in a day. The third column displays the battery consumption estimated with the previously explained method. The fourth column shows the actual measured consumption of the device in the lab.

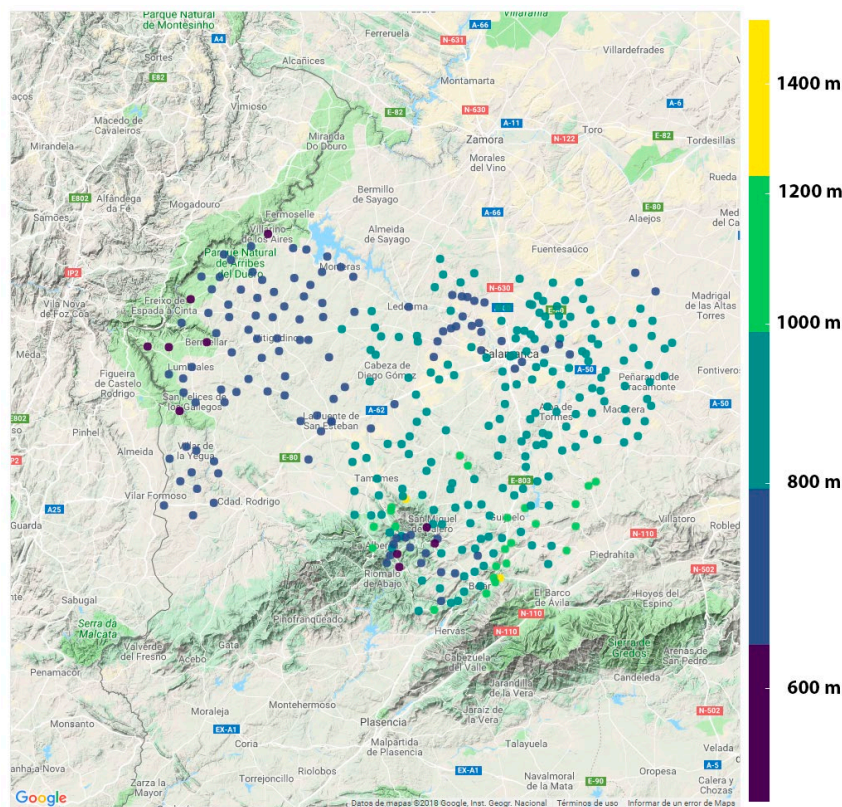
It is necessary to point out that depending on the Spread Factor (SF) and payload length selected for LoRa radio configuration, the total consumption could be affected notably. There are some studies in the literature such as [Cheong et al. 2017] and [Bouguera et al. 2018] which address the comparison of power consumption on different SF and payloads. The result obtained by these studies are very close to the results in this work, their results describe a scenario of SF 11



and a payload of 25 bytes with sends every hour with a total consumption of 114000 mAh every 10 years, that is 31,78 mAh per day. **Table 3** highlights how the shipments are the main factor in the total consumption of the node and how LPWAN technologies such as LoRa are crucial in the development of high power efficiency nodes.

### 6.4.1.2 Network coverage study

This section presents an initial study of the LoRaWAN coverage in the region of Salamanca. The terrain was extracted to identify the best locations for placing the *gateways*. The best options were towns with the highest altitudes in the area with easy-to-access internet to reduce costs. **Figure 18** shows the height in meters of each town intended to receive coverage in the proposed case study.

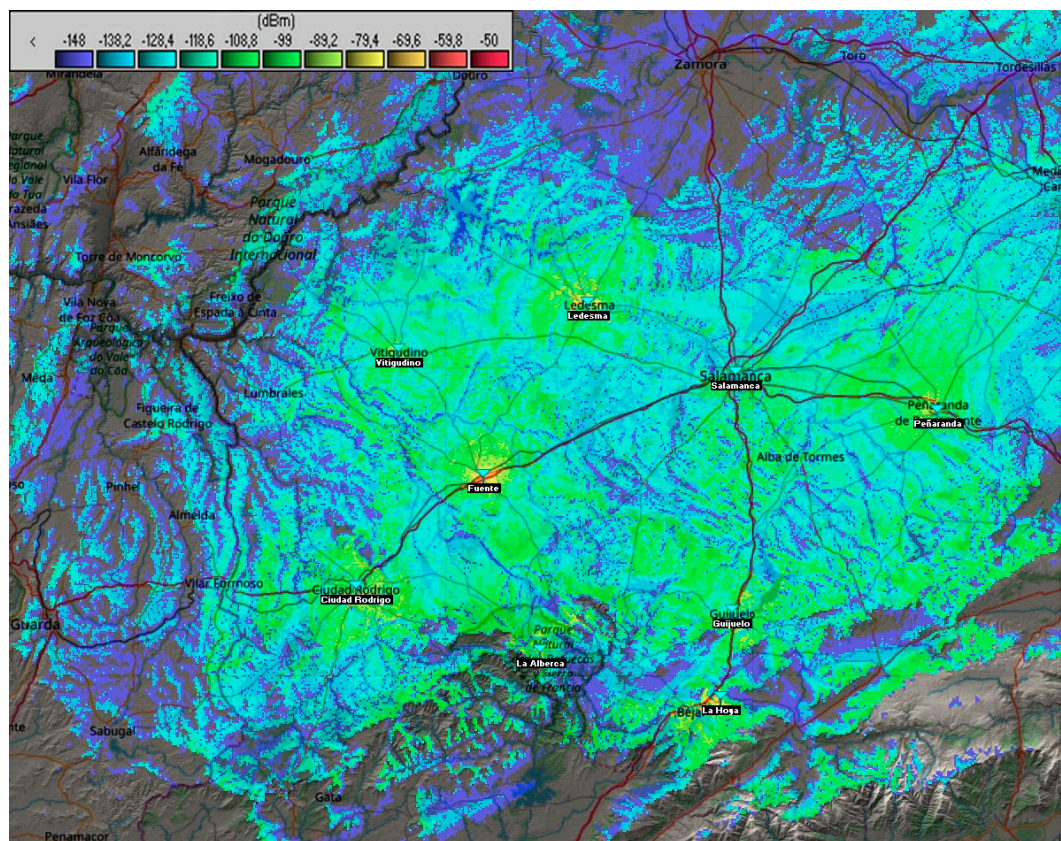


**Figure 18** Towns in the case study and their corresponding altitudes in meters

Radio Mobile [Cu 2014] [Hopengarten 2009] —which is widely used to study radio frequency coverage— has been employed for studying the coverage. This tool was selected over others like CloudRF [CloudRF] because it supports (just like CloudRF) the Longley–Rice propagation model (also known as Irregular Terrain Model ITM) [Kasampalis et al. 2013] and it is free to use. Based on previous data, a decision was taken to deploy a network in the region consisting of nine Gateways, located, as shown in **Figure 19**, in Vitigudino, Ciudad Rodrigo

and Fuente de San Esteban (West), La Alberca, La Hoya and Guijuelo (South), Peñaranda (East) Ledesma (North) and Salamanca.

**Figure 19** also shows the network coverage in dBm on the map. Using the terrain and the properties of LoRa technology, coverage can be provided to all the towns of the region as well as surrounding ones with no more than nine antennas in the area. This makes the deployment cost really low because several towns and their demarcations are covered by one Gateway, reducing the total cost of the deployment.



**Figure 19** Coverage map of the study performed with the application, Radio Mobile

The coverage map in **Figure 19** is a view of merged Cartesian coverage where the transmission is specified by the node.

Variables employed in Radio Mobile to perform the coverage study are, on the one hand, transmitter power of 0.5 W, receiver threshold of 0.0316  $\mu$ V, line loss of 2.5, omnidirectional antenna, antenna gain of 13 dBi and a height of 20 m (**Figure 20**) for the base station.

On the other hand, transmitter power of 0.025 W, receiver threshold of 0.0316  $\mu$ V, line loss of 0.1, a dipole antenna, the antenna gain of 2 dBi and a height of 1 m (**Figure 21**) for the nodes.



These parameters were set according to the recommendations in ERC-REC-70-3E [Committee 2015], European regulations regarding the radio frequency band of 868MHz, which is a public bandwidth for European Low Power Networks (LPWAN).

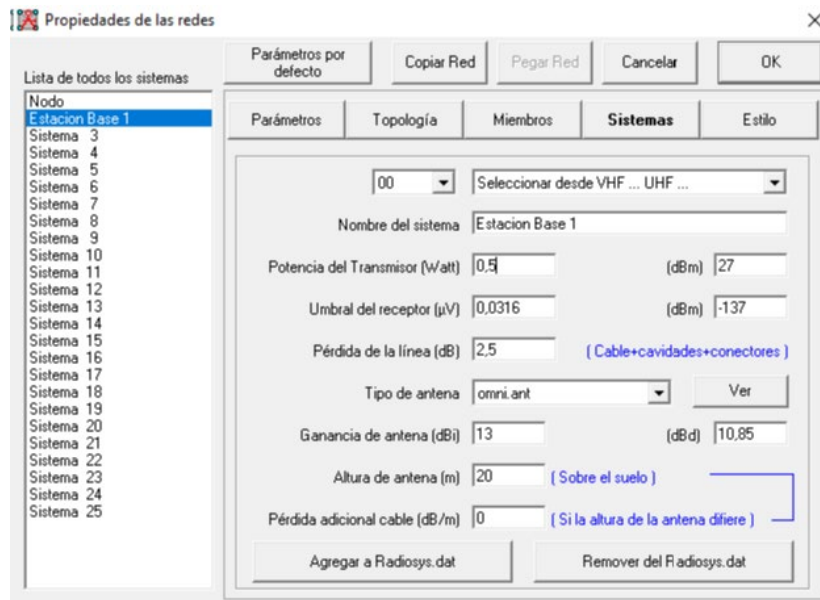


Figure 20 Configuration parameters for base stations in the tool Radio Mobile

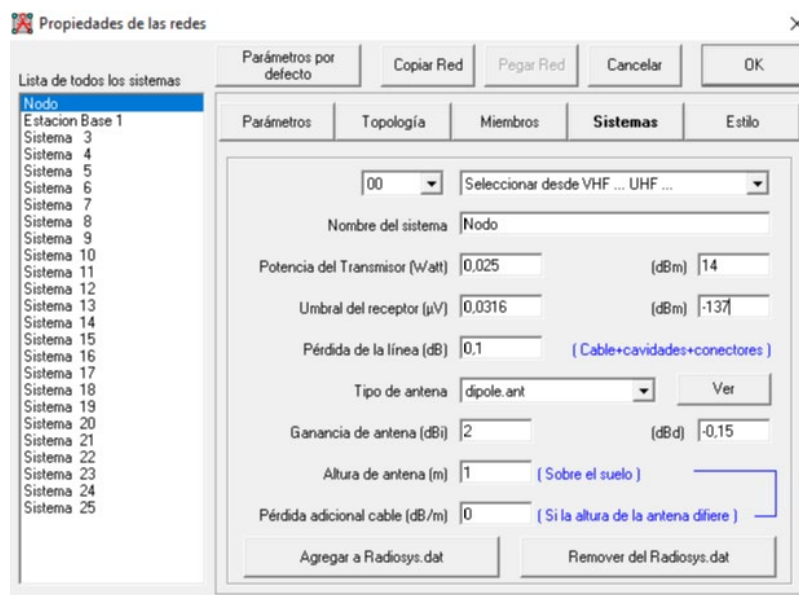


Figure 21 Configuration parameters for nodes in the tool Radio Mobile

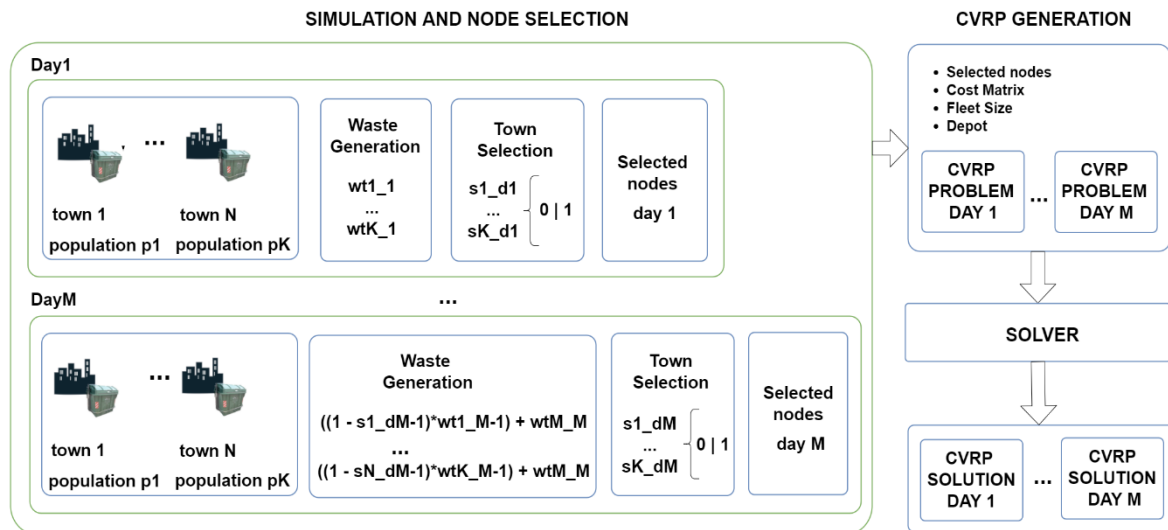
After the theoretical coverage study, several tests were performed in areas with lower coverage to test if the model employed in this case study was correctly adapted to real conditions. This resulted in the possibility of deploying a network in cases where rural areas use this technology,

such as in smart farming domain works. Having this kind of infrastructure promotes the development of applications in this domain, even more, if the deployed network includes LoRaWAN and The Things Network and allows access to the network for the whole region.

### 6.4.1.3 Route optimization results

This section describes a simulation of the route optimization process using the data of waste production from the current waste plan of the regional administration of Castilla y León [Gobierno and Rodríguez 2001a], the technical document of the public tender procedure of 2016 [87] and a report about selective collection of paper [Ecoembes 2008] which will be the waste collected in this case study. Waste collection scenarios were generated for one year with a total of 26 waste collection days. These scenarios were generated taking into account the population of each town and the mean rate of waste production per inhabitant and day for each town.

In the simulation, a visit to a town was assumed to imply the collection of every container in that town; in this case study, the displacement produced by the truck in each town did not significantly penalize the total amount of kilometres of the route.



**Figure 22** Simulation of waste, node selection, CVRP instances generation and solutions

**Figure 22** shows the simulation and selection of the towns on the left side. Firstly, the populations of all towns were obtained. Secondly, the waste generation for each town was performed to obtain a random value of waste that follows a normal distribution [Peebles 1987] considering the inhabitant waste generation ratio which has previously been mentioned:

$$[W_{t11}, \dots, W_{tK1}] = \text{randomNormalValue}(\mu = \text{townwaste}_t, \sigma = \text{townwaste}_t/10) \quad (11)$$

where  $[W_{t11}, \dots, W_{tK1}]$  is the waste generation of  $K$  towns for the first day of collection, and  $\text{townwaste}_t = \text{inhabitantWasteRatio} * \text{daysBetweenCollection} * \text{population}_t$ .

Thirdly, a selection node process was performed in order to include the towns considered for calculating the next waste collection routes. In this simulation, the variable employed was the weight of the waste produced by the whole town. However, in a real production environment, the node selection would involve measurements of volume, and then, the weight of each container would be used for the definition of the CVRP problem. In this work, we only had data about the weight, so an estimation of volume using a paper density of 46,47 kg/m<sup>3</sup> [Ecoembes 2008] was employed in the calculation. The selection criteria for each node was as follows:

$$CurrentPopulationWaste + PopulationWasteUntilNextDay \geq 0.80 * PopulationWasteCapacity \tag{12}$$

where, in Equation (6), *CurrentPopulationWaste* is the total demand up to this day and *PopulationWasteUntilNextDay* is waste generation for the following days until the next collection day. If this is higher than the 80% of a population’s waste capacity, that population will be selected for collection. This percentage was extracted from the technical public tender document [Diputación de Salamanca]. The *PopulationWasteCapacity* was calculated from the total population of the town and the number of containers available for that town.

The previous process was performed for each collection day. In the waste generation step the previous selection of nodes was considered in order to add it to the pending demand of every node which has not previously been visited.

As a result of this simulation process, the selected town nodes and their waste demands for the M collection days were obtained. The next step was generating the CVRP instances employing the selected nodes, the available vehicle fleet and the distance matrix of the selected town nodes.

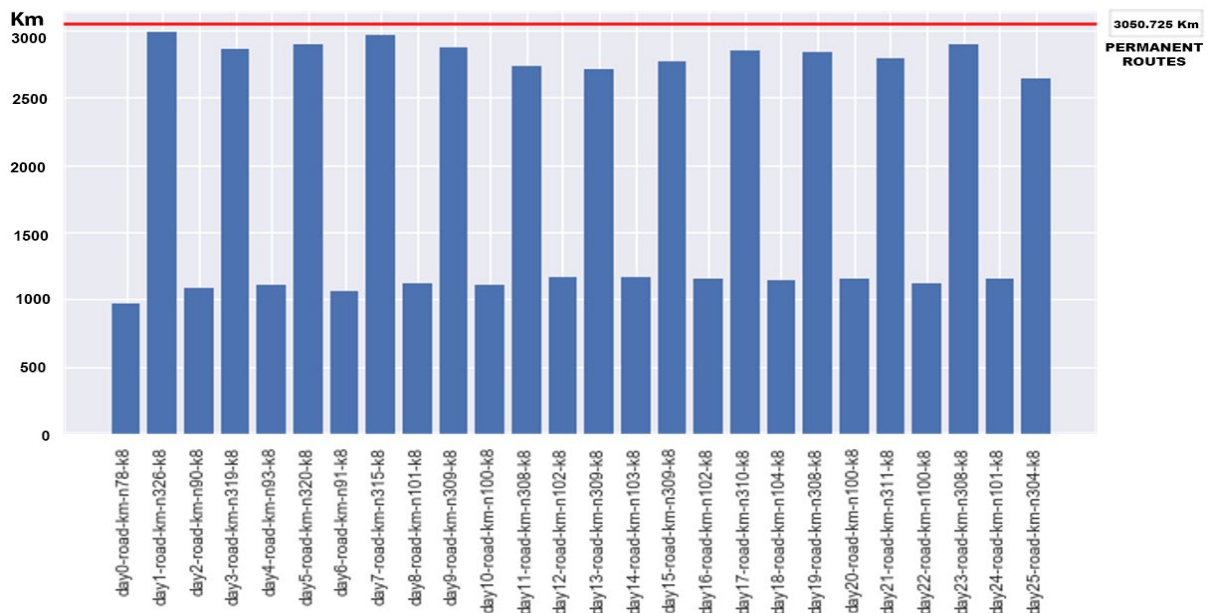
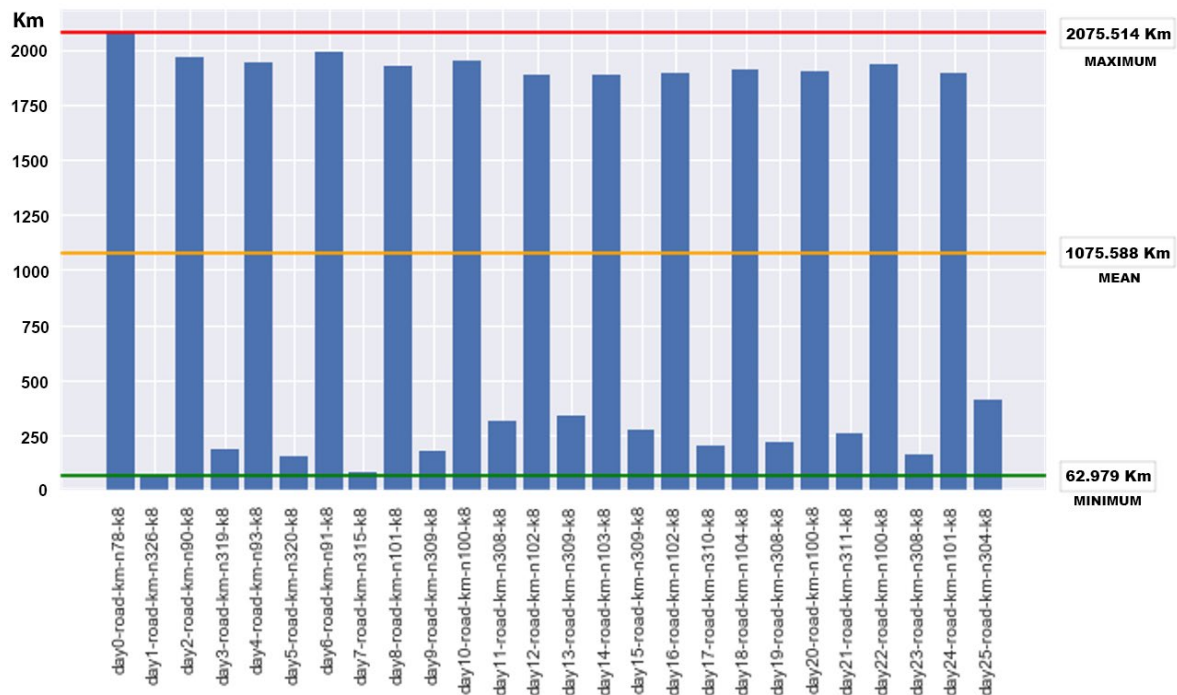


Figure 23 Results of each day of collection and the amount of km of fixed routes

The results of the 26 collection days over a year are displayed in **Figure 23**. The distance in km is shown for each CVRP instance against the consumption of the permanent route approach.

