



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Máster en Ingeniería de Computadores y Redes

Uso de Vehículos como Sensores para Estimar la Intensidad de Precipitación

Autor: Karin Natalia Cicienia Cárdenas

Director: Carlos Tavares Calafate

Valencia - España

Diciembre de 2015

Índice

1. Introducción	3
2. Uso de VANETs para Monitorización Ambiental	5
3. Generación de un Patrón de Precipitación Sintético	7
3.1. Umbrales de Precipitación	8
3.2. Ajuste Gaussiano de los datos	10
3.3. Masa de Precipitación Generada	11
4. Framework de Simulación	15
4.1. Simulation for Urban MObility –SUMO–	16
4.1.1. Movilidad vehicular usando SUMO	16
4.2. OMNeT++	19
4.2.1. Estructura Modular de la Simulación	19
4.3. Integración de datos de sensorización en OMNeT++	20
4.3.1. Precipitation.ned	21
4.3.2. Car.Ned	22
4.3.3. Parámetros de simulación relevantes (<i>omnetpp.ini</i>)	23
5. Simulación y Análisis de Resultados	24
5.1. Lecturas de intensidad de lluvia	25
5.2. Predicción del patrón de precipitación bajo condiciones ideales	27
5.3. Predicción del patrón de precipitación asumiendo errores de muestreo (lectura de intensidad de lluvia)	30
6. Conclusiones	33

1. Introducción

Las redes vehiculares ad-hoc (VANETs) son un tipo especial de redes ad-hoc móviles (MANETs) adaptadas a las comunicaciones entre vehículos. Hoy en día, los vehículos están equipados con dispositivos de comunicación inalámbrica de corto alcance y, por lo general, tienen dispositivos móviles embebidos, en forma de ordenadores a bordo o navegadores GPS, que les permiten compartir diversa información entre ellos. Esta información puede ser sobre asistencia al conductor, seguridad del coche, precios de combustible o información sobre las condiciones de las vías de tráfico. Como resultado, se pueden implementar aplicaciones interesantes en base a la cooperación entre los vehículos [9].

Un requisito esencial en el equipamiento del vehículo es el limpiaparabrisas o limpiacristales; este dispositivo que se utiliza para remover la lluvia y la suciedad de un parabrisas. Actualmente, se ha integrado en los vehículos limpiaparabrisas inteligentes mediante un controlador programable (automático) que detecta la presencia y la cantidad de lluvia utilizando un sensor de precipitación. El sensor ajusta automáticamente la velocidad de los limpiaparabrisas de acuerdo con la cantidad de lluvia detectada [7].

En el presente trabajo, explotando las características y facilidades de los vehículos modernos y la información que podemos obtener de ellos, se propone un estudio basado en simulación para estimar la intensidad de precipitación de lluvia en un área específica de la ciudad de Valencia. Para ello, cada vehículo es considerado como un sensor de lluvia, y la estimación se realiza en base a la información obtenida, de preferencia en tiempo real, con los diferentes vehículos que forman una VANET.

Para el análisis se consideraron varios escenarios, variando parámetros en la simulación como: el número de vehículos en la red, el tiempo total de muestreo, el tiempo entre muestras para cada coche, y el error en la muestra. En los escenarios estudiados se consideraron dos opciones: obtener información directamente del sensor de lluvia, y utilizar la velocidad del limpiacristales, siendo esta última técnica modelada mediante la introducción de un elevado grado de error en la estimación realizada. Nuestro objetivo es averiguar la eficacia de este tipo de soluciones en la estimación de un patrón de lluvia en tiempo real, analizando la sensibilidad del error de estimación cuando variamos distintos parámetros del sistema. Los resultados experimentales muestran que, a pesar de disponer de apenas un número reducido de vehículos participando en el sistema, es posible lograr una buena estimación de estos fenómenos meteorológicos, aún cuando su

variabilidad espacial es especialmente alta.

El documento está estructurado de la siguiente manera: en la Sección 2 se describe cómo utilizar las VANETs para monitorización ambiental. En la Sección 3 se explica la generación del patrón de precipitación que servirá como referencia para este estudio. La Sección 4 presenta los simuladores utilizados y la metodología para llevar a cabo el análisis. En la Sección 5 se detallan los parámetros de simulación, se muestran los resultados obtenidos, y se realiza la evaluación numérica de la estimación de intensidad de lluvia en el área a estudiar. Finalmente, la Sección 6 recoge las conclusiones del trabajo.

2. Uso de VANETs para Monitorización Ambiental

Las VANETs proporcionan una oportunidad para el desarrollo de aplicaciones que mejoren las condiciones del transporte, del tráfico vehicular, o que brinden información al conductor. Un tipo de información relevante que pueden proporcionar al conductor es la información medioambiental, específicamente la medición de intensidad de lluvia en un determinado sector.

Por un lado, en varios trabajos de investigación [5, 4] se han propuesto técnicas de medición de intensidad de lluvia, como el uso de satélites, enlaces de microondas y pluviómetros acústicos. Por otro lado, en [3] los autores propusieron la idea de usar los vehículos en movimiento como dispositivos de medición de la precipitación. En ese trabajo se utilizó la velocidad del limpiaparabrisas (W) como un indicador de la intensidad de lluvia (R) mediante la aplicación de una relación hipotética W - R , asumiendo cierta estimación del error de intensidad de lluvia en los vehículos.

En [7] los autores desarrollaron y analizaron las relaciones entre las lecturas del sensor (W) y la intensidad de las precipitaciones (R) mediante experimentos de laboratorio. Utilizaron vehículos como pluviómetros en movimiento, con limpiaparabrisas o sensores ópticos como dispositivos de medición.

Un enfoque interesante se propone en [8, 9]. Los autores proponen el uso de VANETs en las que los vehículos están equipados con distintos sensores para monitorizar el medioambiente. Con este objetivo definen un proceso de cinco pasos:

Paso1: Determinar el objetivo de la tarea de monitorización.

Paso2: Asignación de vehículos para la monitorización.

Paso3: Recopilación de los datos de interés.

Paso4: Entrega de los datos recogidos.

Paso5: Procesamiento de los datos recolectados.

Siguiendo estos pasos, en primer lugar hemos determinado que el objetivo de la tarea de monitorización en este trabajo es medir la intensidad de lluvia. Para ello, se ha generado una patrón de precipitación (que se detalla en la Sección 3), y se ha realizado una

caracterización espacial de dicho patrón en términos de intensidad, duración y frecuencia (IDF) [16, 7].

A continuación se ha seleccionado un área específica en la ciudad de Valencia para la monitorización, y en la que se han desplegado vehículos que serán considerados como sensores de intensidad de lluvia. Cabe mencionar que, además de encontrar una relación entre la velocidad del limpiaparabrisas del vehículo y la intensidad de lluvia, es necesario realizar la caracterización temporal de la intensidad de la precipitación utilizando un análisis de frecuencia adecuado, y la representación espacial de esta intensidad mediante una herramienta geoestadística rigurosa. Esto permitirá representar correctamente el patrón de lluvia en un mapa generado para estimar su intensidad en el espacio [25, 26].

Las fuentes de información serán sensores instalados en los vehículos, los cuales medirán la intensidad de la lluvia en una determinada ubicación. Posteriormente, esta información puede ser enviada a un lugar predefinido (servidor) utilizando dispositivos inalámbricos de corto alcance a bordo de los vehículos para ser transferida a otros vehículos cercanos. Es importante destacar que la combinación de datos provenientes de diferentes fuentes puede mejorar la estimación de un patrón de precipitación real [7].

Por último, la información recopilada se almacena para su análisis y procesamiento.

3. Generación de un Patrón de Precipitación Sintético

La lluvia es un fenómeno atmosférico que se inicia con la condensación del vapor de agua contenido en las nubes y luego precipitado, i.e., se convierte en lo suficientemente pesado para caer por gravedad, siendo el responsable de depositar la mayor parte de agua dulce en la tierra y, por ende, favorecer la vida en nuestro planeta [16]. No obstante, si su intensidad es demasiado alta, puede provocar importantes estragos.

La medición de la precipitación se efectúa por medio de pluviómetros o pluviógrafos, siendo los segundos utilizados principalmente cuando se trata de determinar precipitaciones intensas en un corto período. Para que los valores de distintas estaciones pluviométricas sean comparables se utilizan instrumentos estandarizados.

La precipitación pluvial se mide en mm , que sería el espesor de la lámina de agua que se formaría, a causa de la precipitación, sobre una superficie plana e impermeable, y que equivale a litros de agua por metro cuadrado de terreno (l/m^2) [23].

Es importante poder clasificar la lluvia registrada. Para ello, la Agencia Estatal de Meteorología (AEMET) define cinco categorías (ver Tabla 1) que definen las diferentes intensidades de lluvia, y que son: débil, moderada, fuerte, muy fuerte y torrencial, así como los valores de la acumulación en una hora para cada tipo de intensidad [11].

Tabla 1: Clasificación de la precipitación según la intensidad (AEMET).

Intensidad	Acumulación por una hora
Débil	entre 0.1 y 2 mm
Moderado	entre 2.1 y 15 mm
Fuerte	entre 15.1 y 30 mm
Muy Fuerte	entre 30.1 y 60 mm
Torrencial	+ 60 mm

A continuación se explica el diseño del patrón de lluvia que se utiliza en el presente trabajo, y los datos necesarios para su modelado, como son: definición de los umbrales de lluvia y ajuste de los datos obtenidos a una distribución normal mediante una función Gaussiana.

3.1. Umbrales de Precipitación

El patrón de lluvia propuesto está diseñado para una porción del mapa de la ciudad de Valencia, específicamente para un área de $4 \times 3 \text{ Km}$. Primeramente es necesario especificar los umbrales de lluvia con los que vamos a trabajar en términos de tiempo de precipitación, distancia e intensidad. Para ello se debe tomar en cuenta los siguientes criterios que proporciona la AEMET [11]:

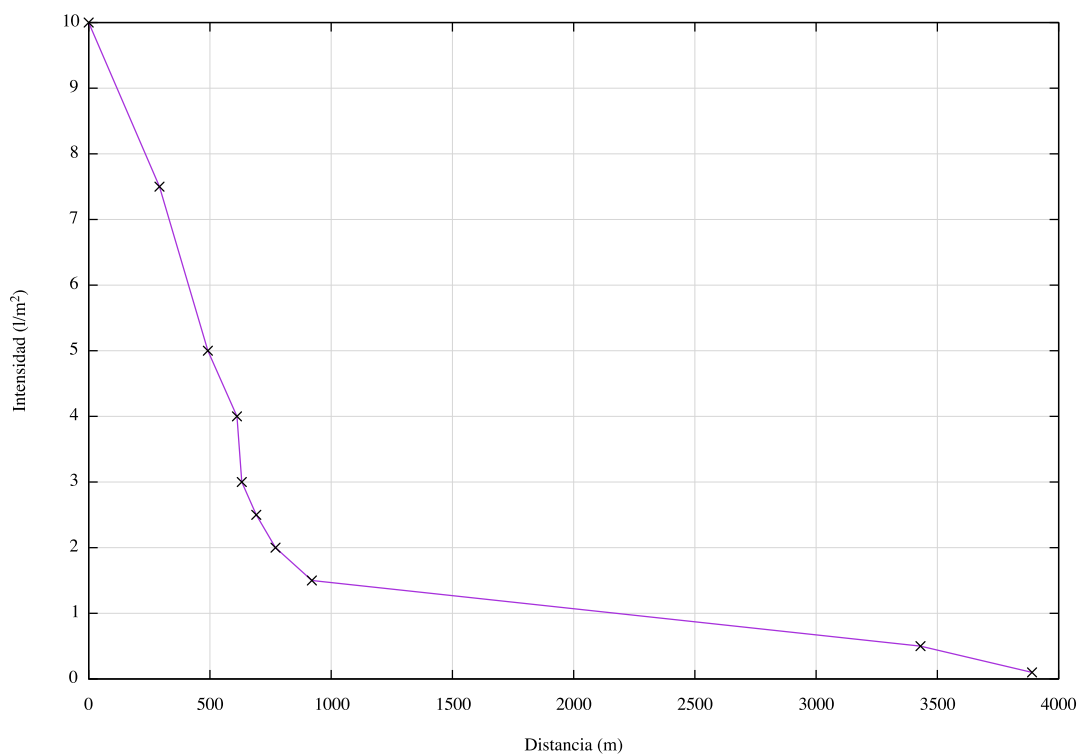
- La lluvia no es ni perfectamente constante ni infinitamente intensa, presentando máximos y mínimos progresivos, con una cierta distribución regular.
- Los fenómenos que tienen más interés son los que presentan una distribución *centrada* en un único máximo, de forma similar a una distribución gaussiana.
- Oficialmente se clasifica la intensidad de la lluvia según la cantidad registrada en una hora, pero sin embargo los registros pueden ser variables, i.e., del total de lluvia registrado, la mayor parte se distribuye en un tiempo inferior a una hora, y el resto se reparte hasta completar la hora.

Considerando lo anteriormente expuesto, la Tabla 2 recoge datos tomados para precipitaciones instantáneas de 10 minutos [17] en los que se midió la distancia en metros (m) y la intensidad de la precipitación medida en litros por metro cuadrado (l/m^2), teniendo $10 l/m^2$ como valor máximo de intensidad.

La Fig. 1 representa la curva que refleja los valores de referencia obtenidos, ilustrando la bajada de intensidad según distancia para un período de 10 minutos. A partir de los datos de esta curva (generada con la información de la Tabla 2) se realiza un ajuste gaussiano que se detalla en la Sección 3.2.

Tabla 2: Datos de intensidades de lluvia para una precipitación de $10'$.

Distancia (m)	Intensidad (l/m^2)
0.0	10.0
292	7.5
491.5	5.0
611.5	4.0
631.5	3.0
691.0	2.5
770.5	2.0
920.5	1.5
3430.5	0.5
3890.5	0.1

**Figura 1:** Curva de Umbrales de Precipitación.

3.2. Ajuste Gaussiano de los datos

Con el fin de predecir la magnitud de intensidad para cualquier distancia al epicentro de la masa de precipitación, realizamos un ajuste gaussiano de los datos. La función Gaussiana está definida por:

$$f(x) = \frac{a}{\sqrt{2\pi s^2}} e^{-\frac{x^2}{2s^2}}, \quad (1)$$

donde a es la media del ajuste y s es la desviación estándar.

El proceso de ajuste se realizó utilizando gnuplot [12] y se obtuvieron los siguientes resultados:

```
FIT:   data read from "gaussian_input.dat" using 1:2
       format = x:z
       #datapoints = 15
       residuals are weighted equally (unit weight)
```

```
function used for fitting: gauss(x)
gauss(x) = a/(2*pi*s**2)**0.5*exp(-x**2/(2*s**2))
```

```
final sum of squares of residuals : 1.41287
rel. change during last iteration : 0
```

```
degrees of freedom      (FIT_NDF)                : 13
rms of residuals        (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.32967
variance of residuals (reduced chisquare) = WSSR/ndf : 0.108682
```

Final set of parameters	Asymptotic Standard Error
=====	=====
s = 429.965	+/- 10.22 (2.377%)
a = 10403.7	+/- 215.5 (2.071%)

La Fig. 2 representa tanto los datos de umbrales de lluvia como la curva de gauss una vez se ha realizado el proceso de ajuste. Se puede observar que el ajuste realizado es

aceptable como punto de partida para la generación de patrones combinados de masas de precipitación.

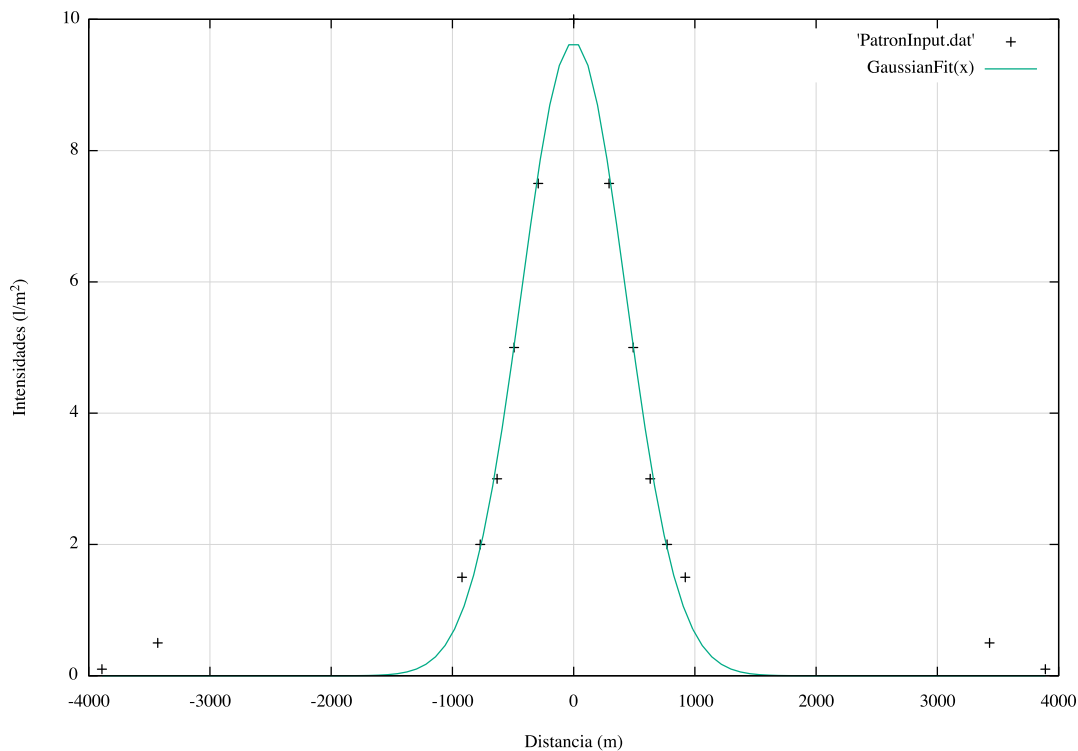


Figura 2: Ajuste gaussiano de los datos.

3.3. Masa de Precipitación Generada

El modelo propuesto se realizó utilizando el lenguaje de programación para cálculo estadístico y gráficos R [18], concretamente mediante la herramienta RStudio [19]. RStudio es un entorno de desarrollo integrado (IDE) para R que incluye una consola, editor de sintaxis que apoya la ejecución directa de código, así como herramientas para el trazado, la depuración y la gestión del espacio del trabajo.