As can be observed, the permanent routes have a cost of 3050.725 km (horizontal red line **Figure 23**) for each collection day whereas with the route optimization, as expected, there are savings in the total cost of distance for all cases



**Figure 24** Savings per each collection day

**Figure 24** shows the savings produced in each instance, with a maximum saving of 2075.514 km (top red line in the chart) and a minimum saving of 62.979 km (green bottom line) and a mean saving of 1075.588 km (medium orange line) for each instance. This means a mean saving of 28%. It should be noted that the pattern in the results is caused, on the one hand, by the nodes which are not selected in previous collections and, on the other hand, by the minimum number of collections which are specified in the public tender as a requirement.

#### 6.4.1.4 Conclusions

The case study described in this work focused on waste collection in the region of Salamanca shows the improvement provided by the proposed system compared to a classic static collection approach. This system offers a better operation in terms of covered distance, fuel consumption, truck use and emissions, operating costs savings and collection of useful data for further waste generation analysis.

Concerning the developed sensor, it features a low consumption rate and sufficient battery life to work throughout the presented case. The sensor correctly obtains data and sends them without penalizing the battery life due to the employed radio technology.

Deploying this sensor allows the monitoring of implemented containers as well as the procurement of information about the waste production of inhabitants in each town. This helps the waste collection company detect whether more containers need to be deployed when the container filling is too fast at specific times of the year, for example, when the population temporally grows during the summer.

Regarding the coverage study, nine antennas in the Salamanca region met the coverage requirements of the presented case study. This meant a reduced investment considering the covered extent of the land and the needed infrastructure in the region.

Concerning the route optimization results, the simulation offered mean savings in distance of 28% in the analysed case. Due to these savings, the time and use of trucks are also decreased, and consequently, the workforce costs decreased.

Finally, the simulation process of collecting paper and board waste along with obtained data showed that collecting waste every 2 weeks did not suit the waste production of every town in the community. The collection capacity was sometimes insufficient and in other cases the opposite. Selecting the indicated collection frequency is fundamental for saving time when performing this task. This is the reason why a system that dynamically generates routes with updated information provides a decision-making tool to perform collections with the proper frequency (when they are really needed).

In conclusion, as indicated by the case study, the proposed system includes a developed node which can operate efficiently for more than a year, obtaining useful data like weight, temperature and volume due to the low battery consumption. In addition, the coverage study showed that with reduced investment, deployment in the region with a small number of antennas covering the whole province of Salamanca is feasible. Together with these results, the optimization system results provide savings in cost, time and workforce against a static collection route approach.

The potential savings of totally deploying the waste collection system and the advantages of this kind of WSN could allow investment in the required infrastructure as well as enabling future IoT projects in the region, improving its technological development.

To conclude, after performing this case study, the proposed data collection system includes a node that can work efficiently for more than 1 year getting useful information like weight, volume and temperature of the containers due to its low consumption. Furthermore, the performed coverage study confirms that reduced investment in gateways infrastructure allows providing coverage to the Salamanca region. Along with these results, route optimization by selecting nodes dynamically and calculating the best route for each setting provides savings in the journey, time and workforce costs against a static route approach.

The potential savings of the Smart Waste Collection System and the advantages of this kind of WSN could allow to invest in the required infrastructure as well as incorporate the sensors to the containers fleet gradually. Furthermore, the investment in the infrastructure would be reusable in other IoT projects in the region promoting the potential technological development of this area.

## 6.4.2 Demand prediction in Bike Sharing Systems

The main goal of this case study is to examine in depth the part of the proposed system related to demand prediction. This work focuses on how to predict the demand in a logistic problem which has common features with waste collection. In this case, it addresses how to calculate the optimum reallocation routes in a Bike Sharing System. For this purpose, firstly, it is which demand will be at each station must be known in order to do further optimizations. This work tackles that problem and focuses on the part of the proposed system related to demand prediction.

In the BSS, users pick bicycles from stations spread around the city to do a trip from that station to another one. The main trouble in this kind of systems is the number of available bicycles (for picking up by users) and free docks (for dropping bicycles by users) at each station.

Thus, a user may not pick up a bicycle due to a shortage or, on the contrary, a user may not drop a bicycle because there is no free dock. The operators in the BSS, therefore, need to balance the stations by changing the bicycles from one station to another during the day.

In order to perform this task, the previous prediction of the future state of the system is required. This is done by employing historical data to detect which station will have more demand during the day. Once the demand is predicted, a Static Bike Relocation Problem SBRP [Chemla et al. 2013] [Erdoğan et al. 2014] arises and it could be formulated as a Pickup and Delivery Routing Problem (PDVRPs) included in the family of VRP problems [Dell'Amico et al. 2014]. In this case study, just the part regarding demand prediction is addressed.

This thesis work is related to the project MOVIURBAN [MOVIURBAN], whose objective is to create a system to manage the urban fleets like those included in the BSS. Thanks to this project the data from BSS SalenBici in the city of Salamanca were provided for the research.

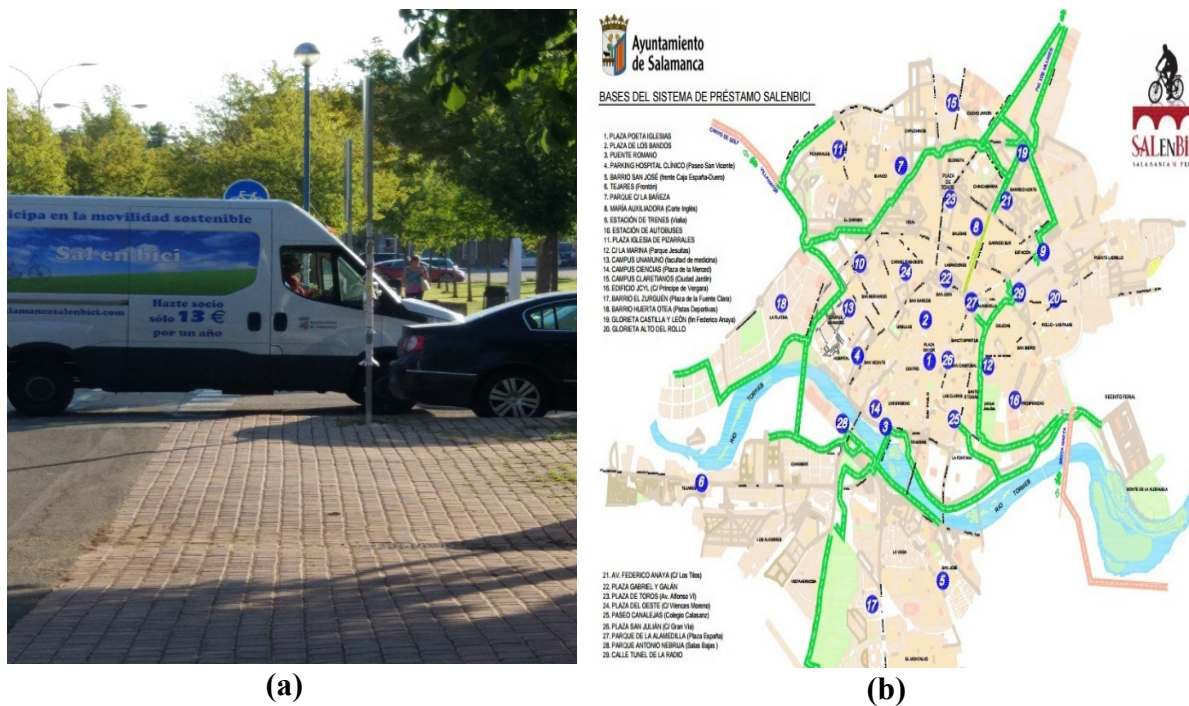
These data were employed to perform this case study whose main goal is to create a system capable of capturing data from different sources and merging them within data from BSS in order to build predictive models for each station in the BSS.

By performing this case study, the demand prediction part of the system proposed has been studied and applied to a real environment of the urban mobility system.

### 6.4.2.1 Bike Sharing System of Salamanca: SalenBici

The SalenBici System (**Figure 25**) is a bike share system located in Salamanca, a medium city with a population of 144.949 according to the last census of 2016 [11]. This system has nowadays 29 stations and 176 bikes throughout the city. The system working hours are labour days from 7:00-22:00 and weekends and bank holidays from 10:00 to 22:00.





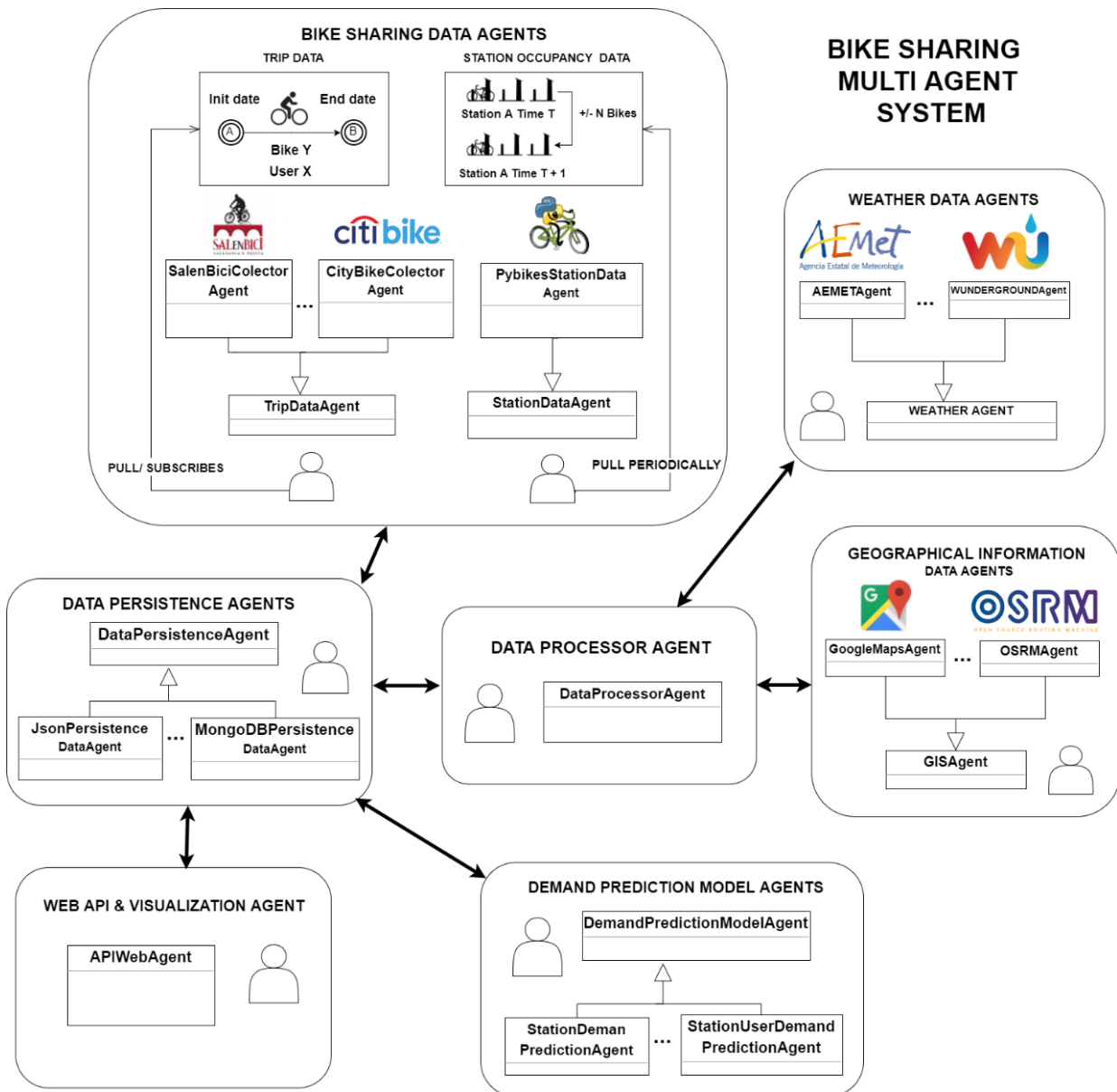
**Figure 25** (a) One of the trucks of the SalenBici system operators; (b) Map of the station provided by the council of Salamanca

The system operators perform reallocation tasks in the evening, at noon and in the afternoon, thus extra information on the expected demand in the mornings and evenings could be useful for performing the reallocation tasks.

#### 6.4.2.2 MAS for demand prediction and route visualization in BSS

A multi-agent system for data collection, demand prediction and route visualization has been designed. Several multi-agent platforms such as SPADE [Gregori et al. 2006], JADE [Bellifemine et al.], PANGAEA [Zato et al. 2012], AIOMAS [Aiomas] and osBrain [OsBrain] were evaluated and osBrain was finally selected for this system because of its ease of use. Furthermore, it is implemented in Python (like SPADE and AIOMAS) and it is on continuous development nowadays.

In this MAS each agent has a specific task to perform and they could communicate between them if the accomplishment of the tasks required it. The system is divided into 4 groups of agents: Bike Sharing Data Agents, Weather Data Agents, Geographic Data Agents, Data Processor Agents, Demand prediction Agents and Web API Agents showed in the **Figure 26**



**Figure 26** General diagram of the proposed multi-agent system for bike sharing systems

- **Bike Sharing Data Agents:** Bike Data Sharing Agents are responsible for obtaining data from the BSS, which usually offers two kinds of data:
  - **Station information:** It indicates the total number of available docks and bicycles on each station at a given time, that is, the current situation at the station. These data are usually available and provided to the end users in order to keep them informed about the stations with available bicycles. If the data are periodically polled, pickups and returns at each station can be calculated



regularly. This kind of information only provides us with knowledge about variations at each station without specifying the flow of bikes from one station to another or data regarding the users involved in these changes. It is usually provided through an API in real time by a great amount of BSS, so much so that there is a project called PyBikes [Esquerda; Lluís Esquerda] which unifies the access to this information through a common API for a large number of current BSS.

- **Trip information:** This sort of information is not commonly published but it is usually available in csv files offered every month as open data [[CSL STYLE ERROR: reference with no printed form.]]. It is related to the trips recorded by the BSS: a user picks a bike at a specific station at a given time and later returns it at another station. This kind of information discloses more details as it provides insights on the flow of bikes between stations and the users who perform the trips recorded in the system.
- **Weather Data Agents:** These agents obtain the weather data requested by the Processor Agent, they provide historical weather data for the requested dates and locations which will later be merged with trip or station data. These agents will also provide weather forecasts to Prediction Model Agents; they need this information when making predictions for a specific station at a given time.
- **Geographic Information Data Agents:** These agents obtain the geographic location, and the distance between stations as well as the altitude of each station. A base agent offers services related to this information and a specific agent implements this functionality with different data providers such as Open Source Routing Machine (OSRM) or Google Maps. This information will be stored and employed later by API agents in the visualization tool. This information will also be useful for future works where a reallocation bike strategy must be calculated.
- **Data Processor Agents:** Data Processor Agent processes the data obtained by Bike Sharing Data agents. Additionally, this agent will request the weather agents to provide information regarding the bike sharing data dates and it will clean and merge that information with the previously obtained bike sharing data. This process will be executed periodically in order to get all the recent information that is pending to be processed.
- **Demand Prediction Agents:** These agents will provide the demand forecasting for each station in the system. The main task of these agents is creating predictive models from historical data, which has been previously pre-processed by the Processor Agent. Apart from this, they must be able to provide demand prediction for a requested period. These agents generate new models periodically and they include last information obtained from the BSS. The predictive models are generated following the

methodology proposed in this work and the algorithms employed are those which had good results in previous studies such as [Regue and Recker 2014] [Malani et al.] [Prakash Nekkanti 2017] [Wang et al. 2016]. The algorithms applied are Random Forest Regressor [Breiman 2001], Gradient Boosting Regressor [Friedman and Friedman 2000b] and Extra Tree Regressor available on *framework* Scikit-learn [Pedregosa FABIANPEDREGOSA et al. 2011].

- **WEBAPI Agent:** This agent will offer access to historical data as well as demand prediction services through a Web Application Programming Interface (WebAPI). Likewise, this agent will provide a web application that will use the previous services to offer information to the BSS operator. The web application will be compound by two main sections:
  - **Historical data visualization:** This section will provide information on the BSS's historical data. It also offers visualizations of trip data information, allowing the user that is filtering historical data to explore user behaviour between stations.
  - **Forecasting & Status:** This section will show the current status of the system and the demand predictions for the selected station.

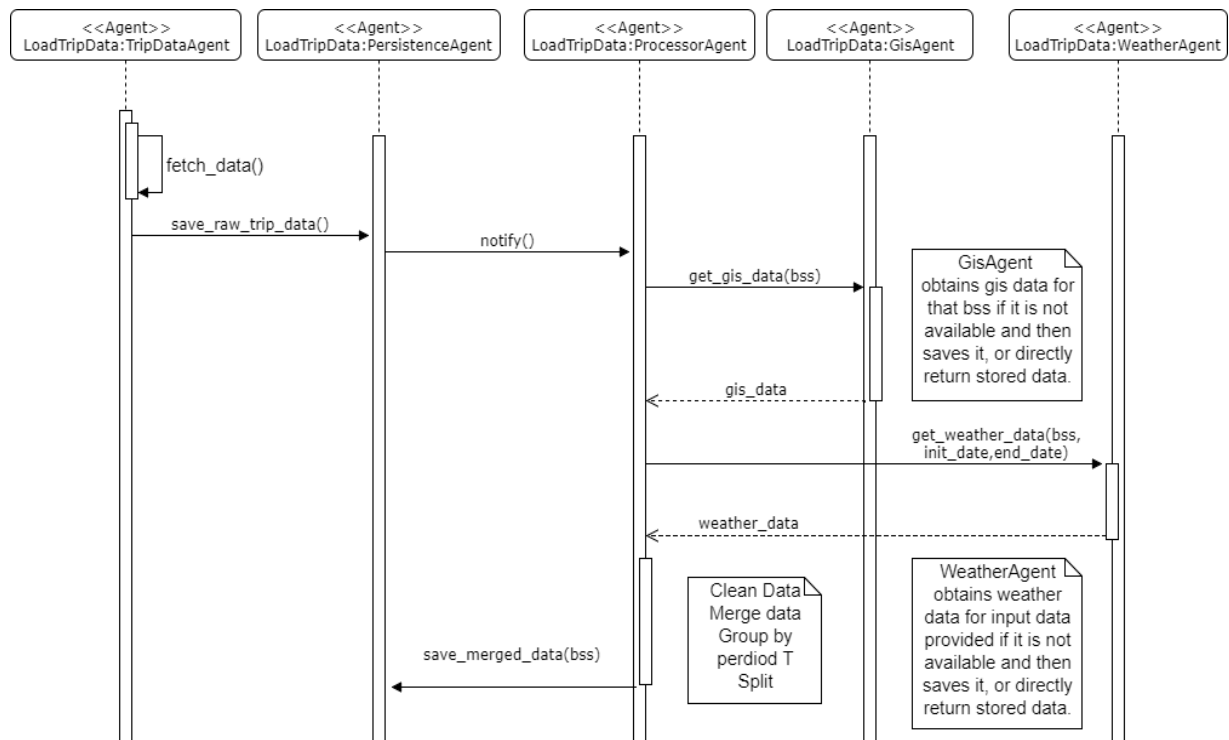
### 6.4.2.3 Data collection process

The data collected regards all trips recorded in the system from an origin station to a final station. In this case, the information provided by SalenBici company is trip data, as mentioned previously and it includes the following details in the original dataset: (1) Time Start: timestamp of the beginning of the trip, (2) Time End: timestamp of the end of the trip, (3) Bicycle ID: unique bike identifier, (4) Origin Station: origin station name, (5) End Station: destination station name, (6) Origin dock: origin dock identifier, (7) End dock: destination dock identifier, (8) User ID: user unique identifier.

The SalenBici company provided this information in csv files format, one file per year. To process this information, the files were given to the multi-agent system. The trip agent loads all the data from the provided data source, the processor agent is notified by the persistence agent because new raw data are added to the system. The processor agent checks the data and request geographic and weather data from the Geographic Information System (GIS) Agent and Weather Agents respectively. Before these agents respond to the request, they check the availability of data with the persistence agent, if the data are available they are sent immediately. On the other hand, if information is not available it is firstly collected and saved, and then sent to the processor agent. **Figure 27** shows a sequence diagram of the activities performed by each agent, the communication between the GIS and weather agents —the persistence agent is omitted for clarity.