Como se mencionó en la Sección 3, el modelo de patrón de lluvia sintético se realizó para una porción representativa del mapa de Valencia (un área de $4 \times 3 \text{ Km}$). Para ello se utilizaron las coordenadas de borde que se muestran en la Tabla 3. En la Fig. 3 se representa el área elegida para el estudio.

Tabla 3: Coordenadas de Borde

	Latitud	Longitud
Punto inicial (Inferior izquierdo)	39.464914	-0.406639
Punto final (Superior derecho)	39.491919	-0.361717

Con el fin de crear el mapa de patrón de lluvia sintético para este sector de la ciudad de Valencia, primero se obtiene una ecuación bidimensional basada en la función gaussiana de la Eq. (1) como se muestra a continuación:

$$f(x, y) = \frac{a}{\sqrt{2\pi s^2}} e^{-\frac{(x-x_0)^2}{(2s^2)} + \frac{(y-y_0)^2}{(2s^2)}}. \quad (2)$$

Después, dentro de la zona se combinaron 6 puntos de coordenadas (hotspots) con los que se formó una especie de *frente frío* entrando por la ciudad; la Fig. 3 ilustra tanto las coordenadas de borde como las coordenadas de los hotspots en la zona elegida de la ciudad de Valencia. Los hotspots representan los puntos con mayor intensidad de lluvia, y sus coordenadas se detallan en la Tabla 4.

Tabla 4: Coordenadas GPS y XY de los puntos con mayor intensidad de lluvia.

<i>Descripción</i>	Coordenadas GPS		Coordenadas XY	
	<i>Latitud</i>	<i>Longitud</i>	<i>Latitud</i>	<i>Longitud</i>
Hotspot 1	39.486500	-0.396678	856.05033369786	2402.942528773
Hotspot 2	39.488363	-0.389219	1497.07828140514	2610.33074124855
Hotspot 3	39.475907	-0.388786	1534.29038617311	1223.73515992644
Hotspot 4	39.488981	-0.380444	2251.20353045773	2679.12618547425
Hotspot 5	39.477729	-0.380776	2222.67138479902	1426.55927703622
Hotspot 6	39.483625	-0.374916	2726.28094943175	2082.89899348579

Finalmente, haciendo coincidir las coordenadas de cada uno de los hotspots con los parámetros (x_0, y_0) en la Eq. (2), somos capaces de generar el patrón de precipitación sintético que se representa en la Fig. 4. En este patrón se puede observar el frente frío generado, donde las intensidades de lluvia más altas se representan con el color amarillo intenso.

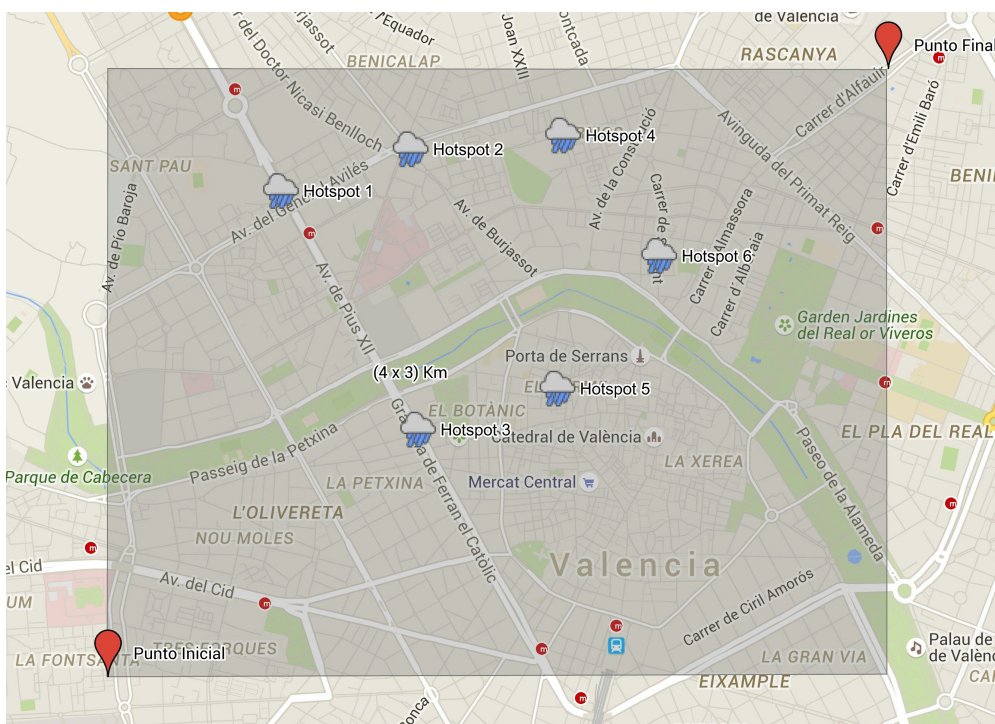


Figura 3: Área a estudiar y hotspots de lluvia definidos para la ciudad de Valencia.

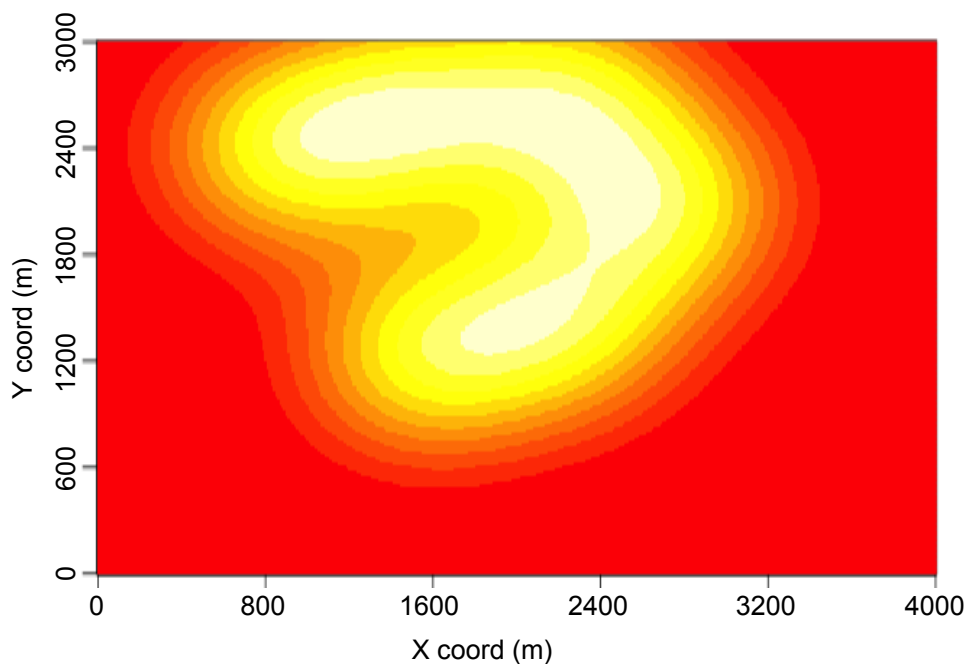


Figura 4: Mapa de Patrón de Lluvia.

Como resultado del proceso detallado anteriormente, se obtiene una matriz de intensidades de lluvia a la que denominamos *mat*. Esta matriz contiene las intensidades de

lluvia que serán leídas por los vehículos en cada punto del mapa por donde circulen en la simulación.

4. Framework de Simulación

La simulación se llevó a cabo mediante la herramienta OMNeT++ [20], la cual es un simulador de eventos discreto orientado a objetos. Para simular el movimiento de los vehículos en un escenario usamos el simulador de tráfico microscópico llamado "Simulación de Movilidad Urbana (*SUMO*)". El archivo de configuración de la red de carreteras y el vehículo en movimiento están contruidos en *SUMO*.

Para integrar los datos de movilidad de los vehículos en el entorno de simulación, las herramientas OMNeT++ y *SUMO* se han ejecutado en paralelo, estando conectadas entre sí a través de un socket TCP, como se puede observar en la Fig. 5. El protocolo para esta comunicación se ha estandarizado como la Interfaz de Control de Tráfico (TraCI) [27]. Esta interfaz nos dará acceso a la simulación del tráfico por carretera utilizando OMNeT++, y permitirá recuperar los valores de los objetos simulados (vehículos) con el fin de controlar su comportamiento (determinar su posición y velocidad en el mapa).

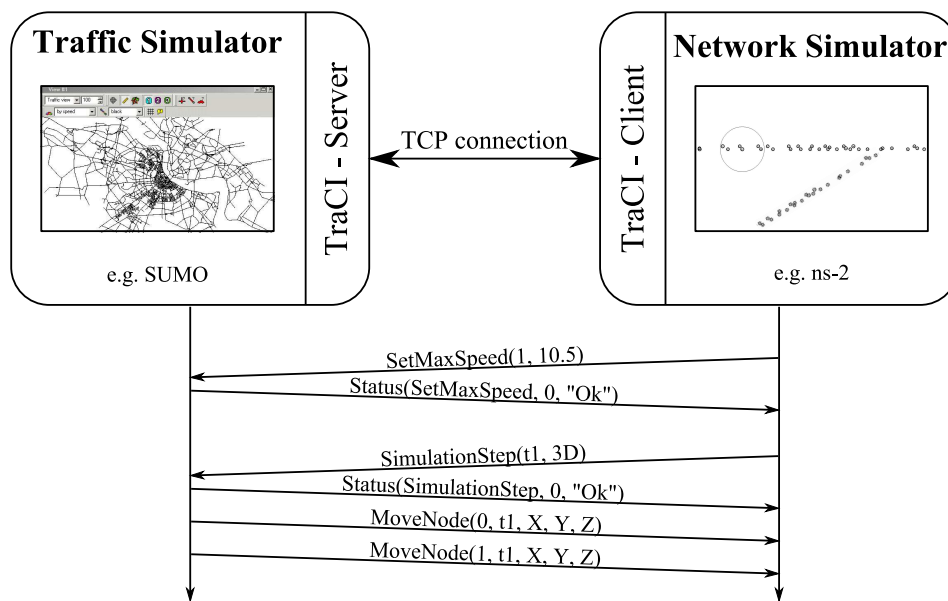


Figura 5: Framework de simulación [27].

A continuación se detallan las configuraciones de SUMO y OMNeT++ para llevar a cabo las simulaciones que nos permitirán cumplir con los objetivos propuestos en este trabajo. Además, se detallará el proceso mediante el cual se han ampliado las funcionalidades existentes de cara a permitir integrar los datos de intensidad de precipitación en el entorno de simulación.

4.1. Simulation for Urban MObility –SUMO–

SUMO (Simulación para Movilidad Urbana) es un software de código abierto altamente portátil, para simulación de tráfico microscópico [21], lo que quiere decir que cada vehículo que se mueve por la red simulada es modelado individualmente, y está caracterizado por una cierta posición y velocidad.

SUMO está diseñado para simular grandes redes viales del tamaño de una ciudad, aunque también es capaz de modelar redes de mayor dimensión, como un sistema de autopistas, puesto que tiene la ventaja de que su simulación es multimodal, donde no sólo se modelan los movimientos de los vehículos, sino también sistemas de transporte público y hasta trayectos peatonales; esto está definido mediante múltiples rutas, las cuales pueden ser compuestas por subrutas, y así poder contar con una sola modalidad de tráfico en cada una.

Las funcionalidades de las que dispone el simulador pueden resumirse en:

- Simulación microscópica
- Modelado explícito de vehículos, peatones y transporte público
- Interacción en línea
- Control de simulación con TraCI
- Simulación de tráfico multimodal, como vehículos, transporte público y peatones
- Los horarios de luces de tráfico pueden ser importados o generados de manera automática por SUMO
- Soporte de distintos formatos.

4.1.1. Movilidad vehicular usando SUMO

Para generar movilidad vehicular en SUMO se ha partido de los mapas disponibles en el proyecto OpenStreetMap [22]. Concretamente, se ha utilizado este último como fuente cartográfica para la ciudad de Valencia, elegida como objeto de nuestro estudio. El mapa ha sido importado mediante el programa Netconvert que acompaña al simulador SUMO. La Fig. 6 muestra un resumen de los pasos a seguir para generar los archivos de red necesarios para la simulación en SUMO.

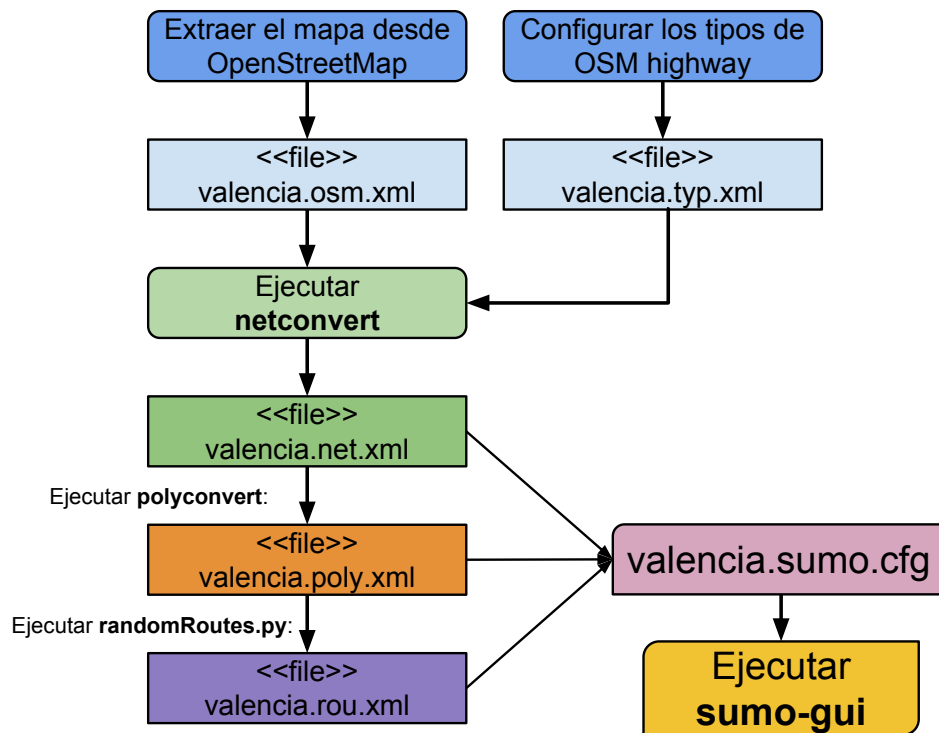


Figura 6: Generación de la Red Vehicular utilizando SUMO.

OpenStreetMap También conocido como OSM, es un proyecto colaborativo para crear mapas libres editables. Estos mapas se crean utilizando datos geográficos capturados por dispositivos GPS móviles, cartografías y otras fuentes libres. Lo primero que se necesita es descargar el mapa desde OpenStreetMap [22], y para ello exportamos las coordenadas de la Tabla 3 utilizadas para la creación del patrón de lluvia. La Fig. 7 ilustra la sección exportada desde OSM del mapa de la ciudad de Valencia.

Netconvert Es una aplicación de línea de comandos que importa redes de carreteras digitales de distintas fuentes, y que genera redes de carreteras que pueden ser utilizadas por otras herramientas del paquete. Para importar la red de OpenStreetMap se generó el archivo "valencia.net.xml" mediante el siguiente comando de consola:

```
netconvert --osm-files valencia.osm -o valencia.net.xml
```

Polyconvert Es un comando de consola que importa formas geométricas (polígonos o puntos de interés) de diferentes fuentes, y los convierte en una representación que puede ser visualizada utilizando SUMO-GUI. Polyconvert es capaz de importar formas de diferentes tipos de archivos. Normalmente, para importar datos de un de-

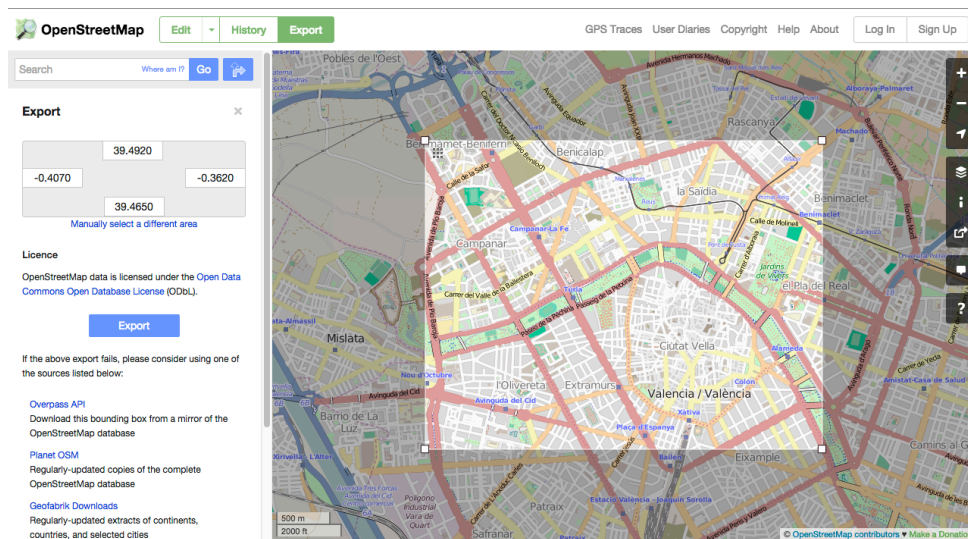


Figura 7: Exportación del mapa desde OSM.

terminado tipo de archivo, se utiliza como nombre de opción el tipo de archivo, y el valor indica la posición del archivo.

```
polyconvert --net-file valencia.net.xml --osm-files valencia.osm
--type-file typemap.xml -o valencia.poly.xml
```

randomRoutes.py Es un script en python que permite generar rutas aleatorias para los vehículos en la red. Como parámetros necesita el tiempo inicial y final de la simulación, y un periodo que permite ajustar el número de vehículos que conformarán la red de carreteras. A continuación se muestra a modo de ejemplo un script que genera 150 vehículos con rutas aleatorias en la red de carreteras durante 60 s.

```
python /Users/kcicenia/Documents/MASTER/TFM/instaladores/SUMO/
sumo-0.19.0/tools/trip/randomTrips.py -n valencia.net.xml
--begin 0 --end 60 --period 0.4 -r valencia.rou.xml
```

sumo-gui Lanza la interfaz de SUMO para generar la simulación de movilidad vehicular, como se observa en la Fig. 8.

```
sumo-gui -c valencia.sumo.cfg
```