After this process, the stored output data contain the following information on arrivals and departures: year, month, day of the week, day of the month, week of the year, hour, season

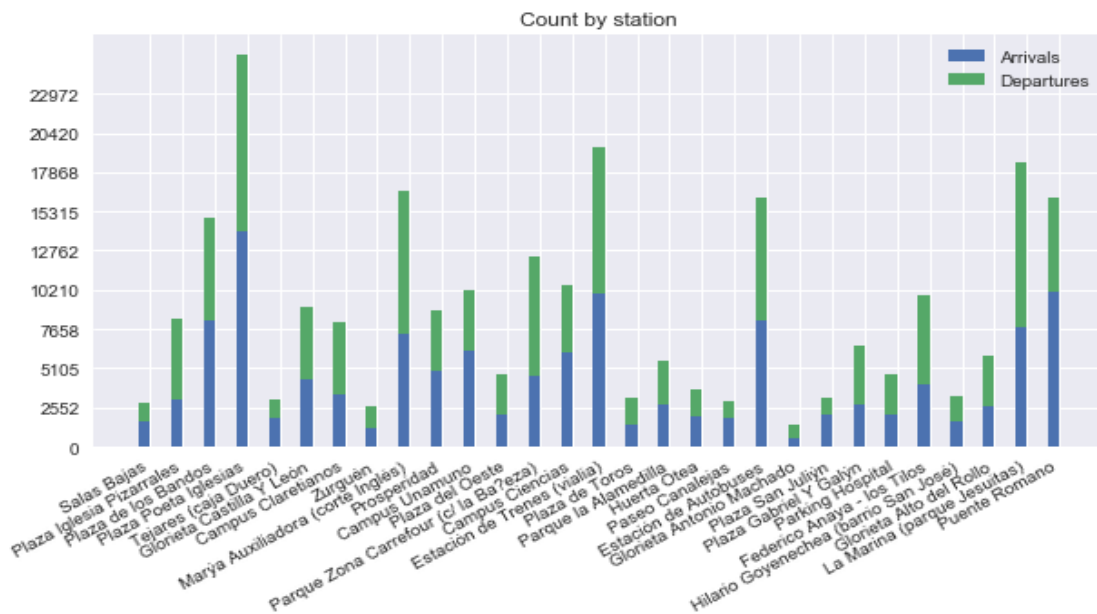
(winter, spring, summer or autumn), weekends and holidays as time-related information. Data available about weather on a particular day are: the maximum, the minimum and the average temperature in degrees Celsius, the average wind speed in km/h, the minimum and maximum atmospheric pressure in millibars and the amount of rainfall. The total amount of arrivals or departures in the grouped period T is a dependent variable.



**Figure 27** Sequence diagram of trip data loading by multiagent system

Before deploying the predictor data agent in the multi-agent system, an exploratory analysis of the processed data was performed. This analysis determined what model should be included in the predictor data agent.

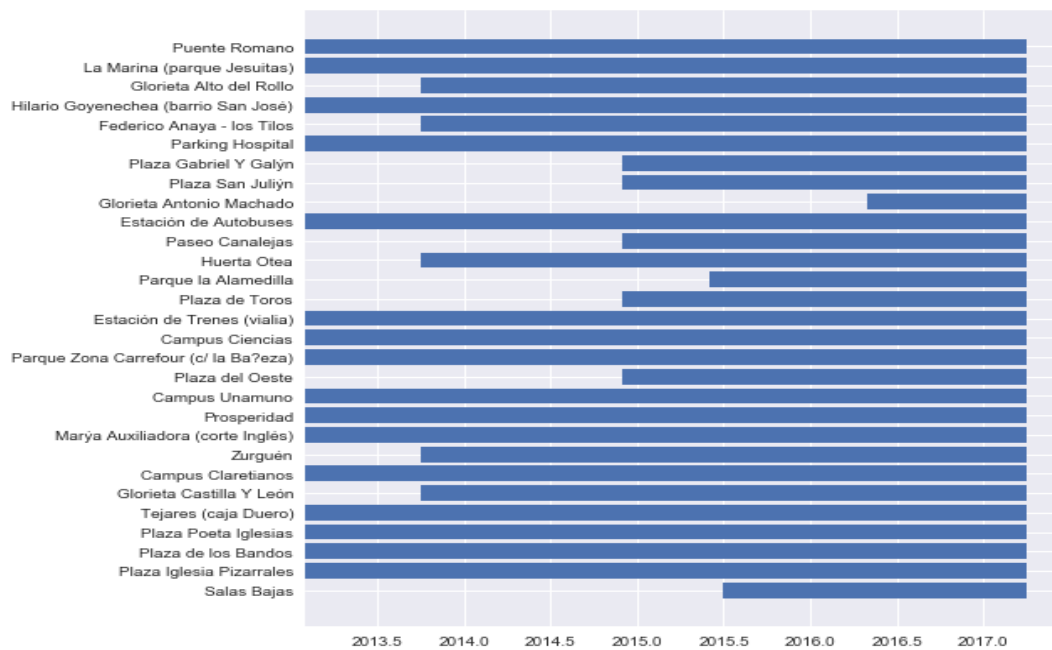
**Figure 28** shows the total number of trips loaded in the system, which are grouped in arrivals and departures for each of the stations in the system. These data date from January 2013 to March 2017, 1520 days in total and a clear difference between the use of stations in the system can be observed. There are a lot of stations where the mean of the bike trips is less than 2 events (arrivals or departures) per day. The station with the highest mean is called “Plaza Poeta Iglesias” near *Plaza Mayor* (the most touristic place in the city of Salamanca), with 15 events per day (arrivals or departures).



**Figure 28** Total number of events (arrivals or departures) performed in Bike sharing systems (BSS) SalenBici from January 2013 to March 2017

In addition,

**Figure 29** shows the period from which each of the stations in the system is active. When reading the total number of trips from the previous figure, an important aspect to be taken into account is that not all stations have been working since January 2013.



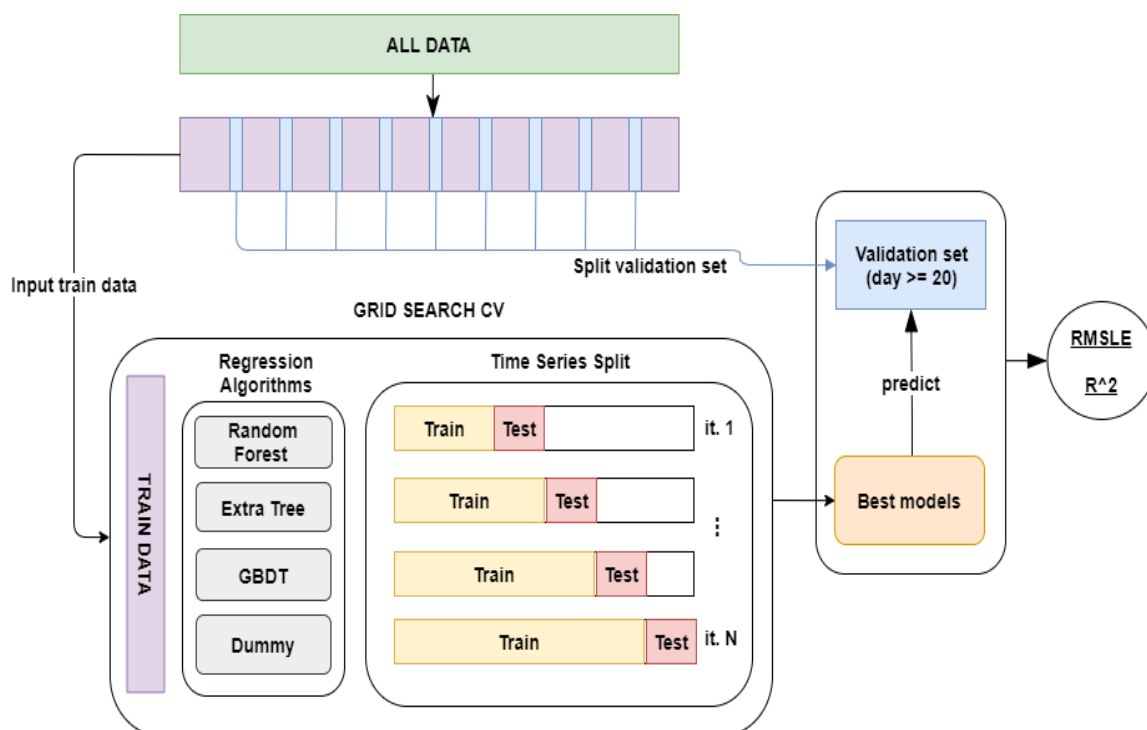
**Figure 29** Periods of working time of each station in the BSS SalenBici

### 6.4.2.4 Predictive model selection

This section describes how the BSS dataset was split and the methodology that was used in order to select the predictive models that will be included in the predictor agent. Like in Kaggle competitions, data have been split into two datasets; a training dataset and a validation dataset.

**Figure 30** shows schematically how the available data were employed in the training, selection and validation of the models used. At the top of the figure, the green part represents the entire dataset. Like in Kaggle competition [Kaggle INC 2014], a validation dataset is initially extracted and it comprises from the 20th to the last day of each month—in the diagram it is represented in blue—. The rest of the dataset, (those from the 1st to the 19th of each month), will be used as training data, — they are represented in violet in the diagram—. These data will be one of the inputs of the hyperparameter search technique: GridSearchCV [Bergstra et al.]. This technique will use the following as inputs: (1) regression algorithms with their corresponding parameter grid; (2) a scoring function in order to evaluate the input models (in this case RMSLE and R2) and (3) a cross-validation method, in this case TimeSeriesSplit, a method that is intended specifically for time series and which resembles the usual functioning of a system in production. This method allows to progressively use data from the past for training and use future data for validation. Its functioning can be observed on the diagram located at the bottom part under GridSearchCV, in

**Figure 30.**



**Figure 30** Scheme of the proposed methodology applied to BSS case study

The GridSearch method will make all the possible combinations for each algorithm with the provided grid parameters; this will be done by employing the cross-validation method (Time Series Split) and evaluating the trained methods with the provided scoring functions. As the output of this method, models for each algorithm with the best results will be obtained and these will be evaluated with the validation dataset that had been split at the beginning, on the right-hand side of

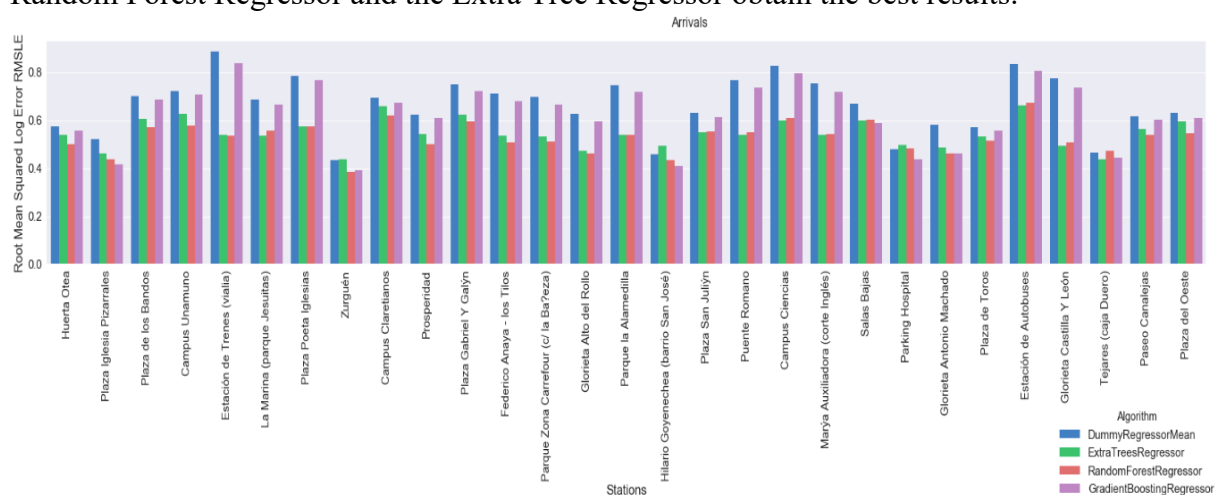
**Figure 30.** A Dummy Regressor has been added to the models used and was established as their prediction strategy in order to continually predict the average. The regression algorithms, as well as the following parameter grids, have been trained using GridSearchCV:

- Extra Tree Regressor: criterion: [mae, mse], number estimators: [10,100,1000], max features: [auto, sqrt, log2]
- Random Forest Regressor: criterion: [mae, mse], number estimators: [10,100,1000], max features: [auto, sqrt, log2]
- Gradient Boosting Regressor: learning rate: [0.1,0.01,0.001], subsample: [1.0, 0.9, 0.8], max. depth: [3,5,7], min. samples leaf: [1,3,5]

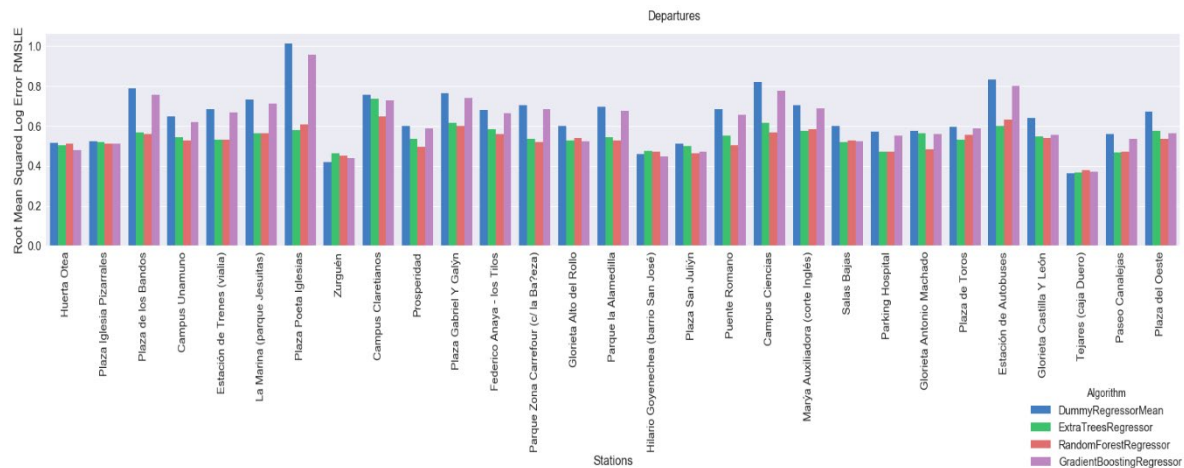
Once the model is selected, its results are validated with the validation dataset that had been split previously. Furthermore, the coefficient of determination  $R^2$  of the models selected for each station has been calculated.

### 6.4.2.5 Obtained results discussion

Next, the results of the RMSLE validation dataset are outlined. The results of the best models selected by the GridSearchCV for each station are described. **Figure 31** shows results for arrivals and **Figure 32** shows results for departures. From these results, the conclusion that can be drawn is that regressors tend to provide better results than the established baseline. The Random Forest Regressor and the Extra Tree Regressor obtain the best results.

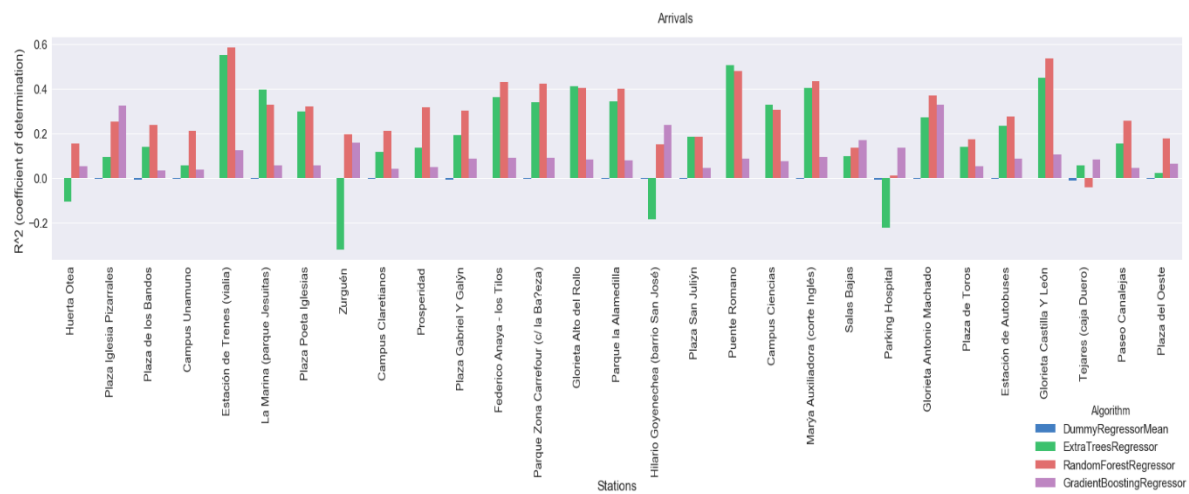


**Figure 31** Root Mean Square Logarithmic Error (RMSLE) provided by each algorithm for each station in the arrivals data

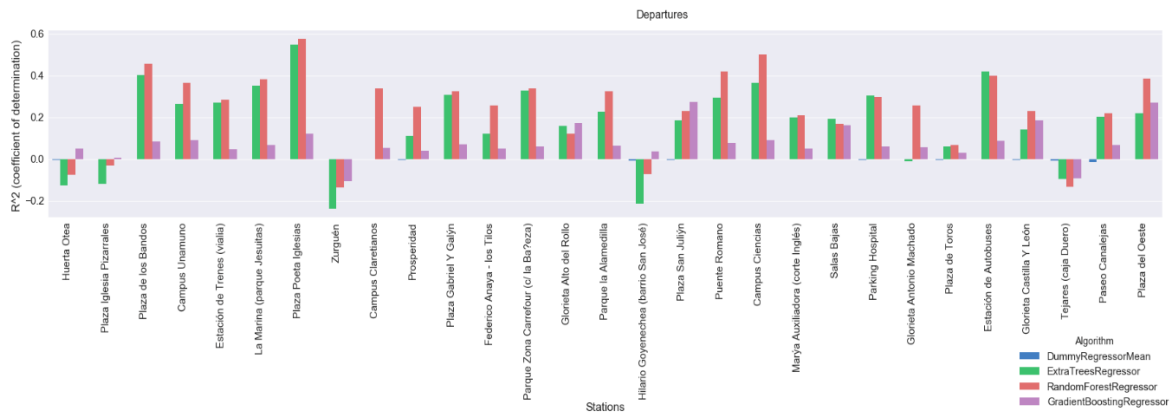


**Figure 32** RMSLE provided by each algorithm for each station in the departures data

Additionally, the coefficient of determination  $R^2$  has been calculated for each of the algorithms aiming to use it to select a model that will be included in the predictor agent. The results of the coefficient of determination are shown for each of the algorithms at each of the stations, —**Figure 33** for arrivals and **Figure 34** for departures—. From a graphical approach, the method that works best with  $R^2$  is Random Forest Regressor. However, the results will be analysed to see if this difference is significant.