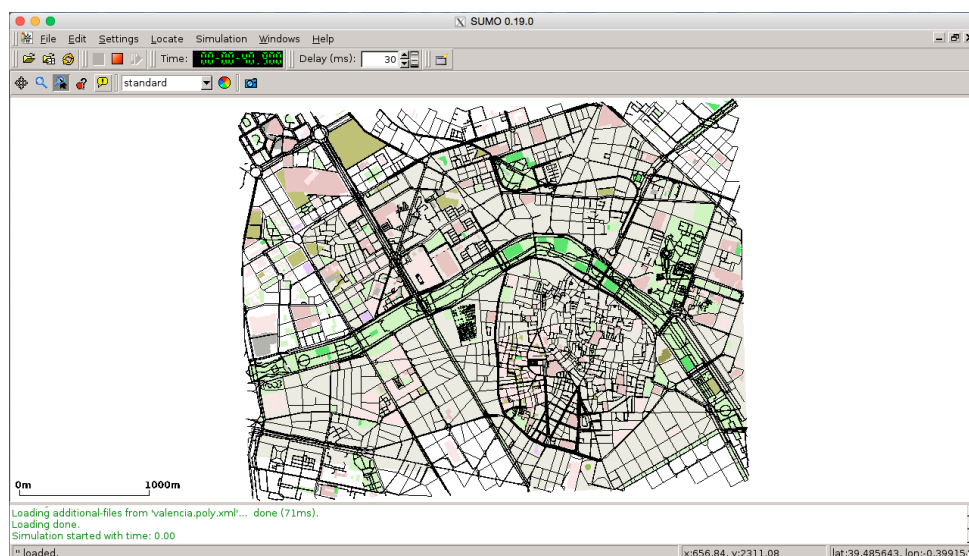


Figura 8: Generación de movilidad vehicular en SUMO.

4.2. OMNeT++

OMNeT++ [20] es un framework extensible y modular que incluye una librería de simulaciones con componentes basados en el lenguaje C++ (orientado a objetos). Tiene una arquitectura genérica, por lo que puede ser utilizado en varios dominios, siendo capaz de resolver varios problemas como:

- Modelado de comunicación de redes, tanto cableadas como inalámbricas.
- Modelado de protocolos.
- Modelado de colas de redes.
- Modelado de multiprocesadores y otras distribuciones de sistemas hardware.
- Validación de arquitecturas de hardware.
- Evaluación de rendimientos en sistemas de software complejos.
- Modelado y simulación de cualquier sistema basado en eventos que pueda ser mapeado en entidades que intercambian mensajes.

4.2.1. Estructura Modular de la Simulación

Un modelo OMNeT++ consiste de módulos jerárquicos que se comunican mediante paso de mensajes. Los módulos básicos están compuestos por módulos simples, los mismos que pueden ser agrupados en módulos compuestos y así sucesivamente, formando

el módulo del sistema; todo está escrito en C++, utilizando la biblioteca de clases de simulación. El modelo de todo, llamado red en OMNeT++, es en sí mismo un módulo compuesto.

4.3. Integración de datos de sensorización en OMNeT++

El simulador OMNeT++ no dispone de un interfaz que permita, de manera transparente, añadir bases de datos de información para consulta por parte de los nodos móviles. Por esa razón, y de cara a cubrir los objetivos definidos para este trabajo, fue necesario la creación de módulos y su interconexión con los existentes. Específicamente, para sensorización se crearon dos módulos: RainManager y RainSensor. La Fig. 9 muestra la estructura modular y conexión de los principales módulos implementados en OMNeT++ para realizar las simulaciones con los criterios definidos.

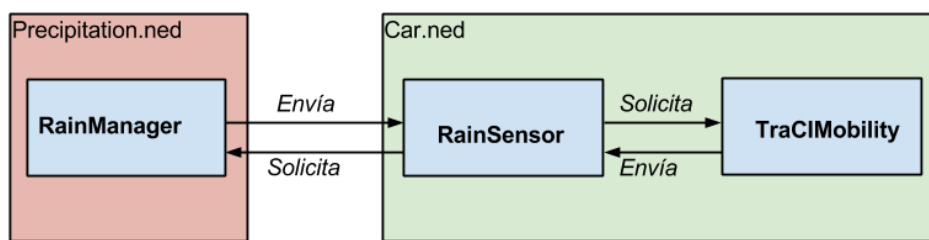


Figura 9: Estructura Modular en OMNeT++.

La transferencia de información es la siguiente: TraCIMobility es un módulo que se conecta con SUMO mediante la interfaz TraCI y se encarga de gestionar la movilidad vehicular. RainSensor es un módulo que, por un lado, se conecta con TraCIMobility para solicitar los valores de posición de un determinado vehículo, y por otro lado, se conecta con RainManager, que es el módulo que se encarga de consultar en la matriz de precipitación *mat* los valores de intensidad de lluvia que el módulo RainSensor solicita. Esta transferencia de información se realiza periódicamente según las actualizaciones de posición del vehículo, y se almacenan los resultados para su análisis.

A continuación se detallan todos los módulos simples y compuestos utilizados. La interconexión de los módulos se muestra en la Fig. 10. Para describir la estructura del modelo de simulación del presente trabajo utilizamos el lenguaje NEtwork Description

(NED). NED permite declarar módulos simples, así como conectarlos y ensamblarlos en módulos compuestos.

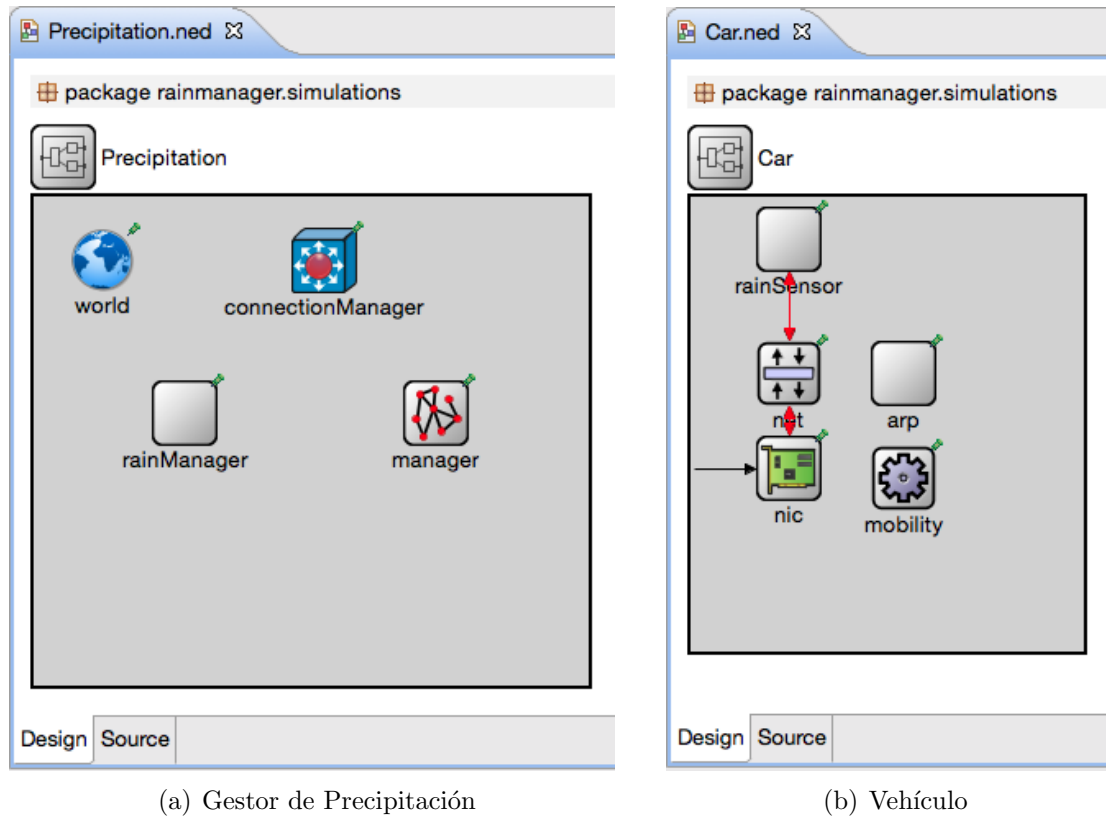


Figura 10: Estructura del Modelo de Simulación

4.3.1. Precipitation.ned

En `Precipitation.ned` declaramos los módulos que nos permitirán, por un lado, leer la matriz de intensidades de lluvia a la que denominamos *mat* (ver Sección 3.3); y, por otro lado, conectarnos con SUMO, el cual se encarga de generar la movilidad vehicular en la red vial del área de Valencia elegida. A continuación se describen los módulos creados:

RainManager Es un módulo simple que, periódicamente y para cada vehículo, consulta al fichero *mat* la intensidad de lluvia en base a la ubicación del vehículo en el simulador. El RainManager tiene los siguientes métodos:

- **initialize:** lee la matriz *mat* y la guarda en un vector de vectores con los tamaños necesarios; además, calcula la tasa de muestreo con la que se obtuvieron

las lecturas de intensidad en el mapa.