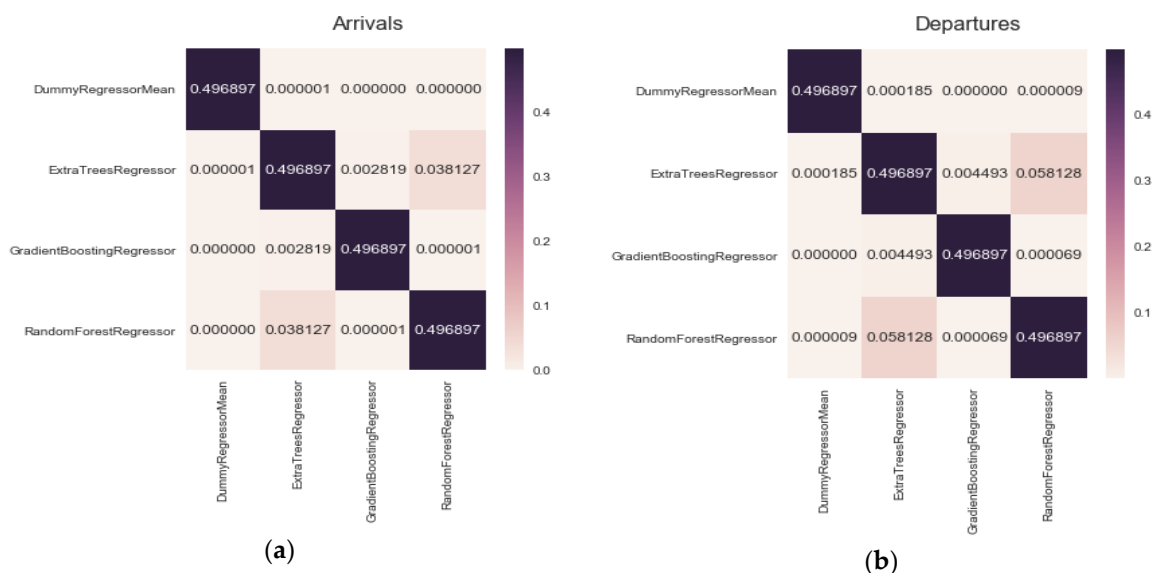


**Figure 33** Coefficient of determination of models for the arrivals at each station



**Figure 34** Coefficient of determination of models for departures from each station

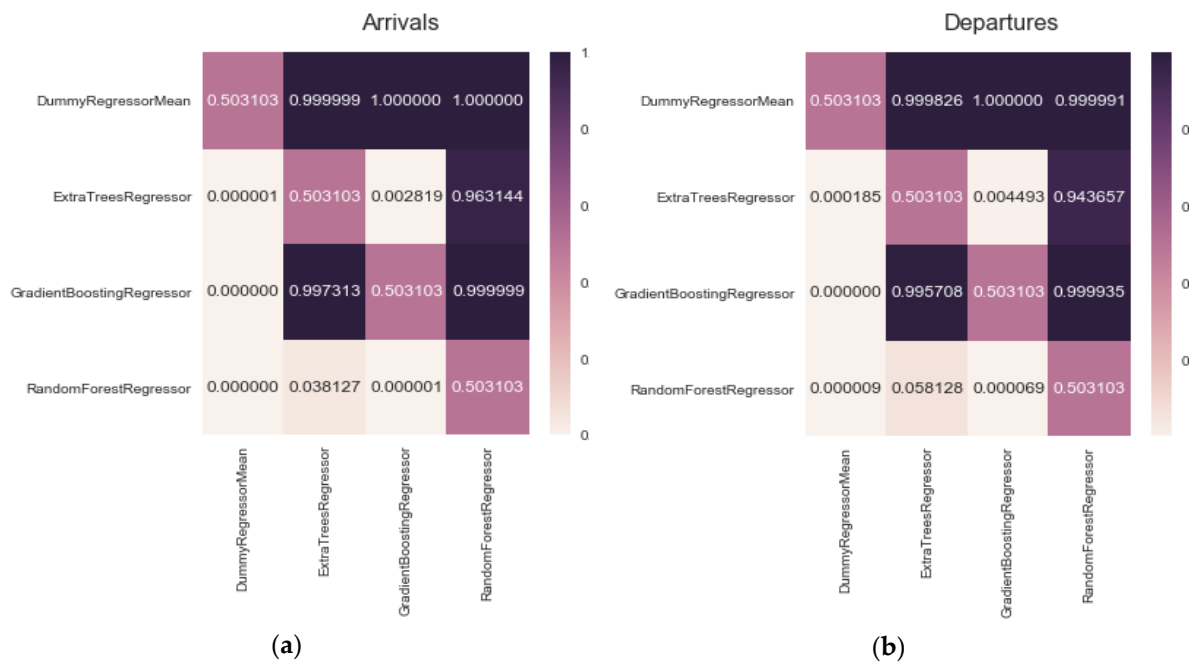
Using a statistical Mann-Whitney U test allowed to select the best algorithm for all the system stations. Firstly, the test was done in order to check if the functioning of one of the algorithms used is different from the others. The following confirmatory data analysis was applied:  $H_0$  considers the median of two equal methods while  $H_1$  considers the median of two different methods. As can be observed in **Figure 35**, in both departure and arrival models, the Random Forest Regressor Algorithm has a median that is different from the rest, so this could be considered as a significant statistical difference. The p-value obtained for the Random Forest Regressor and Extra Tree pair, surpasses slightly the test's 0.05 significance level and consequently this cannot be assumed as a significant statistical difference. However, by calculating the median of both methods the median for Random Forest Regressor is greater in comparison to that of ExtraTreesRegressor.



**Figure 35** Mann-Whitney U test two sided for (a) arrivals and (b) departures



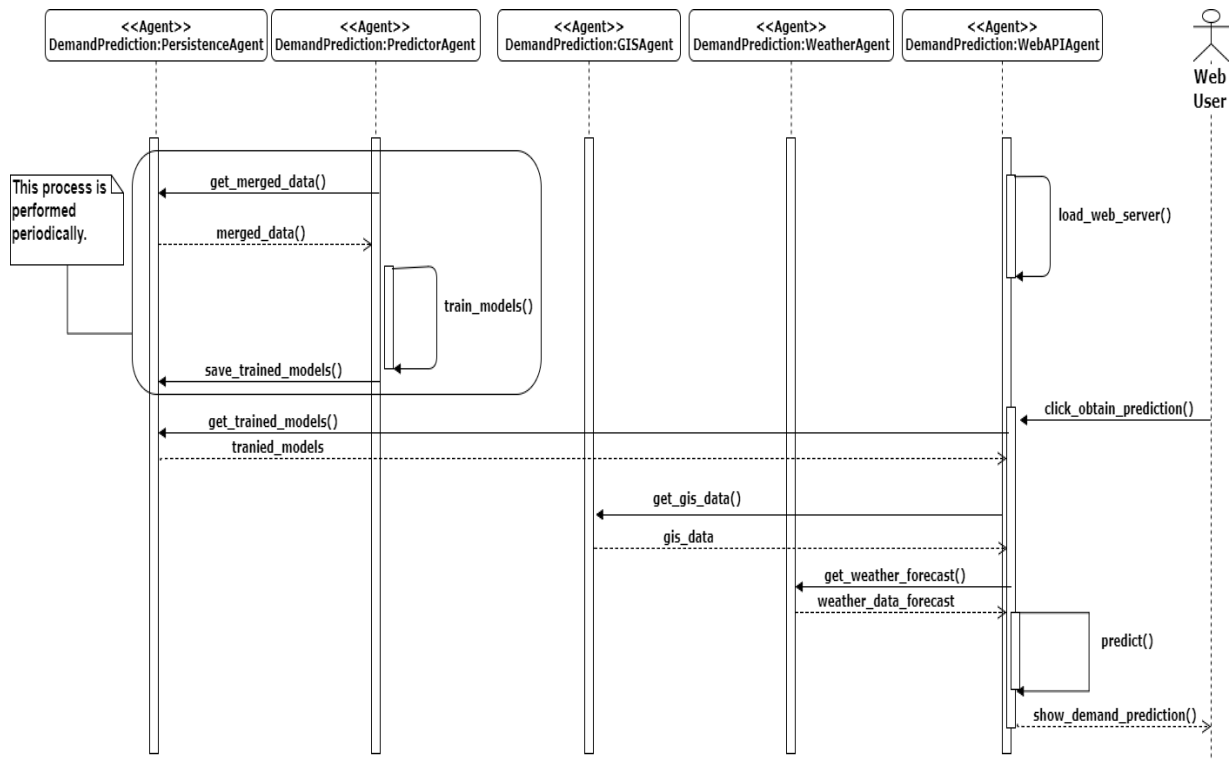
Once this test is completed, it must be repeated if significant statistical differences are detected, to the following step is determining whether the median is smaller or greater. In this case, the defined  $H_1$  states that the median of the classifier from the row is greater than the median of the classifier from the column. **Figure 36** verifies that Random Forest Regressor has a greater median than the rest of the algorithms for all the stations.



**Figure 36** Mann-Whitney U test greater for (a) arrivals and (b) departures

After seeing these results, Random Forest Regressor has been selected for inclusion in the predictor agent; it will generate models and store them to subsequently provide predictions through the WebAPI agent. **Figure 37** shows the sequence diagram for performing the predictions. The predictor agent periodically trains and saves the model for each system station. Later, these models are used by the WebAPI agent in order to show predictions on the web application and the API REST.

The WebAPI agent provides a web application where users can request predictions for a selected station. This agent will communicate with the weather agent, the GIS agent and the persistence agent in order to obtain the information it needs to make the prediction requested by a user. In the next section, the web application will be explained in detail.

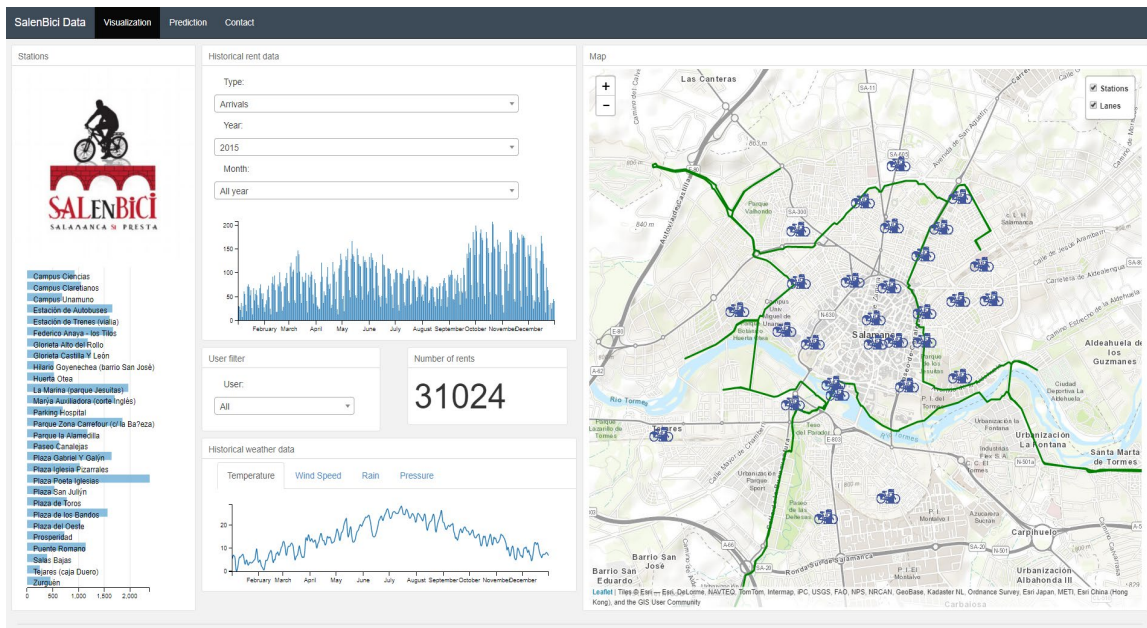


**Figure 37** Sequence diagram of model prediction generation and forecasting request

#### 6.4.2.6 Visualization and demand prediction web tool

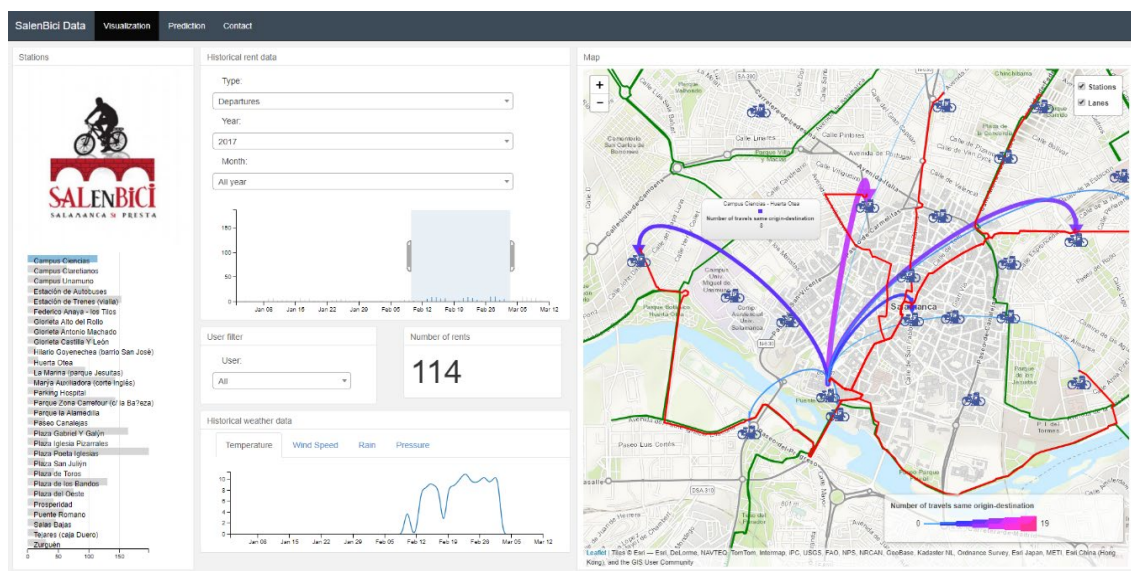
The WebAPI agent offers an API REST for third-party applications which can obtain the data processed in the MAS and ask for demand predictions of the stations in the system. In addition, there is a web application where users can make visualization and prediction tasks. The web application consists of two sections; **Figure 38** shows an initial view of the visualization section. This section presents the trips recorded in the system per year and month.

The visualization feature has various sections where data can be looked up by filtering by station, time (year, month and day), trip (arrivals or departures) and user. These sections act as filters and thus allows to visualize a series of specific data **Figure 39**.displays the map with departures from the selected station, with the period indicated in the left corner of the image. The number of trips at the selected station is represented by an arrow, the colour scale and width are proportional to the number of trips made between the origin station and the arrival one.



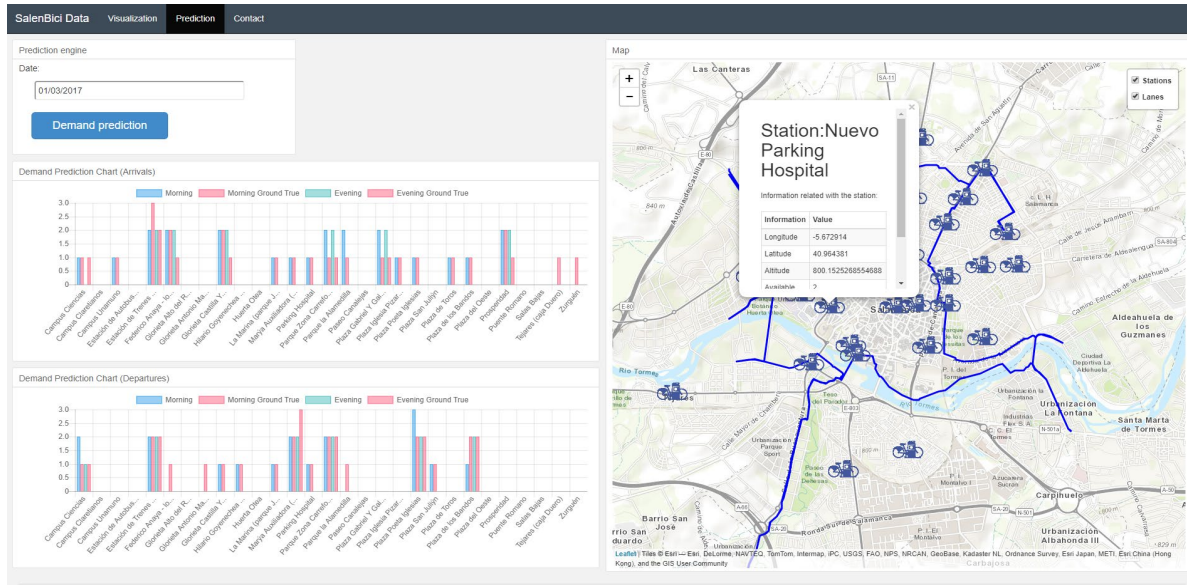
**Figure 38** Web application provided by the WebAPI Agent. Section for visualizing historical data

Both can be selected, an arrow from the past and an arrow for the future. An arrow from the past, apart from showing the data collected by the system, will also visualize the prediction made by the model and the real data that were collected for that station, within the indicated date. These data will be useful to BSS administrators at the time of relocating the bicycles at the stations.



**Figure 39** Visualization section. Where a station is selected for a period of time and the number of trips (origin-destination) is visualized.

**Figure 40** shows the demand prediction section. The selected models will be used to predict demand at each station.



**Figure 40** Prediction tool section

### 6.4.2.7 Conclusions

Throughout this case study, a Multi-agent system oriented to Bike Sharing Systems has been developed and evaluated with data from the SalenBici system of the city of Salamanca. The modelling of the system as a multi-agent system has been especially useful to collect data from different sources and decouple each of the parts of the system. The proposed architecture would allow data to be obtained from other BSS systems in a simple way.

Similarly, the use of agents for predictive tasks has allowed us to use information collected by other agents and offer services to third-party applications. The modelling of a Predictor Agent in an isolated way makes possible the implementation of prediction systems decoupled from other systems, allowing the reuse of code for other cases of use in which it is necessary to implement, train and use prediction models of demand.

In this case study, different algorithms of the system proposed in this thesis have been evaluated and applied to a case study with real data about the demand for a BSS. Their performance, precision and training time have been taken into account and statistical tests have been carried out to determine the use of them in the final implementation of the system (due to the time necessary to train all the models following the proposed methodology). In this case, the algorithm that has obtained the best results compared to the rest is Random Forest Regressor.

---

Additionally, an agent has been implemented that provides a web application as a tool to visualize historical data and another one to predict the demand in the stations (using the generated models).

Regarding the visualization part, the operator can focus on the parts of the historical data of his interest thanks to the filters implemented and the visualization carried out. This tool allows obtaining knowledge about the behaviour of the system stations and how their users behave.

The prediction part of the web application can be used by operators to relocate bicycles throughout the city.

Finally, the predictions obtained can be used in future work of optimization of relocation routes of the fleet of bicycles improving current strategies.

### 6.4.3 Simheuristics combined with demand predictive models

This case study deals with the inclusion of uncertainty in the optimisation of waste collection routes employing the Simheuristics methodology and the modification proposed in this work.

In order to evaluate whether using predictive models in the Simheuristics methodology [Juan et al. 2011] improves the optimization results as opposed to using the expected value of the distribution of waste generated, a theoretical case study has been developed in which a simulation of collection data for classical VRP instances of the literature has been performed. The VRP instances have previously been employed by Juan et al. in their work to validate their methodology.

These types of instances can be found in [COIN-OR] and represent different theoretical CVRP problems with their corresponding parameters, such as the number of customers with their respective demands, the available fleet of vehicles, and a cost matrix. Usually, these instances are accompanied by the best solution obtained so far with their respective cost.

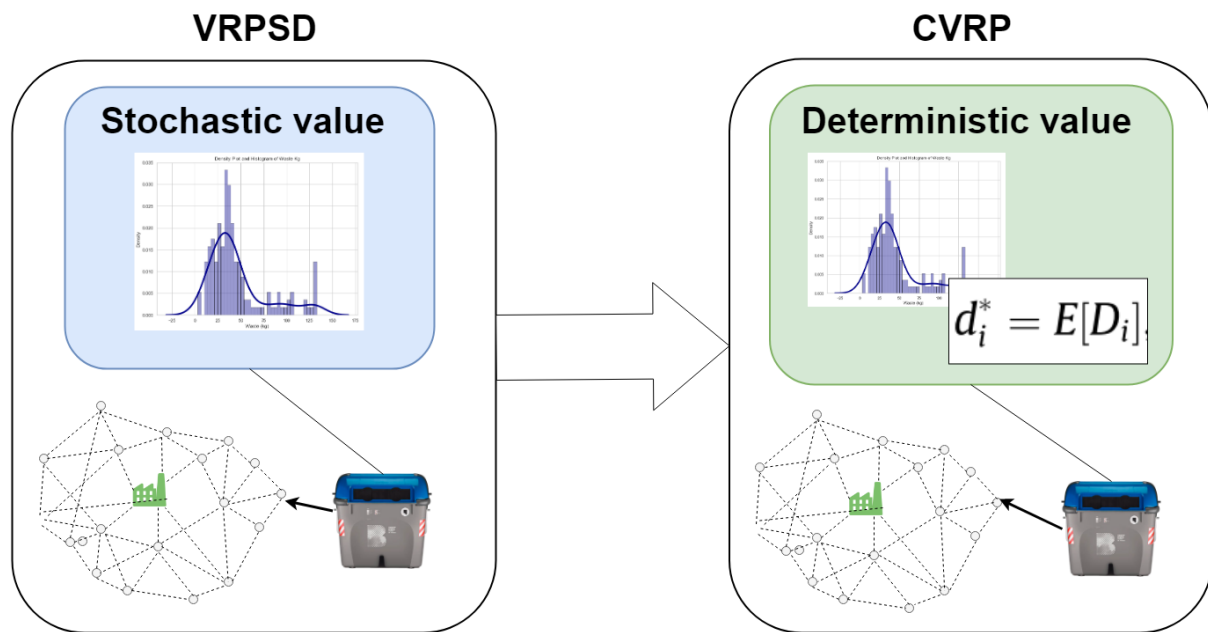
Given that these instances are CVRP problems and not VRPSD, Juan et al. take the demands of each of these problems as the mean value of a Log-Normal distribution converting these demands into stochastic demands and then perform the simulations defined in their methodology with these distributions.

The modification proposed by this work in the Simheuristics methodology is described below.

#### 6.4.3.1 Combination of Simheuristics with prediction models

In this thesis work, we propose to modify the Simheuristics methodology previously described in the system proposed. This modification is specifically focused on one point of the methodology process, that is, the simplification process performed by the authors when a stochastic VRPSD problem is converted into a deterministic CVRP problem.

At this point, Juan et al. [Juan et al. 2011], propose to simplify the problem using the expected value of the distribution (theoretical or empirical) that follows each container, in this way the VRPSD problem becomes a CVRP problem as it is described in **Figure 41**.



**Figure 41** Simplification process in Simheuristic methodology

Each of the containers, therefore, takes the expected value described by equation (13):

$$d_i^* = E[D_i] \tag{13}$$

Where  $D_i$  is the stochastic demand of each container and follows a defined statistical distribution (theoretical or empirical) and  $E[D_i]$  symbolizes the expected value of that distribution.

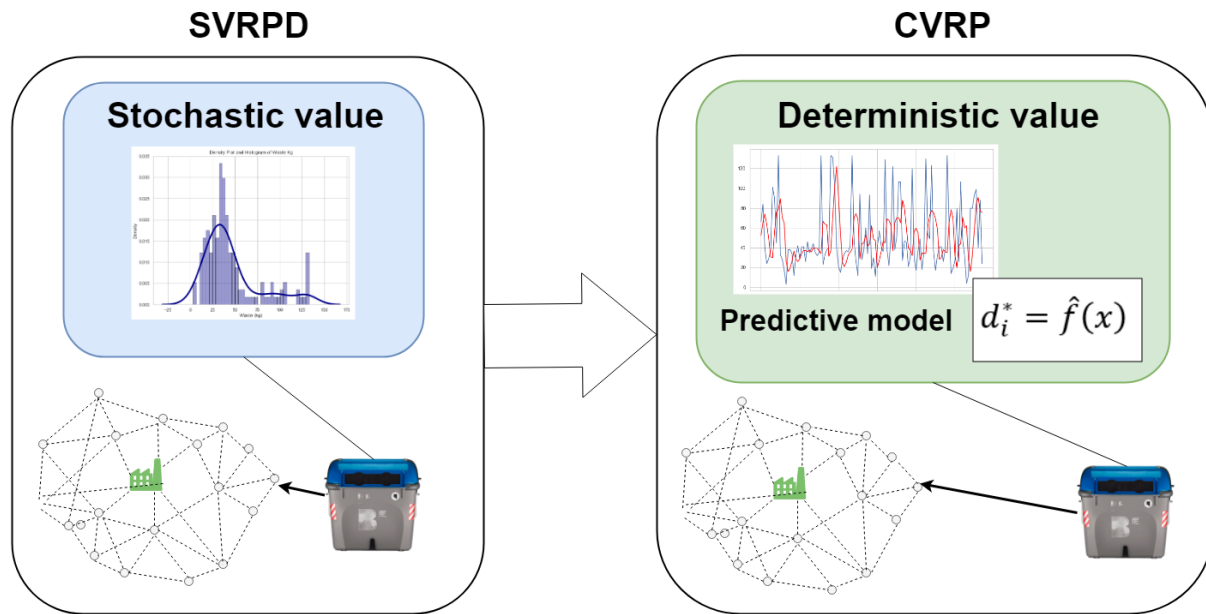
The proposed modification is based on the fact that there may be patterns that guide the behaviour of the demand and that these patterns could be identified and learned by prediction models. In this way, the value provided by the predictor of the demand of each one of the containers could offer a value closer to the future value that will be finally found during the collection. Therefore, if the functioning of the predictors presents an error lower than the one produced by the expected value, the calculation of solutions for the deterministic CVRP case should be closer to the stochastic case solution.

In this case, the values set for a container are produced according to the following equation (14):

$$d_i^* = \hat{f}(x) \tag{14}$$



Where  $\hat{f}(x)$  represents the predictor that has obtained the best results after the training process and  $x$  the input variables for that predictor for the day of collection in question. An illustration of the process can be seen in **Figure 42**.



**Figure 42** Simplification process proposed employing a predictive model for Simheuristic methodology

In this case study, an initial comparison is made between the prediction models and the expected value of the last month of collection for different days of collection of a waste collection system currently in operation. In this way, it is possible to check the error produced by each of the approaches when estimating the real total demand quantity of the containers. Subsequently, once the simplification from SVRP to CVRP has been carried out, the process described by the Simheuristics methodology is continued. To this end, simulations will be carried out following a Log-Normal distribution and a 0.25 mean-variance scenario. Like Juan et al. 's work, the mean and variance for the simulations are described by equations (15) and (16):

$$\mu_i = \ln(E[D_i]) - \frac{1}{2} \cdot \ln\left(1 + \frac{\text{Var}[D_i]}{E[D_i]^2}\right) \quad (15)$$

$$\sigma = \sqrt{\ln\left(1 + \frac{\text{Var}[D_i]}{E[D_i]^2}\right)} \quad (16)$$



Once the simulations have been carried out, the solutions obtained for each of the two approaches are compared (using the expected value and using the best prediction model) and their feasibility indices and mean simulation costs.

### 6.4.3.2 Case study

Once the modification to the methodology has been described, a theoretical case study is presented in which data related to a CVRP problem such as that found in waste collection systems are simulated. It should be noted that the proposed modification can be used in any a CVRP problem regardless of the specific case study (e.g. waste generation).

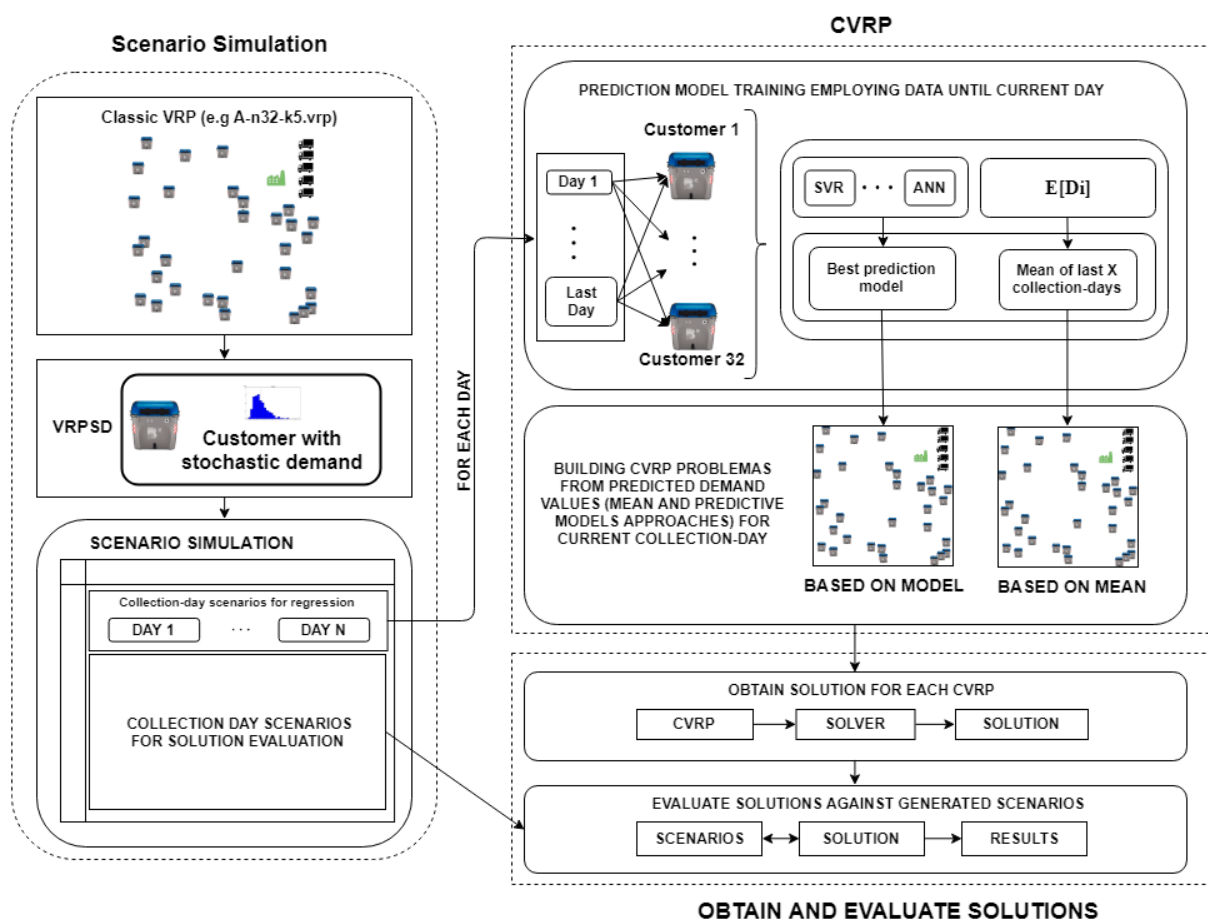


Figure 43 Case study main diagram

Figure 43 shows a general picture of the proposed case study with the phases of the process. Left side pictures the simulation process of the data as well as the scenarios for the subsequent evaluation of solutions. Right side describes the steps followed to get the solutions: firstly, the prediction models are trained with the generated data and then, predictors are obtained from

the average of the last collections (simplification approach of Juan et al.). Once the models are obtained, the demands of each of the customers are generated to get 2 CVRP problems, one based on the use of the mean and the other based on the use of the value predicted by the regressor. Next, a process of calculating solutions for the problems raised is carried out and finally solutions are evaluated against the set of scenarios initially generated. Later sections explain each of them in depth and finally present the results obtained after this process.

#### 6.4.3.2.1 Scenario generation

The parameters (**Table 4**) and the process carried out to generate of the scenarios for the case study are described below. This section corresponds to the left part of the diagram presented above. Demand scenarios (the amount of waste generated in each container) have been generated for each day of the year between *initDate* and *endDate*. Classical CVRP instances from the literature (*InstanceCVRP*) have been used as a basis, such as instance A-n32-k5.vrp, which represents a CVRP problem of 32 clients and 5 vehicles with certain costs and demands. This instance will be used to exemplify the process followed in the case study.

**Table 4** Parameters employed for generating scenarios for instance A-n32-k5.vrp

Parameter	Definition	Value
<i>InstanceCVRP</i>	CVRP instance	A-n32-k5.vrp
<i>numScenarios</i>	Amount of scenarios	10000
<i>collectionDays</i>	Amount of days between every collection day	3 days
<i>initDate</i>	Init date of simulation	01/01/2017
<i>endDate</i>	End date of simulation	01/01/2019
<i>calendar</i>	Country work calendar for holidays	Spain
<i>variance</i>	Variance for distributions employed to generate demand values	0.25
<i>percentageHolidays</i>	Percentage of increase over the average that is taken for each customer on a bank holiday	0.8
<i>seed</i>	Random number seed	7

The distributions used to generate the data are those used by Juan et al. in his work [Juan et al. 2011], described above, that is, the demand of each of the clients of the CVRP instance is taken as the average of the distribution used to generate the demand values of the scenarios.

For the case study, two distributions have been established for the generation of demand values: the first corresponds to the days from Monday to Friday which are labour days and the other

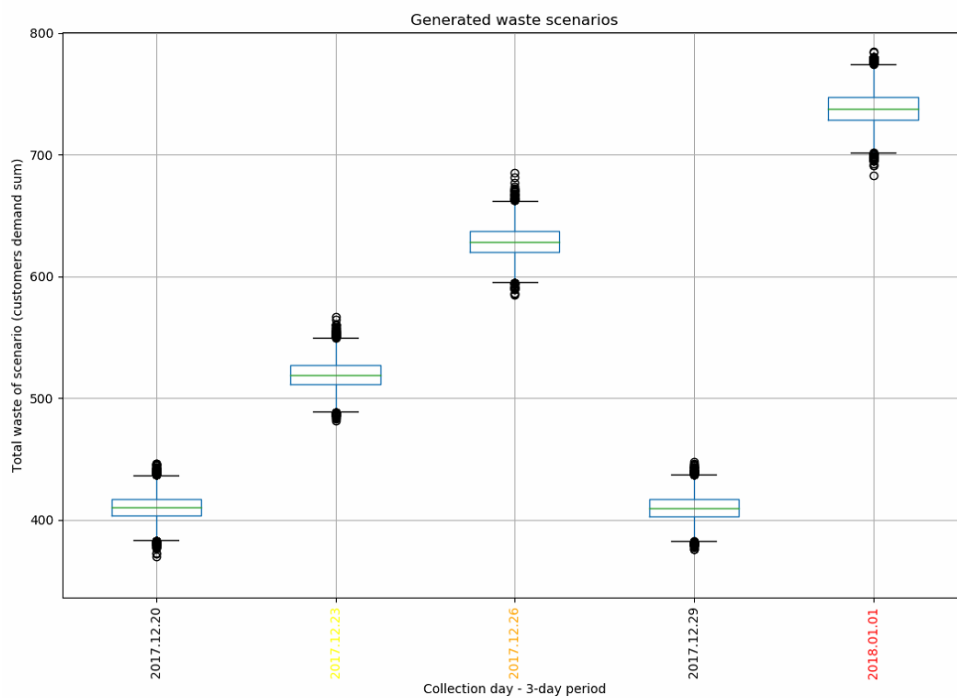
corresponds to the rest of the public holidays and weekends described in the following equations (17) and (18).

$$d_{iworkdays} = \ln(E[D_i]) - \frac{1}{2} \cdot \ln\left(1 + \frac{Var[D_i]}{E[D_i]^2}\right) \tag{17}$$

$$d_{iholydays} = d_{iworkdays} + (d_{iworkdays} \cdot percentageHolidays) \tag{18}$$

With this, it has been established as an arbitrary pattern that weekends and holidays take a distribution which average is a percentage higher than the rest (*percentageHolidays*). This is intended to introduce a pattern in the generation of data that is expected to be detected by the prediction algorithms used.

Next, the data are grouped into collections, simulating the behaviour that would occur in a conventional garbage collection system (parameter *collectionDays*). To ensure the reproducibility of the experiment, a common seed of random numbers has been established in the used libraries (*seed*).



**Figure 44** Box plot of total demand from simulated scenarios for 5 collection days (3-day period)

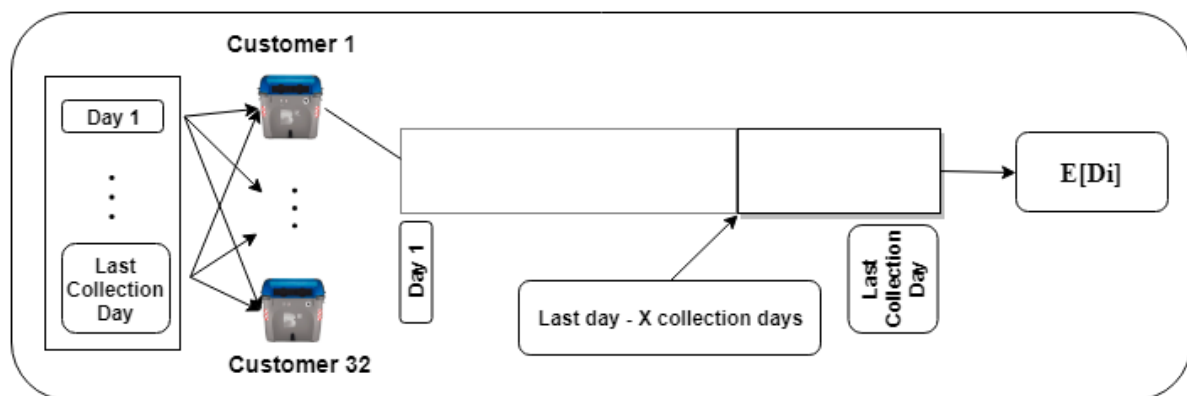
In the **Figure 44**, an extract of the data generated for 5 collection days is shown. The black collections represent those days of collection without previous public days, whereas the yellow, orange and red ones are the collections days with a higher percentage of public holidays prior the collection. It is possible to appreciate how the total demands of the generated scenarios are greater in those collected that involve more public holidays.

#### 6.4.3.2.2 Generating deterministic CVRP

This phase of the case study corresponds to the top right of **Figure 43**. The process of simplification from VRPSD to CVRP is carried out from two different approaches: on the one hand using the mean value of the last  $X$  collection days and, on the other hand, using prediction models trained with historical data up to that moment. The data used for the mean and the training of the prediction models correspond to a single set of scenarios for the days specified above. The two processes used to obtain the deterministic values of CVRP problems are described below.

#### 6.4.3.2.3 Mean

For the average process, the available information will be taken from a temporal window of  $X$  collected. This process will be carried out for each of the customers involved in the problem. **Figure 45** diagram shows the process for each day.



**Figure 45** Data used for each container to obtain the average for the simplification process

The value calculated for each customer will be the one used for the prediction for the demand of the day that is being treated at that moment. This will be applied to each collection day for the specified period between *initDate* and *endDate*

#### 6.4.3.2.4 Predictive models

In this case, the information from all the collection days is used to train different prediction algorithms in order to obtain an accurate forecast for the day being treated. Table 5 shows the characteristics selected to train the regressors. On **Table 5**,  $t$  represents the current collection and a total of 7 lag features, not included in the table for brevity, have been considered both for demand and holiday percentage.

**Table 5** Input features for regression algorithms

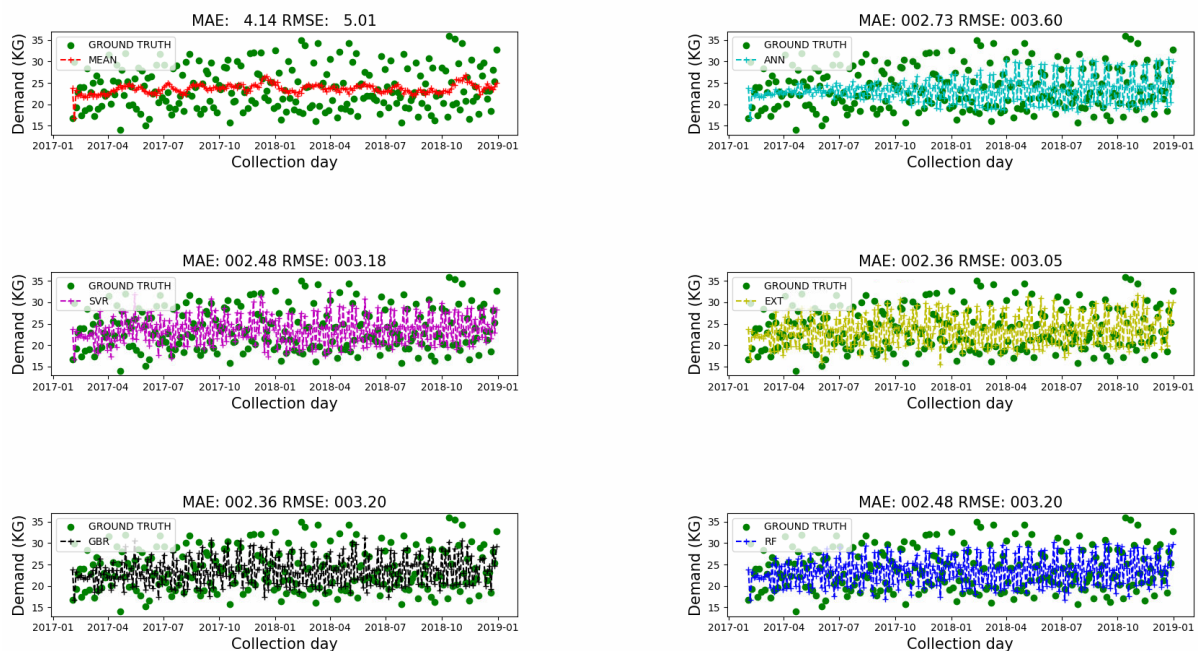
Feature	Description	Type
$demand(t - 1)$	Demand of last collection day (kg)	Continuous
...	...	
$demand(t - 7)$	Demand of 7 <sup>th</sup> collection before (kg)	Continuous
$holiday\_percentage$	Percentage of holidays up to the current day [0 - 1]	Continuous
$holiday\_percentage(t - 1)$	Percentage of holidays up to the last collection day [0 - 1]	Continuous
...	...	
$holiday\_percentage(t - 7)$	Percentage of holidays up to the 7 <sup>th</sup> collection day before [0 - 1]	Categorical
$week\_day$	Current week day [0-6]	Categorical
$month$	Current month [1-12]	Categorical

Furthermore, **Table 6** presents the regression algorithms employed to generate demand prediction models and the parameter grid used for the selection of hyperparameters using GridSearch together with TimeSeriesSplit cross validation scheme.

The process of training and selection of models has been carried out following the scheme described in the proposed system. The model with the lowest Mean Absolute Error has been selected after this process and has been used to generate the prediction of the next day of collection for each of the customers and the days involved in the simulation.