- **getRain(Coord pos)** devuelve el valor de intensidad de lluvia para una posición dada.

manager: Es un módulo simple que extiende la funcionalidad de `TraCIScenarioManager` para su uso con `sumo-launchd.py` y SUMO. Se conecta a una instancia en ejecución del script `sumo-launchd.py` para iniciar/detener automáticamente SUMO cuando se inicia/finaliza la simulación. Las demás funcionalidades son proporcionadas por el `TraCIScenarioManager`.

TraCIScenarioManager Es un módulo simple que se encarga de crear y mover los nodos (vehículos) controlado por un servidor TraCI.

4.3.2. Car.Ned

En `Car.Ned` declaramos los módulos que nos permitirán, por un lado implementar la capa de red de comunicación entre los vehículos y, por otro lado, implementar un módulo de aplicación en los vehículos para obtener las lecturas de intensidad de lluvia correspondientes a la ubicación del vehículo. A continuación se describen los módulos creados:

RainSensor Es un módulo simple de aplicación que se suscribe a la señal de TraCI. TraCI, al estar enlazado con SUMO (herramienta que realmente gestiona la movilidad vehicular) le indicará que ha habido un cambio en la posición. Por lo tanto, este módulo obtiene un valor de intensidad de lluvia para la posición dada mediante un puntero al módulo `RainManager`. El `RainSensor` tiene los siguientes métodos:

- **initialize:** contiene un puntero al módulo **RainManager** y un puntero al módulo **TraCIMobility**. Además, inicializa el vector donde se almacenan las lecturas de intensidad de lluvia.
- **receiveSignal** se suscribe a la señal desde TraCI que le indicará que ha habido un cambio en la posición; a continuación llamará al método **handlePositionUpdate** que se detalla a continuación.
- **handlePositionUpdate** primeramente obtiene la posición en la que se encuentra el vehículo; a continuación llama al método **getRain** del módulo

RainManager para obtener el valor de intensidad de lluvia en la posición del vehículo.

TraCIMobility Es un módulo que gestiona la movilidad de nodos controlados por TraCIScenarioManager. Recibe actualizaciones de posición y estado de un módulo externo (SUMO), y actualiza el módulo matriz en consecuencia (RainSensor).

4.3.3. Parámetros de simulación relevantes (*omnetpp.ini*)

A continuación, en la Tabla 5 se presentan los parámetros de configuración relevantes que se han utilizado en la estructura del modelo de simulación de la Fig. 10:

Tabla 5: Parámetros de simulación relevantes

Nombre Parámetro	Valor	Comentario
Simulation parameters		
sim-time-limit	60s	Tiempo de Muestreo
rainManager.myFilemat	"myFilemat.txt"	Fichero <i>mat</i>
rainManager.mapSizeX	4000m	Tamaño del Mapa en x
rainManager.mapSizeY	3100m	Tamaño del Mapa en y
TraCIScenarioManager parameters		
manager.updateInterval	1.2s	Tiempo entre Muestras
manager.host	"localhost"	
manager.port	9999	Puerto para Interfaz TraCI
manager.moduleType	rainmanager.simulations.Car"	
manager.moduleName	"node"	
manager.launchConfig	xmldoc("valencia.launchd.xml")	Fichero de Configuración SUMO
Mobility parameters		
node[*].mobilityType	"TraCIMobility"	
node[*].mobility.speed	1mps	
node[*].mobility.updateInterval	1.2s	Tiempo entre muestras
node[*].applType	RainSensor"	

5. Simulación y Análisis de Resultados

Usando el framework de simulación detallado en la Sección 4, en este apartado se estudia en qué medida una red de vehículos es capaz de ofrecer suficiente información como para permitir reconstruir un determinado patrón de precipitación. Para la obtención de lecturas de intensidad de lluvia, se partió del escenario definido en la Sección 3.3, y se han variado diferentes parámetros: el número de vehículos en la red, el tiempo entre muestras, y finalmente el tiempo de simulación. De esta manera es posible averiguar el impacto de estos parámetros en la calidad de la reconstrucción del patrón de precipitación usado como referencia.

El procedimiento para obtener los resultados fue el siguiente: inicialmente se fijó en 120s el tiempo de simulación en OMNeT++. Esto se debe a que la movilidad vehicular generada mediante SUMO se configuró para que un determinado vehículo permanezca aproximadamente 60s en el sistema, contados desde el momento que ingresa en la red hasta que la abandona. El tiempo entre muestras se calculó de tal manera que, en los 120s de simulación, cada vehículo en la red sea capaz de generar como mínimo entre 50 y 100 muestras, dando como resultado tiempos entre muestras de 1.2 y 0.6s, respectivamente.

Como parte del análisis, se realizó la predicción del patrón de precipitación para la zona elegida de la ciudad de Valencia (Sección 3.3) a partir de las lecturas de intensidad de lluvia de los vehículos obtenidas por simulación. Para realizar dicha predicción, existen diferentes técnicas geoestadísticas posibles. Para este trabajo se utilizó la técnica conocida como kriging, considerando que es la solución más adecuada en este contexto, y que se ha utilizado en diversos trabajos de índole similar [13, 15, 14].

Una vez que se tiene la predicción del patrón de precipitación, se calcula su error relativo, e_r , con respecto al patrón de referencia utilizando (3):

$$e_r = \frac{\sum_{k=1}^n |referencia_k - reconstruido_k|}{n}, \quad (3)$$

donde n es el número total de elementos de la matriz reconstruida.

Para el análisis de resultados, se han considerado también errores del 10% y del 20% en la lectura de los sensores con el fin de observar su impacto en la predicción del patrón de precipitación. Este error es esperable cuando el sensor sea de bajo coste, o en aquellos

casos en los que una lectura directa no sea posible, siendo realizada una estimación a partir de la velocidad del limpiaparabrisas. Para ello se modeló una medida de error de $\pm\alpha\%$ como valores uniformemente distribuidos; por lo tanto, en esta parte del análisis, la lectura del sensor de lluvia, r_i , tomará un valor aleatorio en el intervalo $[(1 - \alpha) \cdot r_i, (1 + \alpha) \cdot r_i]$. Considerando estos valores se realiza nuevamente la predicción utilizando una técnica geoestadística (kriging), y se calcula su e_r . A continuación se detallan los resultados obtenidos de la simulación.

5.1. Lecturas de intensidad de lluvia

En las figuras que se presentan a continuación se representa la ejecución de la simulación. Primeramente, la Fig. 11 ilustra la movilidad vehicular generada mediante SUMO en una zona concreta de la ciudad de Valencia. Los vehículos están representados con un triángulo de color amarillo.



Figura 11: Movilidad vehicular generada utilizando SUMO.

Después, la Fig. 12 representa la conexión de OMNeT++ y SUMO para realizar la

comunicación entre vehículos mediante el protocolo TraCI. Podemos ver, en la parte superior izquierda de la figura, parte de los módulos creados, y que intervienen en la comunicación de los nodos. En la parte inferior se observa como los nodos en la red van actualizando su ubicación, lo que implica que leerán la intensidad de lluvia según la posición en la que se encuentren. Para ello, dependiendo del tiempo entre muestras, cada vehículo consulta la matriz de precipitación generada, mat (ver Sección 3.3).