**Table 6** Parameter grids for GridSearch

Algoritmo	Parameter GRID
Support Vector Regressor	{"C": [1e0, 1e1, 1e2, 1e3], "gamma": np.logspace(-2, 2, 5)}
GradientBoostingRegressor	{learning rate: [0.1, 0.01, 0.001], subsample: [1.0, 0.9, 0.8], max depth: [3, 5, 7], min samples leaf: [1, 3, 5]}
ExtraTreeRegressor	{number estimators: [10, 100, 1000], max features: [auto, sqrt, log2]}
RandomForestRegressor	{number estimators: [10, 100, 1000], max features: [auto, sqrt, log2]}
ANN	{optimizer: [sgd, adam], epochs [10, 20, 30], activations = ['relu', 'tanh']}

**Figure 46** Demand prediction model results for a customer in the instance A-n32-k5.vrp

**Figure 46** shows the predictions for one of the containers of instance A-n32-k5.vrp, as well as it shows the values of the scenarios versus the predictions of the mean and the regressor that has performed better in the previous process.

Therefore, after this process, as many CVRP instances have been obtained as collection days have been raised in the study, both for the mean approach and for the prediction model approach. These CVRP problems will be solved later and the solutions obtained will be evaluated with the rest of the previously generated scenarios.

### 6.4.3.3 Solving and evaluating against scenarios CVRP instances

After the generation of the deterministic CVRP instances for each one of the collection days, the solutions to these instances have been calculated using the parameters shown in **Table 7**. The Google OR-Tools library [Google 2019] has been employed for the implementation of this part of the case study.

**Table 7** Parameters for CVRP solver

Parameter	Description	Value
<i>firsSolutionStrategy</i>	The method employed to find the first solution	PARALLEL_CHEAPEST_INSERTION
<i>LocalSearchMetaheuristic</i>	Local search strategy performed	GUIDED_LOCAL_SEARCH
<i>timeLimit</i>	Solution search time limit	15 minutes

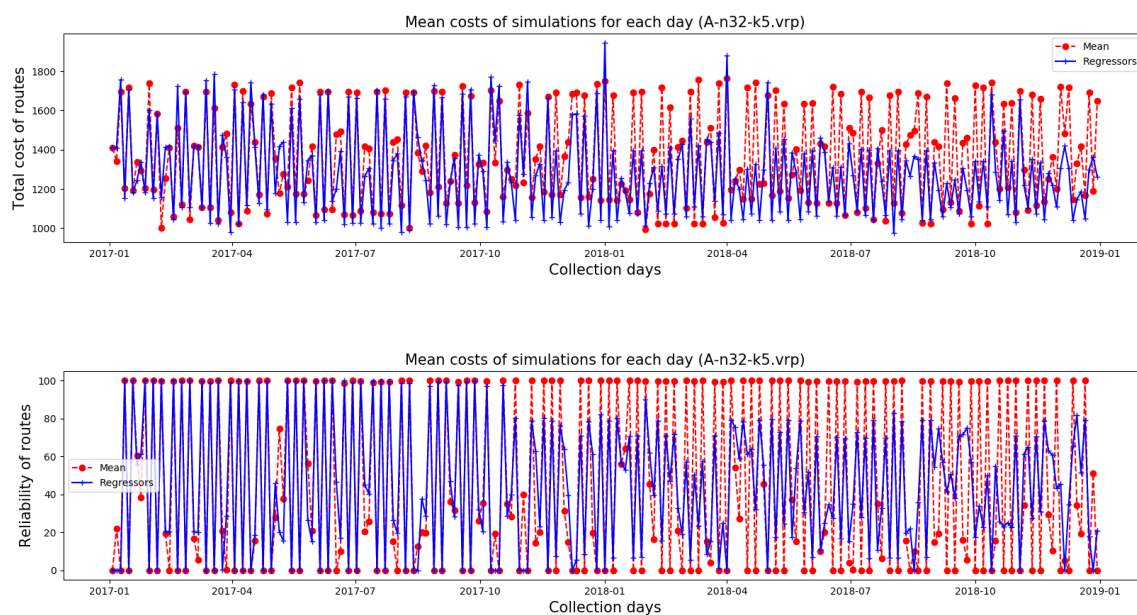
These parameters will be used to solve previously generated CVRP problems. For each of these problems, another set of problems will be generated with different  $k$  values between 0.95 and 1.0. After this process, the solutions will be obtained (set of routes, vehicles and associated costs) for each CVRP problem established by the average approach and by the regressors approach.

The solutions will be then evaluated against previously generated scenarios. This evaluation consists of applying the solution to each generated scenario. This evaluation could end on being able to carry out the routes or, on the contrary, producing excesses of demand in certain points and therefore producing recourse actions that will increase the cost of the solution.

After this evaluation, the solution with the lowest average simulation cost is selected, that is, the average evaluation cost of all evaluations. After the solution evaluation process, the average simulation cost, the  $k$  safety stock index used in each case and a reliability percentage of the solution are obtained for each one, that is, a percentage value that expresses the number of simulation problems that could have been carried out without performing recourse actions.

Below, **Figure 47** shows the results of the average costs of simulation for the problems generated, both with the mean approach and with the prediction models approach. The average costs of the 10000 simulations for each day of collection are presented together with the feasibility values of the routes.

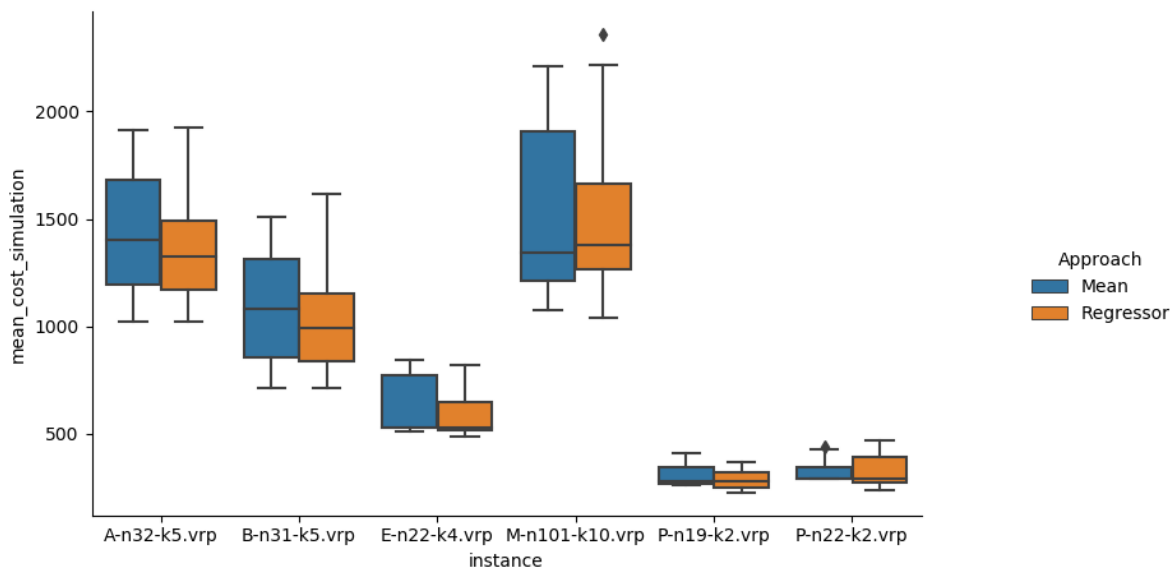
The upper chart of **Figure 47** shows the same results for the first collections because the regressors did not have much information to make the inference. However, as time goes by, the regressors can better estimate demand and produce solutions with a lower simulation cost. On the bottom chart, the reliability percentage is displayed, and it shows that the regressors present a more robust reliability percentage than the mean approach, since the last one proposes solutions with a reliability percentage either very high (due to a solution with a high cost), or null (due to a very poor adjustment to the scenarios). It is possible to observe that the feasibility in the case of the average is in the half of cases practically 100% because a solution is proposed with a very high cost or 0% because a solution has been proposed with a very low cost. In the case of the regressors, they offer an average percentage with a lower simulation cost.



**Figure 47** Cost results for each day employing mean and predicted values

The case study has been applied to different CVRP instances widely used in the literature such as those studied by Juan et al. The results are shown in **Figure 48**. Mann Whitney's U statistical test was performed for each instance studied. In instances A-n32-k5, B-n31-k5, E-n22-k4 and P-n19-k2 the test has determined that there are significant statistical differences to affirm that the results obtained from the regressors are lower than the mean. On the other hand, for instances P-n22-k2 and M-n101-k10 no statistically significant differences were found.





**Figure 48** Cost results for each day employing mean and predicted values

#### 6.4.3.4 Conclusions

After conducting the case study and evaluating each of the parts, different conclusions are extracted regarding the results of the prediction models, the simplification modification proposed in this work and, finally, the results in cost obtained in the optimization.

Firstly, prediction models have been trained using different state of the art algorithms for predicting the demand in each waste container. The values obtained from these models have been compared with those obtained by the mean of the last collection days, offering a better performance as could be expected. It is important to highlight that the models generated could capture other patterns than those introduced in the case study, such as trends in certain months or weeks of the year or peaks around specific dates of the year.

Secondly, regarding the results obtained in the case study, we see that the solutions provided by the prediction models are closer to the stochastic case, allowing to reduce the simulation costs and resulting in more robust solutions.

It is worth mentioning that such modification is only useful in cases in where the prediction models present a great performance in comparison to obtaining the expected value of the distribution that follows the stochastic variable. Besides, it should be noted that this proposal, as opposed to the one used by Juan et al., incorporates more dimensions to the problem (all of them related to a time series and the additional variables they include).

Another point to highlight is that the results in the prediction models are directly related to the quality of the solution obtained later. Therefore, a good adjustment of the model will facilitate better results since the stochastic scenario will be closer to the deterministic obtained.

After carrying out the case study, the use of this same modification is proposed in other case studies where the Simheuristics methodology is applied since the proposed approach can be transferred to other application domains beyond the Smart Waste Collection Systems.

## 6.5 Conclusions and future lines of work

This chapter brings together the general conclusions drawn from the research performed in the course of this thesis work. The contributions made with this work and the lines of research to be followed in future works are also described.

### 6.5.1 Research contributions

This part summarizes the main contributions of this thesis work. As main contributions regarding the study and review of the state of the art, the following can be highlighted:

- A review of the state of the art of wireless sensor networks, which can be used to collect data from intelligent logistics systems, has been performed. The different types of networks have been studied, specifically, the so-called LPWAN networks and, within these, their characteristics have been reviewed and compared, both at the level of energy efficiency and schemes of use.
- A study of different algorithms used in the predictive demand modelling has been performed as well as a review of different techniques, which have been used in the literature to generate demand prediction models in different domains.
- Likewise, a review of the formulation of VRP problems, which are widely used in logistics systems, has been prepared. Next, a study of their taxonomy, their solving methods and, finally, a specific type of these problems has been studied in depth, addressing the inclusion of uncertainty in their resolution. Solving methods and new methodologies have been revised to approach this type of problem.

After reviewing the state of the art related to the main topics of the thesis, the design of an intelligent logistics system with the following characteristics has been presented in this work:

- Incorporation of wireless sensor networks together with low consumption sensors to obtain data remotely.
- Use of the data obtained to generate demand prediction models able to model the amount of waste that will be generated in each container.
- Inclusion of uncertainty in route optimization methods and use of the above prediction models to achieve better solutions.

For the evaluation and design of this system, different individualized case studies have been elaborated that have allowed the research to focus on each of the parts that make up the system.

In addition, this working approach is fully in line with the methodology *action research* that promotes the realization of case studies that through action and obtaining results refines the proposed solution in an iterative way.

Firstly, a case study was proposed, it addressed the collection of data through a network of sensors. The sensors involved must have a series of characteristics focused on their performance in the system. Thus, this work presents the design of a low-consumption system together with a route optimization system using data obtained by the sensors. The following conclusions were drawn:

- After the study of the different technologies, it is concluded that the use of sensors with a microprocessor of ultra-low energy consumption, together with a wireless network LoRaWAN and following a collaborative scheme is one of the best options to take when building this type of systems is needed.
- Route optimization following a dynamic approach to route calculation versus conventional static route methods provides savings in fuel, time, and workforce.
- The cost of deployment and sensorization of containers remains relatively high but its price is decreasing progressively. This coupled with the use of LPWAN networks significantly reduces the investment in necessary infrastructure, being possible to cover large areas of land only using a very limited number of gateways.
- The use of a collaborative approach in the deployment of such networks favours the development of IoT solutions and can provide an infrastructure for future solutions in both rural and urban areas.

Secondly, thanks to the MOVIURBAN project [MOVIURBAN], it was possible to access historical data from a logistics system such as a bicycle rental system or Bike Sharing System from Salamanca.

It was proposed to perform a case study with these data to evaluate the use of demand prediction techniques in this area for later use the knowledge and conclusions drawn for this case study. In this type of systems, like many other intelligent logistics systems, it is necessary to predict that demand for a particular good will need to collect or deposit and depending on this make an optimization of routes. In this case study, a Multi-Agent system was proposed to perform the tasks of obtaining, processing data from different data sources and generating prediction models. It was concluded that:

- The use of prediction models can be especially suitable for the subsequent optimization of the routes of bicycle rebalancing vehicles in the system.
- The use of a Multi-Agent system can be very useful when modelling a system with different heterogeneous subsystems and very different data sources.

- The development of a visualization tool could be valuable for operators to find patterns in the use of the system by users and their behaviour.

Finally, it was proposed to develop an optimisation system that would take into account the uncertainty present in the VRP systems and make use of the prediction models previously studied in the second case study to improve the resolution of this type of problem. After reviewing the state of the art, it was proposed to employ the Simheuristics methodology (which deals with the resolution of SVRPD problems) and introduce a modification in this methodology. This modification is based on the inclusion of demand prediction models in the simplification step of the problem carried out in this methodology. These are the main conclusions after the work accomplished:

- The Simheuristics proposal offers great advantages over the other resolution methods seen in the literature for the case of stochastic VRP, such as less complexity in the solving algorithms.
- After evaluating the modification proposed for this methodology, it is concluded that it can present improvements compared to the original provided approach, as long as the prediction models perform well.
- If the demand prediction models generated present a good fit and manage to accurately predict the demand for each container, the simplification process will be closer to the deterministic case and therefore better solutions will be obtained than if only the expected value of the probability distribution of demand is used.
- The Simheuristics methodology provides a great decision-making tool for managing the uncertainty associated with waste generation and for evaluating the calculated solution.
- It should be noted that it is necessary to study the modification introduced in this methodology in other domains in which the demand presents a higher variability.

Finally, throughout this research, we have attended workshops and conferences that have fostered the exchange of knowledge with other students and researchers. In addition, the work carried out has been disseminated through the preparation of articles for scientific journals with impact indices, while at the same time international congresses have been attended where lectures and discussions have been held with other researchers.

### **6.5.2 Future lines of work**

This section summarizes the main future lines of research after the work carried out in the present theses.

First of all, for the first case study and research on sensor networks, it is necessary to carry out future studies with a large part of the fleet of containers equipped with sensors to check the evolution of waste generation in the containers. In the same way, it is necessary to carry out coverage tests in isolated places with little line of sight in order to evaluate and validate the theoretical coverage study presented.

Secondly, the second topic of study and relating to prediction models and how to use their results, it is required to carry out case studies involving the relocation of bicycles using the data obtained by the predictive models. In addition, it would be interesting to consider the uncertainty in this type of problem.

Finally, regarding the calculation of routes including uncertainty and the modification of the proposed Simheuristics methodology, it is necessary to carry out studies in other domains in which more information is available that can be used to generate predictive models of demand accurately. It would also be possible to predict different parameters other than the customer demands, that is, including uncertainty in other parameters of the problem such as the arcs between customers (in this case related to traffic) and, in consequence, performing route optimization taking into account these parameters and the predictions made for them.

## Bibliografía

- ABBASI, M. AND EL HANANDEH, A. 2016. Forecasting municipal solid waste generation using artificial intelligence modelling approaches. *Waste Management* 56, 13–22.
- ADAMOVIĆ, V.M., ANTANASIJEVIĆ, D.Z., RISTIĆ, M.Đ., PERIĆ-GRUJIĆ, A.A., AND POCAJT, V. V. 2017. Prediction of municipal solid waste generation using artificial neural network approach enhanced by structural break analysis. *Environmental Science and Pollution Research* 24, 1, 299–311.
- AIOMAS. Aiomas — aiomas 1.0.3 documentation. <http://aiomas.readthedocs.io/en/latest/>.
- ALVAREZ-CAMPANA, M., LÓPEZ, G., VÁZQUEZ, E., VILLAGRÁ, V.A., AND BERROCAL, J. 2017. Smart CEI moncloa: An iot-based platform for people flow and environmental monitoring on a Smart University Campus. *Sensors (Switzerland)* 17, 12, 2856.
- ARCHETTI, C. AND SPERANZA, M.G. 2008. The Split Delivery Vehicle Routing Problem: A Survey. In: *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, Boston, MA, 103–122.
- AREBEY, M., HANNAN, M.A., BASRI, H., ARA BEGUM, R., HANNAN, M., AND BEGUM, R. 2010. Solid waste monitoring system integration based on RFID, GPS and camera  
INVESTIGATION ON THE USE OF MICROGRIDS AND VIRTUAL POWER PLANTS FOR THE INTERGRATION OF RENEWABLE ENERGY SOURCES View project  
Phytoremediation of Lead contaminated water View project  
Radio Frequency Identification (RFID) and communication technologies for solid waste bin and truck monitoring system. .
- AUGUSTIN, A., YI, J., CLAUSEN, T., AND TOWNSLEY, W. 2016a. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors* 16, 12, 1466.
- AUGUSTIN, A., YI, J., CLAUSEN, T., AND TOWNSLEY, W. 2016b. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors* 16, 9, 1466.
- AYOUB, W., MROUE, M., NOUVEL, F., SAMHAT, A.E., AND PREVOTET, J. 2018. Towards IP over LPWANs technologies: LoRaWAN, DASH7, NB-IoT. *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, IEEE, 43–47.
- BAGULA, A., CASTELLI, L., AND ZENNARO, M. 2015. On the Design of Smart Parking Networks in the Smart Cities: An Optimal Sensor Placement Model. *Sensors* 15, 7, 15443–15467.
- BAKAR, S.A.A., ONG, N.R., AZIZ, M.H.A., ALCAIN, J.B., HAIMI, W.M.W.N., AND SAULI, Z. 2017. Underwater detection by using ultrasonic sensor. *AIP Conference Proceedings*,

- AIP Publishing LLC , 020305.
- BARNHART, C., JOHNSON, E.L., NEMHAUSER, G.L., SAVELSBERGH, M.W.P., AND VANCE, P.H. 1998. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research* 46, 3, 316–329.
- BARRIUSO, A.L., GONZÁLEZ, G.V., DE PAZ, J.F., LOZANO, Á., AND BAJO, J. 2018. Combination of multi-agent systems and wireless sensor networks for the monitoring of cattle. *Sensors (Switzerland)* 18, 1.
- BEALE, E.M.L. 1955. On Minimizing A Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society. Series B* 17, 173–184. <http://www.jstor.org/stable/2983952>.
- BEASLEY, J. 1983. Route first—Cluster second methods for vehicle routing. *Omega* 11, 4, 403–408.
- BELL, J.E. AND MCMULLEN, P.R. 2004. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics* 18, 1, 41–48.
- BELLIFEMINE, F., POGGI, A., AND RIMASSA, G. JADE – A FIPA-compliant agent framework. .
- BELLMAN, R. 1958. Dynamic programming and stochastic control processes. *Information and Control* 1, 3, 228–239.
- BERGSTRA, J., BARDENET, R., BENGIO, Y., AND KÉGL, B. Algorithms for Hyper-Parameter Optimization. .
- BERHAN, E., BESHAH, B., KITAW, D., AND ABRAHAM, A. Stochastic Vehicle Routing Problem: A Literature Survey. .
- BERTSIMAS, D. AND SIM, M. 2004. The Price of Robustness. *Operations Research* 52, 1, 35–53.
- BIAU, G. AND SCORNET, E. 2015. A Random Forest Guided Tour. .
- BODIN, L.D. AND BERMAN, L. 1979. Routing and Scheduling of School Buses by Computer. *Transportation Science* 13, 113–129. <https://www.jstor.org/stable/25767942>.
- BOSER, B.E., GUYON, I.M., AND VAPNIK, V.N. 1992. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, ACM Press, 144–152.
- BOUGUERA, T., DIOURIS, J.F., CHAILLOUT, J.J., JAOUADI, R., AND ANDRIEUX, G. 2018. Energy consumption model for sensor nodes based on LoRa and LoRaWAN. *Sensors (Switzerland)* 18, 7.
- BRAEKERS, K., RAMAEKERS, K., AND VAN NIEUWENHUYSE, I. 2016. The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering* 99, 300–313.