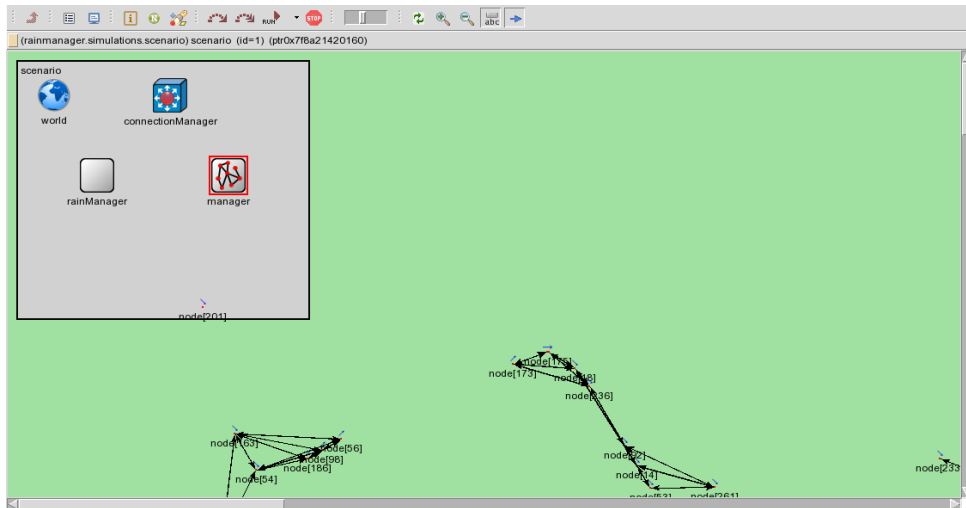
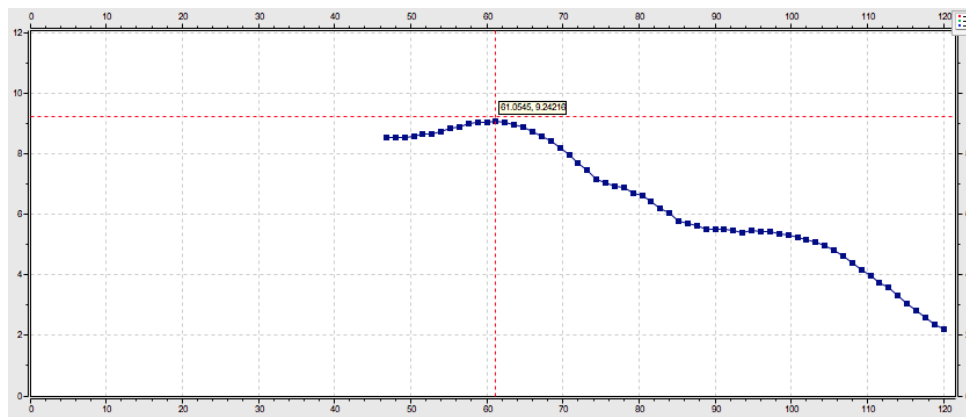
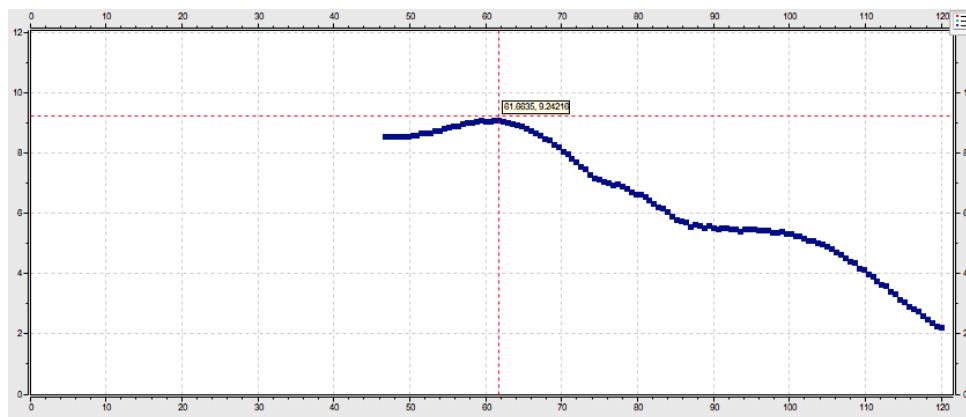


Figura 12: Comunicación entre vehículos utilizando protocolo TraCI.

Finalmente, la Fig. 13 muestra la curva de intensidades de lluvia que un vehículo ha leído durante su permanencia en la red. En el eje de abscisas se representa el tiempo, y en el eje de ordenadas la intensidad de lluvia. Se puede observar como el vehículo va obteniendo diferentes lecturas de intensidad de lluvia según la posición (x, y) del mapa por donde se mueve, y que se relaciona con la matriz mat . Como ejemplo se ilustra que, aproximadamente en la posición $(x, y) = (61, 9)$, la intensidad de lluvia que lee el sensor del vehículo es de $\approx 9,2$. La Fig. 13(a) corresponde a una configuración en la que el tiempo entre muestras es de 1.2s en cada vehículo; se puede observar que, como mínimo, tendremos 50 muestras debido a que el vehículo del ejemplo permanece en la red durante aproximadamente 70s. En la Fig. 13(b) se observa el caso en el que el tiempo entre muestras es de 0.6s; como mínimo tendremos 100 muestras dado que el vehículo permanece en la red aproximadamente 70s.



(a) Tiempo entre muestras igual a 1.2 s.



(b) Tiempo entre muestras igual a 0.6 s.

Figura 13: Variación de la intensidad de precipitación medida a lo largo del tiempo.

5.2. Predicción del patrón de precipitación bajo condiciones ideales

Para este tipo de análisis se considera la lectura de intensidad de precipitación obtenida directamente de la simulación, es decir, se asume que no hubo error en las lecturas de intensidad de lluvia por parte de los sensores.

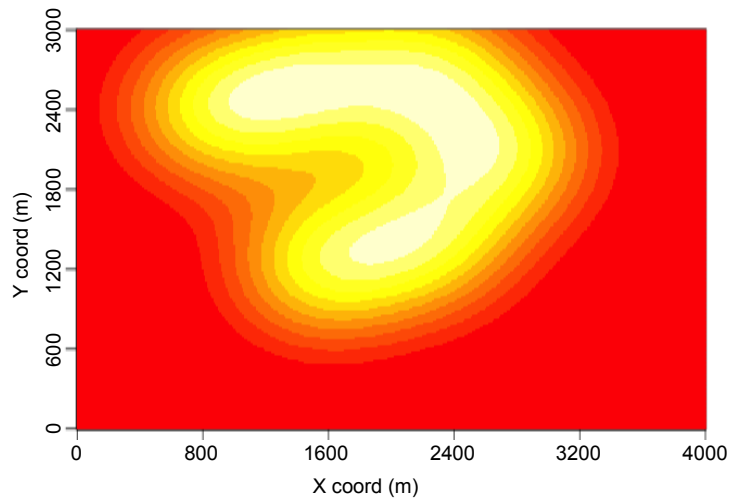
La Fig. 14 ilustra el recorrido que han hecho los vehículos en la red, y las intensidades que han detectado. De color rojo se representan las lecturas de mayor intensidad, y de color azul las de menor intensidad. Se puede observar que se corresponden con el patrón de lluvia de referencia presentado en la Sección 3.3, y que, por comodidad, se repite en la Fig.14(c). Claramente podemos comprobar que, con 50 vehículos en la red, no hay suficiente información como para reconstruir correctamente o tener una idea clara del

patrón de lluvia, aunque se verifica que un incremento del número de vehículos favorece a la reconstrucción del mapa.



(a) Mapa de patrón de lluvia generado con 50 vehículos, tiempo entre muestras=0.6 s

(b) Mapa de patrón de lluvia generado con 250 vehículos, tiempo entre muestras=0.6 s

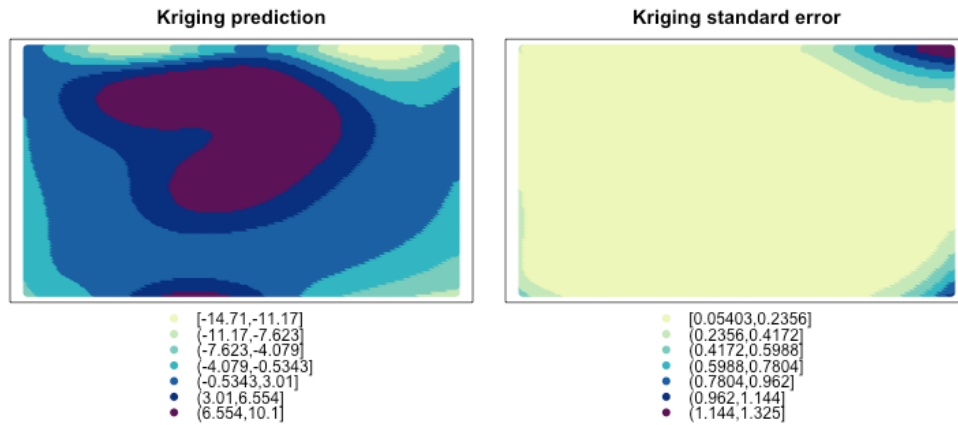


(c) Patrón de precipitación original.

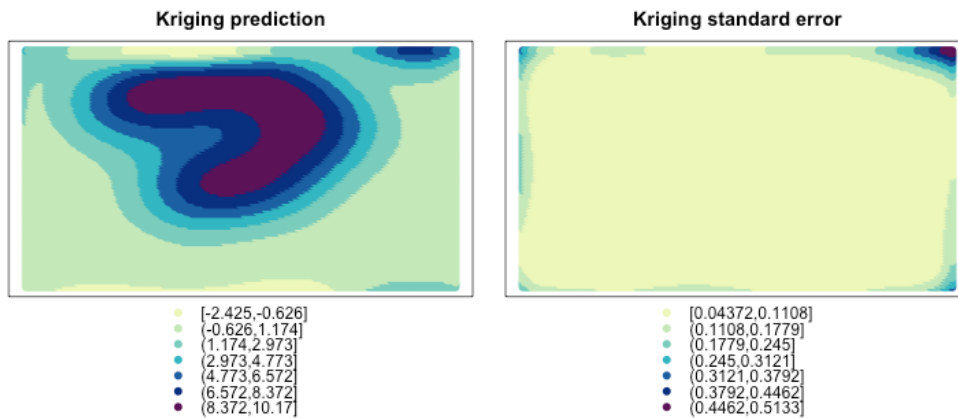
Figura 14: Representación en el mapa de las lecturas de intensidad de precipitación de los vehículos.

A continuación, la Fig. 15 ilustra las predicciones obtenidas utilizando técnicas de interpolación espacial (kriging). Se muestra el caso para 50 vehículos en la red con un tiempo entre muestras de 0.6 s (ver Fig. 15(a)) y para el caso en el que hay 250 vehículos en la red con el mismo tiempo entre muestras (ver Fig. 15(b)). Claramente se observa la

diferencia en la predicción al incrementar el número de vehículos en el escenario, aunque el patrón obtenido con tan solo 50 vehículos ya muestra algunas similitudes con el patrón de referencia.