- BREIMAN, L. 2001. Random Forests. *Machine Learning* 45, 1, 5–32.
- BREIMAN, L., FRIEDMAN, J.H. (JEROME H.), OLSEN, R.A., AND STONE, C.J. *Classification and regression trees*. .
- BULLNHEIMER, B., HARTL, R.F., AND STRAUSS, C. 1999. Applying the ANT System to the Vehicle Routing Problem. In: *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer US, Boston, MA, 285–296.
- BURATTI, C., CONTI, A., DARDARI, D., AND VERDONE, R. 2009. An overview on wireless sensor networks technology and evolution. *Sensors* 9, 9, 6869–6896.
- CASALS, L., MIR, B., VIDAL, R., AND GOMEZ, C. 2017. Modeling the Energy Performance of LoRaWAN. *Sensors* 17, 10, 2364.
- CATHERWOOD, P.A., MCCOMB, S., LITTLE, M., AND MCLAUGHLIN, J.A.D. Channel Characterisation for Wearable LoRaWAN Monitors. *uir.ulster.ac.uk*, 6–9.
- ČERNÝ, V. 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45, 1, 41–51.
- CHARNES, A. AND COOPER, W.W. 1959. Chance-Constrained Programming. *Management Science* 6, 1, 73–79.
- CHEMLA, D., MEUNIER, F., AND WOLFLER CALVO, R. 2013. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization* 10, 2, 120–146.
- CHEN, S.-T., LIN, S.-S., LAN, C.-W., AND HSU, H.-Y. 2017. Design and Development of a Wearable Device for Heat Stroke Detection. *Sensors* 18, 2, 17.
- CHEONG, P.S., BERGS, J., HAWINKEL, C., AND FAMAHEY, J. 2017. Comparison of LoRaWAN classes and their power consumption. *2017 IEEE Symposium on Communications and Vehicular Technology (SCVT)*, IEEE, 1–6.
- CHOWDHURY, B. AND CHOWDHURY, M.U. 2007. RFID-based real-time smart waste management system. *2007 Australasian Telecommunication Networks and Applications Conference*, IEEE, 175–180.
- CHRISTOFIDES, N. AND BEASLEY, J.E. 1984. The period routing problem. *Networks* 14, 2, 237–256.
- CITI BIKE SYSTEM DATA | CITI BIKE NYC. <https://www.citibikenyc.com/system-data>.
- CLARKE, G. AND WRIGHT, J.W. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* 12, 4, 568–581.
- CLOUDRF. LoRa coverage planning | CloudRF. [https://cloudrf.com/LoRa\\_planning](https://cloudrf.com/LoRa_planning).
- COIN-OR. Vehicle Routing Data Sets. <https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old#V>.

- COMMITTEE, E.C. 2015. ERC Recommendation 70-03. *Short Range Devices*, 1–78.  
<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:ERC+Recommendation+70-03#1>.
- CONTARDO, C., CORDEAU, J.-F., AND GENDRON, B. 2014. An Exact Algorithm Based on Cut-and-Column Generation for the Capacitated Location-Routing Problem. *INFORMS Journal on Computing* 26, 1, 88–102.
- CORTES, C. AND VAPNIK, V. 1995. Support-vector networks. *Machine Learning* 20, 3, 273–297.
- CU, R. 2014. Radio Mobile - RF propagation simulation software - Maplink.txt.  
[http://radiomobile.pe1mew.nl/?The\\_program:File\\_formats:Maplink.txt](http://radiomobile.pe1mew.nl/?The_program:File_formats:Maplink.txt).
- CYBENKO, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2, 4, 303–314.
- DANTZIG, G.B. 1955. Linear Programming under Uncertainty. *Management Science* 1, 3–4, 197–206.
- DANTZIG, G.B. (GEORGE B. 1963. *Linear programming and extensions*. Princeton University Press.
- DANTZIG, G.B. AND RAMSER, J.H. 1959. The Truck Dispatching Problem. *Management Science* 6, 80–91. <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.6.1.80>.
- DELL'AMICO, M., HADJICOSTANTINO, E., IORI, M., AND NOVELLANI, S. 2014. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega* 45, 7–19.
- DIPUTACIÓN DE SALAMANCA. *Servicio De Recogida Selectiva De Residuos Urbanos En Municipios De La Provincia De Salamanca*. . *Tramitación Anticipada (Art. 110,2 Del Trlcspl)*. .
- DORIGO AND M. 1992. Optimization, Learning and Natural Algorithms. *Ph.D. Thesis, Politecnico di Milano, Italy*.
- DROR, M. AND TRUDEAU, P. 1989. Savings by Split Delivery Routing. *Transportation Science* 23, 2, 141–145.
- DRUCKER, H., BURGESS, C.J.C., KAUFMAN, L., SMOLA, A.J., AND VAPNIK, V. 1997. Support Vector Regression Machines. 155–161. <https://papers.nips.cc/paper/1238-support-vector-regression-machines>.
- DUDEK, G. 2011. *Short-Term Load Forecasting using Random Forests*. .
- ECOEMBES. 2008. Recomendaciones para el diseño de un servicio de recogida selectiva de monomaterial de papel y cartón en contenedor. 51.
- EKSIOGLU, B., VURAL, A.V., AND REISMAN, A. 2009. The vehicle routing problem: A

- taxonomic review. *Computers & Industrial Engineering* 57, 4, 1472–1483.
- ERDOĞAN, G., LAPORTE, G., AND WOLFLER CALVO, R. 2014. The static bicycle relocation problem with demand intervals. *European Journal of Operational Research* 238, 2, 451–457.
- ERKIN, D. AND KOOLE, S.G.M. 2017. The Stochastic Vehicle Routing Problem. .
- ESQUERDA, L. Documentation | CityBikes API. <https://api.citybik.es/v2/>.
- FAHMY, H.M.A. 2016. Protocol Stack of WSNs. In: 55–68.
- FISHER, M.L. 1994. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. *Operations Research* 42, 4, 626–642.
- FISHER, M.L. AND JAIKUMAR, R. 1981. A generalized assignment heuristic for vehicle routing. *Networks* 11, 2, 109–124.
- FOSTER, B.A. AND RYAN, D.M. 1976. An Integer Programming Approach to the Vehicle Scheduling Problem. *Operational Research Quarterly (1970-1977)* 27, 2, 367.
- FRANCIS, P.M., SMILOWITZ, K.R., AND TZUR, M. 2008. The Period Vehicle Routing Problem and its Extensions. In: *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, Boston, MA, 73–102.
- FREUND, Y. AND SCHAPIRE, R.E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* 55, 1, 119–139.
- FRIEDMAN, J.H. AND FRIEDMAN, J.H. 2000a. Greedy Function Approximation: A Gradient Boosting Machine. *ANNALS OF STATISTICS* 29, 1189--1232.
- FRIEDMAN, J.H. AND FRIEDMAN, J.H. 2000b. Greedy Function Approximation: A Gradient Boosting Machine. *ANNALS OF STATISTICS* 29, 1189--1232.
- GEURTS, P., ERNST, D., AND WEHENKEL, L. 2006. Extremely randomized trees. *Machine Learning* 63, 1, 3–42.
- GILLETT, B.E. AND MILLER, L.R. 1974. A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research* 22, 2, 340–349.
- GLOVER, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13, 5, 533–549.
- GOBIERNO, D.E.L. AND RODRÍGUEZ, R.R. 2001a. Plan Integral de Residuos de. [http://medioambiente.jcyl.es/web/jcyl/MedioAmbiente/es/Plantilla100/1284312829695/\\_/\\_/](http://medioambiente.jcyl.es/web/jcyl/MedioAmbiente/es/Plantilla100/1284312829695/_/_/).
- GOBIERNO, D.E.L. AND RODRÍGUEZ, R.R. 2001b. Plan Integral de Residuos de. <https://medioambiente.jcyl.es/web/jcyl/MedioAmbiente/es/Plantilla100/1284312829695/>

././.

- GOOGLE. 2019. Google OR-Tools. *Google*, 1.  
<https://developer.linkedin.com/%0Ahttps://developers.google.com/glass/overview%0Ahttps://developers.google.com/>.
- GREGORI, M.E., CÁMARA, J.P., AND BADA, G.A. 2006. A jabber-based multi-agent system platform. *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems - AAMAS '06*, ACM Press, 1282.
- GRIBKOVSKAIA, I., LAPORTE, G., AND SHYSHOU, A. 2008. The single vehicle routing problem with deliveries and selective pickups. *Computers & Operations Research* 35, 9, 2908–2924.
- GRULER, A., JUAN, A.A., FIKAR, C., HIRSCH, P., AND CONTRERAS BOLTON, C. A Simheuristic for the Waste Collection Problem with Stochastic Demands in Smart Cities  
 Simheuristischer Ansatz zur Optimierung der Müllentsorgung mit stochastischer Nachfrage in Smart Cities. .
- GSMA. 2018. Narrowband – Internet of Things (NB-IoT) | Internet of Things. *Gsma*.  
<https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/>.
- GUTIÉRREZ-JARPA, G., DESAULNIERS, G., LAPORTE, G., AND MARIANOV, V. 2010. Production, Manufacturing and Logistics A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows. .
- GUTIERREZ, J.M., JENSEN, M., HENIUS, M., AND RIAZ, T. 2015. Smart Waste Collection System Based on Location Intelligence. *Procedia Computer Science* 61, 120–127.
- HANNAN, M.A., ABDULLA AL MAMUN, M., HUSSAIN, A., BASRI, H., AND BEGUM, R.A. 2015. A review on technologies and their usage in solid waste monitoring and management systems: Issues and challenges. *Waste Management* 43, 509–523.
- HASHIMOTO, H., YAGIURA, M., IMAHORI, S., AND IBARAKI, T. 2010. Recent progress of local search in handling the time window constraints of the vehicle routing problem. *4OR* 8, 3, 221–238.
- HAYSTACK. Haystack Technologies: Turbocharge Your IoT Mission.  
<http://haystacktechnologies.com/>.
- HERRERA, M., TORGO, L., IZQUIERDO, J., AND PÉREZ-GARCÍA, R. 2010. Predictive models for forecasting hourly urban water demand. *Journal of Hydrology* 387, 1–2, 141–150.
- HOLLAND, J.H. (JOHN H. 1975. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press.
- HOPENGARTEN, F. 2009. *Antenna Zoning: Broadcast, Cellular & Mobile Radio, Wireless Internet: Laws, Permits & Leases*. Focal Press.

- JOHN LU, Z.Q. 2010. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 173, 3, 693–694.
- JOHNSON, N.E., IANIUK, O., CAZAP, D., ET AL. 2017. Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York City. *Waste Management* 62, 3–11.
- JOSHI, R., GUPTA, R., AND SARAVANAN, P. 2018. A Random Forest Approach for Predicting Online Buying Behavior of Indian Customers. *Theoretical Economics Letters* 08, 03, 448–475.
- JUAN, A., FAULIN, J., GRASMAN, S., RIERA, D., MARULL, J., AND MENDEZ, C. 2011. Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. *Transportation Research Part C: Emerging Technologies* 19, 5, 751–765.
- JUAN, A.A., FAULIN, J., GRASMAN, S.E., RABE, M., AND FIGUEIRA, G. 2015a. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives* 2, 62–72.
- JUAN, A.A., FAULIN, J., GRASMAN, S.E., RABE, M., AND FIGUEIRA, G. 2015b. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives* 2, 62–72.
- KAGGLE INC. 2014. Bike Sharing Demand | Kaggle. <https://www.kaggle.com/c/bike-sharing-demand>.
- KANE, M.J., PRICE, N., SCOTCH, M., AND RABINOWITZ, P. 2014. Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinformatics* 15, 1, 276.
- KANNANGARA, M., DUA, R., AHMADI, L., AND BENSEBAA, F. 2018. Modeling and prediction of regional municipal solid waste generation and diversion in Canada using machine learning approaches. *Waste Management* 74, 3–15.
- KARADIMAS, D., PAPALAMBROU, A., GIALELIS, J., AND KOUBIAS, S. 2016. An integrated node for Smart-City applications based on active RFID tags; Use case on waste-bins. *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 1–7.
- KASAMPALIS, S., LAZARIDIS, P.I., ZAHARIS, Z.D., BIZOPOULOS, A., ZETTAS, S., AND COSMAS, J. 2013. Comparison of Longley-Rice, ITM and ITWOM propagation models for DTV and FM broadcasting. *undefined*.
- KIRKPATRICK, S., GELATT, ; C D, AND VECCHI, ; M P. 1983. *Optimization by Simulated Annealing*. .
- KOLMOGOROV, A.N. 1957. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady*

- Akademii Nauk*, 953–956.
- KUMAR, M. AND THENMOZHI, M. 2014. Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. *International Journal of Banking, Accounting and Finance* 5, 3, 284.
- LENSTRA, J.K. AND KAN, A.H.G.R. 1981. Complexity of vehicle routing and scheduling problems. *Networks* 11, 2, 221–227.
- LIBELIUM. Libelium - Connecting Sensors to the Cloud. <http://www.libelium.com/>.
- LIBERATORE, F., RIGHINI, G., AND SALANI, M. 2011. A column generation algorithm for the vehicle routing problem with soft time windows. *4OR* 9, 1, 49–82.
- LIN, J.H. AND CHOU, T.C. 2012. A Geo-Aware and VRP-Based Public Bicycle Redistribution System. *International Journal of Vehicular Technology* 2012, 1–14.
- LIN, S. 1965. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal* 44, 10, 2245–2269.
- LIN, S. AND KERNIGHAN, B.W. 1973. An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research* 21, 2, 498–516.
- LLUÍS ESQUERDA. CityBikes: bike sharing networks around the world. <https://citybik.es/>.
- LONGHI, S., MARZIONI, D., ALIDORI, E., ET AL. 2012. Solid Waste Management Architecture Using Wireless Sensor Network Technology. *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, IEEE, 1–5.
- LORA ALLIANCE. 2017. Lora-Alliance Technology. <https://www.lora-alliance.org/technology>.
- LORASERVERIO. 2018. LoRa Server, open-source LoRaWAN network-server. <https://www.loraserver.io/>.
- LORIENT. LORIENT | LoRaWAN Services and Software. <https://www.loriot.io/>.
- LUNDH, O. 2017. Communication protocols for mid-range mobile IoT devices : And their applicability to a publicly shared bikes platform. .
- MAGGIONI, F., POTRA, F.A., AND BERTOCCHI, M. Stochastic versus Robust Optimization for a Transportation Problem. .
- MALANI, J., SINHA, N., PRASAD, N., AND LOKESH, V. Forecasting Bike Sharing Demand. .
- MICROCHIP TECHNOLOGY INC. 868MHz RN2483 LoRa(TM) Technology Mote - dm164138 | Microchip Technology Inc. [http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=dm164138&utm\\_source=&utm\\_medium=MicroSolutions&utm\\_term=&utm\\_content=DevTools&utm\\_campaign=RN2483+LoRa+Mote](http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=dm164138&utm_source=&utm_medium=MicroSolutions&utm_term=&utm_content=DevTools&utm_campaign=RN2483+LoRa+Mote).

- MOHAMED, S.M., HAMZA, H.S., AND SAROIT, I.A. 2017. Coverage in mobile wireless sensor networks (M-WSN): A survey. *Computer Communications* 110, 133–150.
- MORA, H., GILART-IGLESIAS, V., PÉREZ-DEL HOYO, R., AND ANDÚJAR-MONTOYA, M.D. 2017. A Comprehensive System for Monitoring Urban Accessibility in Smart Cities. *Sensors (Basel, Switzerland)* 17, 8.
- MOVIURBAN. MOVIURBAN. <https://moviurban.usal.es/>.
- MUSAT, G.A., COLEZEA, M., POP, F., ET AL. 2018. Advanced services for efficient management of smart farms. *Journal of Parallel and Distributed Computing* 116, 3–17.
- NASSIF, A.B. 2016. Short term power demand prediction using stochastic gradient boosting. *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, IEEE, 1–4.
- NEWTON, R.M. AND THOMAS, W.H. 1974. Bus routing in a multi-school system. *Computers & Operations Research* 1, 2, 213–222.
- NGUYEN, T.-T., NGUYEN HUU, Q., AND JUNJIE LI, M. 2015. Forecasting time series water levels on Mekong river using machine learning models. .
- NICE RF. LoRa1276-C1. [http://www.nicerf.com/Upload/ueditor/files/2017-06-21/LORA1276-C1\\_100mW\\_long\\_range\\_Spread\\_Spectrum\\_modulation\\_wireless\\_transceiver\\_module\\_V1.1-ac9630bd-6925-4c75-9149-db1a18d00426.pdf](http://www.nicerf.com/Upload/ueditor/files/2017-06-21/LORA1276-C1_100mW_long_range_Spread_Spectrum_modulation_wireless_transceiver_module_V1.1-ac9630bd-6925-4c75-9149-db1a18d00426.pdf).
- NOBERT, Y. 1987. Exact Algorithms for the Vehicle Routing Problem. *North-Holland Mathematics Studies* 132, 147–184.
- OSBRAIN. osBrain - 0.5.0 — osBrain 0.5.0 documentation. <http://osbrain.readthedocs.io/en/stable/>.
- OSMAN, I.H. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41, 4, 421–451.
- PADBERG, M. AND RINALDI, G. A Branch-and-Cut Approach to a Traveling Salesman Problem with Side Constraints. *Management Science* 35, 1393–1412. <https://www.jstor.org/stable/2632284>.
- PADBERG, M. AND RINALDI, G. 1991. A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Review* 33, 1, 60–100.
- PAL, A. AND ZHANG, Y. 2017. Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. *Transportation Research Part C: Emerging Technologies* 80, 92–116.
- PARK, S., YUN, S., KIM, H., KWON, R., ... J.G.-P. OF THE 8TH, AND 2018, UNDEFINED. Forestry Monitoring System using LoRa and Drone. *dl.acm.org*.
- PARRAGH, S.N., DOERNER, K.F., AND HARTL, R.F. 2008. A survey on pickup and delivery

- problems. *Journal für Betriebswirtschaft* 58, 1, 21–51.
- DE PAZ, J.F., BAJO, J., RODRÍGUEZ, S., VILLARRUBIA, G., AND CORCHADO, J.M. 2016. Intelligent system for lighting control in smart cities. *Information Sciences* 372, 241–255.
- PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., ET AL. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, Oct, 2825–2830.
- PEDREGOSA FABIANPEDREGOSA, F., ALEXANDRE GRAMFORT, N., MICHEL, V., ET AL. 2011. Scikit-learn: Machine Learning in Python Gaël Varoquaux. *Journal of Machine Learning Research* 12, 2825–2830.
- PETÄJÄJÄRVI, J., MIKHAYLOV, K., PETTISSALO, M., JANHUNEN, J., AND IINATTI, J. 2017. Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. *International Journal of Distributed Sensor Networks* 13, 3, 155014771769941.
- PETÄJÄJÄRVI, J., MIKHAYLOV, K., YASMIN, R., HÄMÄLÄINEN, M., AND IINATTI, J. *Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring*. .
- POWELL, W.B. AND TOPALOGLU, H. 2003. Stochastic Programming in Transportation and Logistics. *Handbooks in Operations Research and Management Science* 10, 555–635.
- PRAKASH NEKKANTI, O. 2017. PREDICTION OF RENTAL DEMAND FOR A BIKE-SHARE PROGRAM. .
- PULE, M., YAHYA, A., AND CHUMA, J. 2017a. Wireless sensor networks: A survey on monitoring water quality. *Journal of Applied Research and Technology* 15, 6, 562–570.
- PULE, M., YAHYA, A., AND CHUMA, J. 2017b. Wireless sensor networks: A survey on monitoring water quality. *Journal of Applied Research and Technology* 15, 6, 562–570.
- RAMSON, S.R.J. AND MONI, D.J. 2017a. Wireless sensor networks based smart bin. *Computers and Electrical Engineering* 64, 337–353.
- RAMSON, S.R.J. AND MONI, D.J. 2017b. Wireless sensor networks based smart bin. *Computers & Electrical Engineering* 64, 337–353.
- REASON, P. AND BRADBURY, H. 2008. *The SAGE handbook of action research : participative inquiry and practice*. SAGE.
- REGUE, R. AND RECKER, W. 2014. Using Gradient Boosting Machines to Predict Bikesharing Station States. .
- ROMAN, R., LOPEZ, J., AND MAMBO, M. 2018. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems* 78, 680–698.
- ROSSI, F., VAN BEEK, P., AND WALSH, T. 2006. *Handbook of constraint programming*.



Elsevier.

- SALZBERG, S.L. 1994. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* 16, 3, 235–240.
- SAMMY-L21. Sammy-L21 Infosheet OEM module with Atmel/Microchip ATSAML21 processor. .
- SEMTECH. 2018. LoRa | LoRa Wireless Technology | Semtech | Semtech. <https://www.semtech.com/technology/lora>.
- SENET. 2018. Senet | Internet of Things Network for IoT Devices. <https://www.senetco.com/>.
- SHAHABI, H., KHEZRI, S., AHMAD, B. BIN, AND ZABIHI, H. 2012. Application of Artificial Neural Network in Prediction of Municipal Solid Waste Generation (Case Study: Saqez City in Kurdistan Province). *World Applied Sciences Journal* 20, 2, 336–343.
- SIGFOX. 2016. Sigfox - The Global Communications Service Provider for the Internet of Things (IoT). <https://www.sigfox.com/en%0Ahttp://www.sigfox.com/en/>.
- SIGFOX. 2017. Sigfox Spain | Sigfox Partner Network. <https://partners.sigfox.com/companies/sigfox-spain>.
- SILVA, B.N., KHAN, M., AND HAN, K. 2018. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society* 38, 697–713.
- SINHA, R.S., WEI, Y., AND HWANG, S.H. 2017. A survey on LPWA technology: LoRa and NB-IoT. *ICT Express* 3, 1, 14–21.
- SMOLA, A.J., SCHÖLKOPF, B., AND SCHÖLKOPF, S. 2004. *A tutorial on support vector regression* \*. Kluwer Academic Publishers.
- SOLANO-CHARRIS, E.L. 2015. Optimization Methods for the Robust Vehicle Routing Problem. .
- SOLOMON, M.M. 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research* 35, 2, 254–265.
- SUJUAN LIU, CHUYU XIA, AND ZHENZHEN ZHAO. 2016. A low-power real-time air quality monitoring system using LPWAN based on LoRa. *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, IEEE, 379–381.
- SYMPHONY LINK. Symphony Link - Internet of Things Wireless LPWA. <https://www.link-labs.com/symphony>.
- TAILLARD, É., BADEAU, P., GENDREAU, M., GUERTIN, F., AND POTVIN, J.-Y. 1997. A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science* 31, 2, 170–186.

- THE THINGS NETWORK. 2016. The Things Network. <https://www.thethingsnetwork.org/>.
- TING, C.-J. AND CHEN, C.-H. 2013. A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics* 141, 1, 34–44.
- TOTH, P. AND VIGO, D. 2002. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics.
- TOTH, P. AND VIGO, D. 2003. The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing* 15, 4, 333–346.
- ULPMARK. EEMBC - ULPMark - Low Power Benchmark. <https://www.eembc.org/ulpmark/index.php>.
- UPLMARK. EEMBC - ULPMark - Low Power Benchmark. <https://www.eembc.org/ulpmark/index.php>.
- VATCHARATIANSAKUL, N., TUWANUT, P., AND PORNAVALAI, C. 2017. Experimental performance evaluation of LoRaWAN: A case study in Bangkok. *Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017*.
- VERIKAS, A., GELZINIS, A., AND BACAUSKIENE, M. 2011. Mining data with random forests: A survey and results of new tests. *Pattern Recognition* 44, 2, 330–349.
- VIGO, D. AND TOTH, P. 2014. *Vehicle Routing; Problems, Methods, and Applications*. .
- VILLARRUBIA, G., PAZ, J.F. DE, IGLESIA, D.H.D. LA, AND BAJO, J. 2017a. Combining Multi-Agent Systems and Wireless Sensor Networks for Monitoring Crop Irrigation. *Sensors* 17, 8, 1775.
- VILLARRUBIA, G., DE PAZ, J.F., DE LA IGLESIA, D.H., AND BAJO, J. 2017b. Combining multi-agent systems and wireless sensor networks for monitoring crop irrigation. *Sensors (Switzerland)* 17, 8, 1775.
- VINAGRE, E., DE PAZ, J.F., PINTO, T., VALE, Z., CORCHADO, J.M., AND GARCIA, O. 2016. Intelligent energy forecasting based on the correlation between solar radiation and consumption patterns. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 1–7.
- WALKER, H.R. 2004. Ultra narrow band modulation. *2004 IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communications*.
- WANG, Q. AND BALASINGHAM, I. 2010. Wireless Sensor Networks - An Introduction. In: *Wireless Sensor Networks: Application-Centric Design*. InTech.
- WANG, W., WANG, W., AND CURLEY, A. 2016. Forecasting Bike Rental Demand Using New York Citi Bike Data Forecasting bike rental demand using New York Citi Bike data. .

- WANG, Y.P.E., LIN, X., ADHIKARY, A., ET AL. 2017. A Primer on 3GPP Narrowband Internet of Things. *IEEE Communications Magazine* 55, 3, 117–123.
- WEIGHTLESS. 2017. Weightless - Setting the Standard for IoT. <http://www.weightless.org/>.
- WREN, A. AND HOLLIDAY, A. 1972. Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. *Journal of the Operational Research Society* 23, 3, 333–344.
- YANG, S.-H. 2014. Internet of Things. *Bus iNF Syst Eng* 57(3), 247–261. [http://link.springer.com/10.1007/978-1-4471-5505-8\\_12](http://link.springer.com/10.1007/978-1-4471-5505-8_12).
- YU, Y.-C., WANG, Y.-W., LIAO, M.-S., JIANG, J.-A., AND LEE, Y.-C. 2017. A Long Range Wide Area Network-Based Smart Pest Monitoring System. *World Academy of Science, Engineering and Technology International Journal of Agricultural and Biosystems Engineering International Science Index, Agricultural and Biosystems Engineering International Scholarly and Scientific Research & Innovation* 1111, 115, 2017–1.
- ZATO, C., VILLARRUBIA, G., SÁNCHEZ, A., ET AL. 2012. PANGEA – Platform for Automatic coNstruction of orGanizations of intElligent Agents. In: Springer, Berlin, Heidelberg, 229–239.
- ZHANG, Y. AND HAGHANI, A. 2015. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies* 58, 308–324.
- ZIMMERMANN, H. 1980. OSI Reference Model--The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications* 28, 4, 425–432.



## Glosario

***Bike Sharing System:*** servicio en el que bicicletas son puestas a disposición de los usuarios a lo largo de una ciudad para su uso en periodos cortos de tiempo, habitualmente recorridos ente un origen y un destino que pueden ser estaciones para depositar dichas bicicletas. Habitualmente tienen un modelo de negocio de pago por uso o suscripción.

***Customer:*** en el ámbito de los VRP hace referencia a los nodos que deben visitar los vehículos en sus rutas. Presentan una demanda de un bien o un servicio que debe ser atendido y pueden presentar un rango temporal en el que pueden estar disponibles.

***Decision maker:*** en la teoría de la decisión, es la persona encargada de tomar las decisiones que afectan a la resolución de un problema o la aplicación de una orden, intentando conseguir el mejor resultado con su decisión.

***Depot:*** en el ámbito de los VRP, el *depot* es el nodo del que parten y donde acaban los vehículos que realizarán las rutas en el problema.

***Downlink:*** en la especificación LoRaWAN hace referencia a los mensajes enviados desde un servidor de aplicación hasta un nodo.

***Duty Cycle:*** en el ámbito de LoRa hace referencia al tiempo en el que un nodo está transmitiendo respecto a la fracción tota de tiempo. Si un nodo transmite durante 1 unidad de tiempo cada 100 unidades de tiempo, presenta un *duty cycle* de 1%.

***Ensemble:*** en aprendizaje automático, los métodos combinados o de *ensemble*, emplean múltiples algoritmos de aprendizaje para obtener un rendimiento mayor que el rendimiento que podría obtenerse con alguno de los algoritmos que lo componen por separado en su tarea de clasificación o regresión.

***Gateway:*** en una arquitectura de red LoRaWAN hace referencia al elemento que sirve como nexa entre los nodos de la red y el servidor de red. Es el encargado de retransmitir los paquetes que le llegan de los nodos al servidor de red y del servidor de red a los nodos.

***Internet of Things:*** concepto que hace referencia a la interconexión digital de cualquier elemento de la vida cotidiana con Internet.

**LPWAN:** (*Low Power Wide Area Networks*) Son un tipo de redes orientadas principalmente a dispositivos de u bajo consumo energético, con una baja transmisión de datos y un largo alcance.

**Payload:** en telecomunicaciones hace referencia a la carga útil de un mensaje que es transmitido por una red. Esta carga útil excluye cabeceras o metadatos incluidos en el mensaje transmitido.

**Recourse action:** en problemas VRPSD hace referencia a la acción que se toma cuando la demanda encontrada en un *customer* sobrepasa la capacidad del vehículo que lo visita.

**Regressor:** en aprendizaje automático hace referencia a un tipo de estimador cuya salida es un valor real, es decir, trata de resolver un problema de regresión.

**Simheuristics:** metodología para la resolución de problemas VRPSD que combina técnicas de simulación con técnicas de resolución de VRP deterministas.

**Smart City:** ciudad inteligente en castellano, es un concepto en auge que hace referencia a un tipo de desarrollo urbano basado en la sostenibilidad y en la eficiencia e inteligencia de los servicios que provee a los ciudadanos e instituciones.

**Smart Farming:** representa la integración de las tecnologías de la información y telecomunicación TIC la agricultura y ganadería además de la dotación de aspectos inteligentes en los procesos involucrados en estas áreas.

**Solver:** pieza de software matemático que resuelve un determinado problema matemático. Los solvers toman la descripción de un problema de forma genérica y calculan mediante un algoritmo su solución.

**Uplink:** en la especificación LoRaWAN hace referencia a los mensajes enviados desde un nodo hasta un servidor de aplicación.

**Variable estocástica:** o aleatoria, hace referencia a una variable cuyo valor no es determinista y depende de una distribución de probabilidad.

**Vehicle Routing Problem:** problema de optimización combinatoria y de programación entera que trata de buscar el conjunto óptimo de rutas para una flota de vehículos que debe satisfacer las demandas de un conjunto de clientes.

**Wireless Sensor Network:** o redes inalámbricas de sensores, hace referencia a un grupo de sensores diseminados para monitorizar o actuar sobre las condiciones físicas de un entorno y que presentan un sistema de comunicación inalámbrico en la transmisión de la información.

## Siglas y acrónimos

<b>API</b>	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface
<b>BRP</b>	<b>B</b> ike <b>R</b> ebalancing <b>P</b> roblem
<b>BSS</b>	<b>B</b> ike <b>S</b> haring <b>S</b> ystems
<b>CART</b>	<b>C</b> lassification and <b>R</b> egression <b>T</b> rees
<b>CCL</b>	<b>C</b> ustomized <b>L</b> ogic <b>O</b> perations
<b>CP</b>	<b>C</b> onstraint <b>P</b> rogramming
<b>CSS</b>	<b>C</b> hirp <b>S</b> pread <b>S</b> pectrum
<b>CSV</b>	<b>C</b> omma <b>S</b> eparated <b>V</b> alues
<b>CV</b>	<b>C</b> ross <b>V</b> alidation
<b>CVRP</b>	<b>C</b> apacitated <b>V</b> ehicle <b>R</b> outing <b>P</b> roblem
<b>GIS</b>	<b>G</b> eographic <b>I</b> nformation <b>S</b> ystem
<b>GPRS</b>	<b>G</b> eneral <b>P</b> acket <b>R</b> adio <b>S</b> ervice
<b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning <b>S</b> ystem
<b>HFVRP</b>	<b>H</b> eterogeneous <b>F</b> leet <b>V</b> RP
<b>ILP</b>	<b>I</b> nteger <b>L</b> inear <b>P</b> rogramming
<b>IOT</b>	<b>I</b> nternet of <b>T</b> hings
<b>ISM</b>	<b>I</b> ndustrial, <b>S</b> cientific and <b>M</b> edical
<b>ISP</b>	<b>I</b> nternet <b>S</b> ervice <b>P</b> rovider
<b>ITM</b>	<b>I</b> rregular <b>T</b> errain <b>M</b> odel
<b>LPWAN</b>	<b>L</b> ow- <b>P</b> ower <b>W</b> ide- <b>A</b> rea <b>N</b> etwork
<b>LTE</b>	<b>L</b> ong <b>T</b> erm <b>E</b> volution
<b>MAC</b>	<b>M</b> edia <b>A</b> ccess <b>C</b> ontrol

---

<b>MAS</b>	<b>Multi Agent System</b>
<b>MCU</b>	<b>Microcontroller Unit</b>
<b>MDVRP</b>	<b>Multiple Depot VRP</b>
<b>MQTT</b>	<b>Message Queuing Telemetry Transport</b>
<b>OSI</b>	<b>Open System Interconnection</b>
<b>PVRP</b>	<b>Periodic VRP</b>
<b>QoS</b>	<b>Quality of Service</b>
<b>REST</b>	<b>Representational State Transfer</b>
<b>RFID</b>	<b>Radio Frequency Identification</b>
<b>RPMA</b>	<b>Random Phase Multiple Access</b>
<b>SDVRP</b>	<b>Split Delivery VRP</b>
<b>SIG</b>	<b>Special Interest Group</b>
<b>SRTM</b>	<b>Shuttle Radar Topography Mission</b>
<b>STSP</b>	<b>Symmetric Travel Salesman Problem</b>
<b>SVM</b>	<b>Support Vector Machine</b>
<b>SVR</b>	<b>Support Vector Regression</b>
<b>SVRP</b>	<b>Stochastic VRP</b>
<b>TSP</b>	<b>Travel Salesman Problem</b>
<b>TTN</b>	<b>The Thing Network</b>
<b>UNB</b>	<b>Ultra Narrow Band</b>
<b>VRP</b>	<b>Vehicle Routing Problem</b>
<b>VRPB</b>	<b>VRP with Backhauls</b>
<b>VRPPD</b>	<b>VRP with Pick-Up and Delivering</b>
<b>VRPSC</b>	<b>VRP Stochastic Clients</b>



<b>VRPSD</b>	<b>VRP Stochastic Demands</b>
<b>VRPSTT</b>	<b>VRP Stochastic Travel Times</b>
<b>VRPTW</b>	<b>VRP with Time Windows</b>
<b>WAN</b>	<b>Wide Area Network</b>
<b>WPAN</b>	<b>Wireless Personal Area Network</b>
<b>WSN</b>	<b>Wireless Sensor Network</b>



---

# ANEXO I

---



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

---

## Publicaciones y trabajos relacionados

---



## ANEXO I: Publicaciones y trabajos relacionados

Artículos en revistas y congresos internacionales publicados durante el trabajo de tesis.

### ARTÍCULOS REVISTA PUBLICADOS DURANTE EL TRABAJO DE TESIS

<b>Título</b>	<b>Smart Waste Collection System with Low Consumption LoRaWAN Nodes and Route Optimization</b>
<b>Factor de impacto JCR</b>	2.475 Q2 (2017)
<b>Autores</b>	<u>Álvaro Lozano</u> Javier Caridad, Juan Francisco De Paz, Gabriel Villarrubia González y Javier Bajo
<b>DOI</b>	<a href="https://doi.org/10.3390/s18051465">https://doi.org/10.3390/s18051465</a>
<b>Revista</b>	Sensors (Basel). Volumen 18, Issue 5, May 2018

<b>Título</b>	<b>Multi-Agent System for Demand Prediction and Trip Visualization in Bike Sharing Systems</b>
<b>Factor de impacto JCR</b>	1.679 Q3 (2017)
<b>Autores</b>	<u>Álvaro Lozano</u> , Juan F. De Paz, Gabriel Villarrubia González, Daniel H. De La Iglesia y Javier Bajo
<b>DOI</b>	<a href="https://doi.org/10.3390/app8010067">https://doi.org/10.3390/app8010067</a>
<b>Revista</b>	Applied Sciences. Volumen 8, January Issue 1

<b>Título</b>	<b>MOVICLOUD: Agent-Based 3D Platform for the Labor Integration of Disabled People</b>
<b>Factor de impacto JCR</b>	1.679 Q3 (2017)
<b>Autores</b>	Alberto L. Barriuso, Fernando De la Prieta, Gabriel Villarrubia González, Daniel H. De La Iglesia y <b><u>Álvaro Lozano</u></b>
<b>DOI</b>	<a href="https://doi.org/10.3390/app8030337">https://doi.org/10.3390/app8030337</a>
<b>Revista</b>	Applied Sciences. Volumen 8, Issue 3 February 2018

<b>Título</b>	<b>Combination of Multi-Agent Systems and Wireless Sensor Networks for the Monitoring of Cattle</b>
<b>Factor de impacto JCR</b>	2.475 Q2 (2017)
<b>Autores</b>	Alberto L. Barriuso, Gabriel Villarrubia González, Juan F. De Paz, <b><u>Álvaro Lozano</u></b> and Javier Bajo
<b>DOI</b>	<a href="https://doi.org/10.3390/s18010108">https://doi.org/10.3390/s18010108</a>
<b>Revista</b>	Sensors (Basel). Volumen 18, Issue 1, January 2018

**COMUNICACIONES A CONGRESO REALIZADAS DURANTE EL DESARROLLO DE TRABAJO DE TESIS**

<b>Título</b>	<b>Smart Waste Collection Platform Based on WSN and Route Optimization</b>
<b>Autores</b>	<u>Álvaro Lozano Murciego</u> , Gabriel Villarrubia-González, Alberto López Barriuso, Daniel Hernández de La Iglesia, Jorge Revuelta Herrero, Juan F. De Paz Santana
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-40159-1_11">https://doi.org/10.1007/978-3-319-40159-1_11</a>
<b>Publicación</b>	Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection
<b>Lugar de Celebración</b>	Sevilla, España. 01-03/06/2016

<b>Título</b>	<b>New Architecture for Electric Bikes Control Based on Smartphones and Wireless Sensors</b>
<b>Autores</b>	Jorge Revuelta Herrero, Gabriel Villarrubia, Alberto López Barriuso, Daniel Hernández de La Iglesia, <u>Álvaro Lozano Murciego</u> , Marco Antonio de la Serna González
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-40159-1_10">https://doi.org/10.1007/978-3-319-40159-1_10</a>

<b>Publicación</b>	Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection
<b>Lugar de Celebración</b>	Sevilla, España. 01-03/06/2016

<b>Título</b>	<b>Multi Agent Application for Chronic Patients: Monitoring and Detection of Remote Anomalous Situations</b>
<b>Autores</b>	Daniel Hernández de La Iglesia, Gabriel Villarrubia, Alberto L. Barriuso, <u>Álvaro Lozano Murciego</u> , Jorge Revuelta Herrero, Juan Francisco de Paz
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems 2016
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-39387-2_3">https://doi.org/10.1007/978-3-319-39387-2_3</a>
<b>Publicación</b>	PAAMS 2016: Highlights of Practical Applications of Scalable Multi-Agent Systems. The PAAMS Collection
<b>Lugar de Celebración</b>	Sevilla, España. 01-03/06/2016

<b>Título</b>	<b>An Intelligent Agent-Based Journalism Platform</b>
<b>Autores</b>	Alberto L. Barriuso, Fernando de la Prieta, <u>Álvaro Lozano Murciego</u> , Daniel Hernández de La Iglesia, Jorge Revuelta Herrero
<b>Congreso / Evento</b>	



	International Conference on Practical Applications of Agents and Multi-Agent Systems 2016
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-39387-2_27">https://doi.org/10.1007/978-3-319-39387-2_27</a>
<b>Publicación</b>	PAAMS 2016: Highlights of Practical Applications of Scalable Multi-Agent Systems. The PAAMS Collection
<b>Lugar de Celebración</b>	Sevilla, España. 01-03/06/2016

<b>Título</b>	<b>Single appliance automatic recognition: comparison of classifiers</b>
<b>Autores</b>	Daniel Hernández de la Iglesia, Alberto López Barriuso, <u>Álvaro Lozano Murciego</u> , Jorge Revuelta Herrero, Jorge Landeck, Juan F. de Paz, Juan M. Corchado
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-61578-3_11">https://doi.org/10.1007/978-3-319-61578-3_11</a>
<b>Publicación</b>	PAAMS 2017: Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference
<b>Lugar de Celebración</b>	Porto (Portugal), 21-23/06/2017

<b>Título</b>	<b>Non-Intrusive Load Monitoring (NILM): a state of the art</b>
<b>Autores</b>	Jorge Revuelta Herrero, <u>Álvaro Lozano Murciego</u> , Alberto López Barriuso, Daniel Hernández de la Iglesia, Gabriel Villarrubia González, Juan Manuel Corchado Rodríguez, Rita Carreira
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-61578-3_12">https://doi.org/10.1007/978-3-319-61578-3_12</a>
<b>Publicación</b>	PAAMS 2017: Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference
<b>Lugar de Celebración</b>	Porto (Portugal), 21-23/06/2017

<b>Título</b>	<b>Household Occupancy Detection Based on Electric Energy Consumption</b>
<b>Autores</b>	Alberto L Barriuso, <u>Álvaro Lozano</u> , H Daniel, Gabriel Villarrubia, Juan F de Paz
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-94779-2_20">https://doi.org/10.1007/978-3-319-94779-2_20</a>
<b>Publicación</b>	

	PAAMS 2018: Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection
<b>Lugar de Celebración</b>	Toledo (España), 21-23/06/2018

<b>Título</b>	<b>Multi-agent System for the Recommendation of Electric Bicycle Routes</b>
<b>Autores</b>	H Daniel, Álvaro Lozano Murciego, Alberto L Barriuso, Gabriel Villarrubia, Juan F de Paz
<b>Congreso / Evento</b>	International Conference on Practical Applications of Agents and Multi-Agent Systems
<b>DOI</b>	<a href="https://doi.org/10.1007/978-3-319-94779-2_4">https://doi.org/10.1007/978-3-319-94779-2_4</a>
<b>Publicación</b>	PAAMS 2018: Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection
<b>Lugar de Celebración</b>	Toledo (España), 21-23/06/2018