(a) 50 vehículos, tiempo entre muestras=0.6 s



(b) 250 vehículos, tiempo entre muestras=0.6 s

Figura 15: Predicción utilizando *ordinary kriging*.

Finalmente, en la Fig. 16 se representa el error relativo de la predicción, e_r , variando los parámetros considerados para el estudio, como son el número de vehículos en la red y el tiempo entre muestras. Se observa que aumentar la frecuencia de muestreo beneficia, aunque no de manera notable, a la predicción, siendo el número de vehículos el factor más determinante.

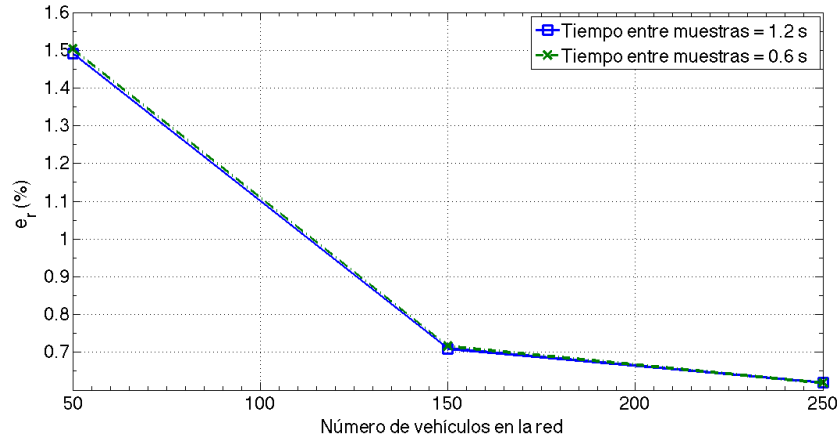


Figura 16: Error relativo en la predicción considerando lectura ideal de los sensores.

5.3. Predicción del patrón de precipitación asumiendo errores de muestreo (lectura de intensidad de lluvia)

Como se mencionó anteriormente, en el estudio se han considerado también errores del 10 % y del 20 % en la lectura de los sensores con el fin de observar su impacto en la predicción del patrón de precipitación.

Con este fin, se modeló una medida de error de $\pm\alpha\%$ como valores uniformemente distribuidos en este intervalo; esto implica que la lectura del sensor de lluvia, r_i , tomará un valor aleatorio en el rango $[(1 - \alpha) \cdot r_i, (1 + \alpha) \cdot r_i]$. La Fig. 17 ilustra como el valor de las lecturas de intensidad de precipitación registradas por un determinado vehículo oscilan alrededor del valor ideal para los dos valores máximos de error usados en nuestros experimentos.

En base a las nuevas muestras con error de medida se procedió nuevamente a realizar la predicción utilizando técnicas de interpolación espacial (kriging) para ver el impacto del error en la estimación realizada. A continuación, la Fig. 18 ilustra las predicciones obtenidas. Se muestra el caso de un escenario formado por 50 vehículos en la red con un tiempo entre muestras de 1.2s. Se observa la predicción obtenida considerando un 10 % de error en las muestras (ver Fig. 18(a)) y un 20 % de error en las muestras (ver Fig. 18(b)). Aunque el patrón obtenido muestra algunas similitudes con el patrón de referencia, se puede observar el impacto negativo en la predicción al incrementar el porcentaje de error en las muestras.

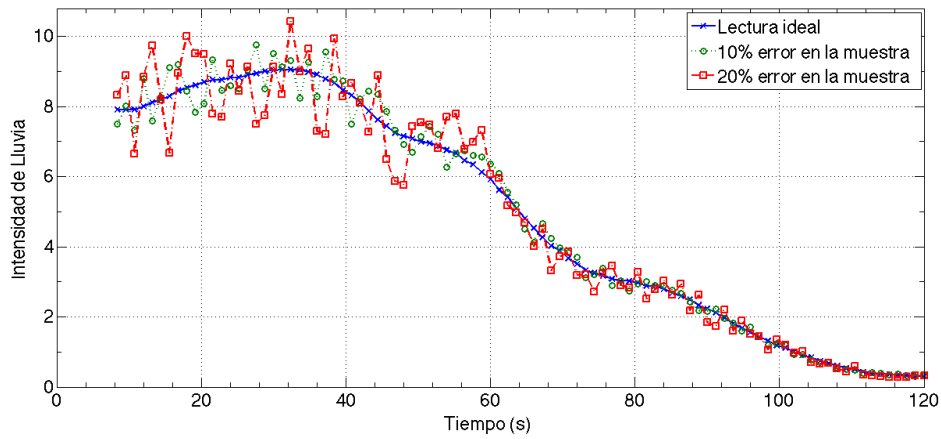
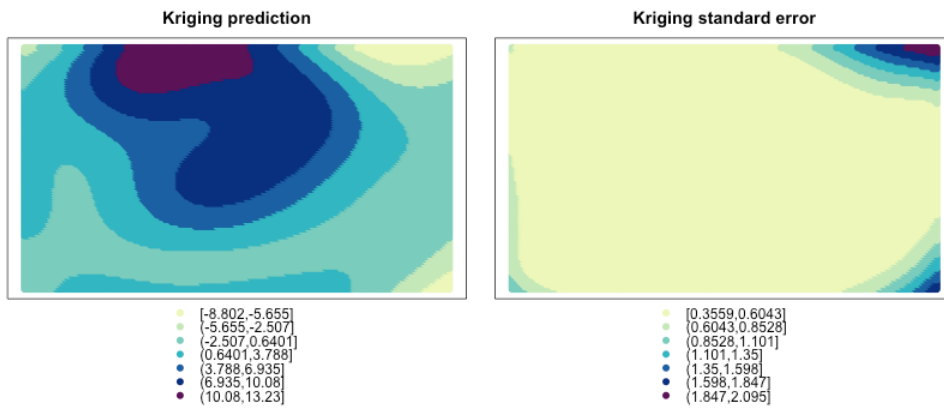
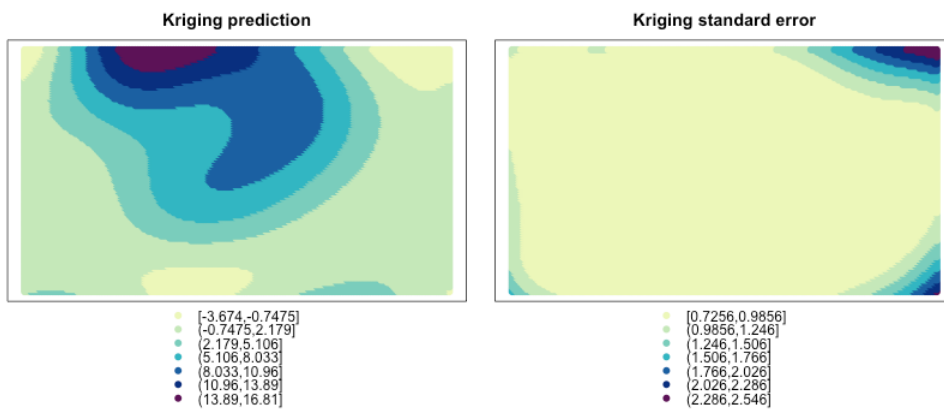


Figura 17: Modelado de error en las muestras.



(a) 50 vehículos, error en la muestra=10%, tiempo entre muestras=1.2 s



(b) 50 vehículos, error en la muestra=20%, tiempo entre muestras=1.2 s

Figura 18: Predicción utilizando *ordinary kriging*.

La Fig. 19 muestra los resultados obtenidos variando tanto el número de vehículos en la red como la tasa de muestreo. Se aprecia un leve incremento en el valor de e_r de la predicción realizada cuando el error del dispositivo sensor es elevado. No obstante, se verifica que un incremento en el número de vehículos que participan en el proceso de sensorización ayuda a mitigar el error del sensor, alcanzando valores bastante bajos si el número de vehículos es superior a 200, independientemente de la tasa de muestreo.

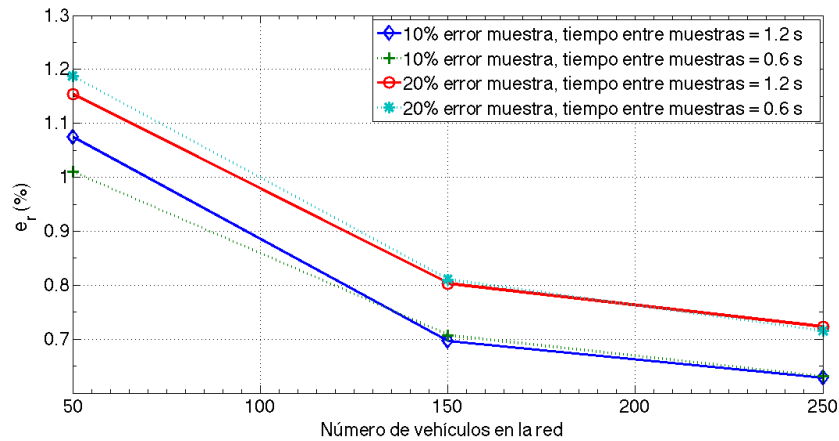


Figura 19: Predicción considerando diferentes niveles de error en las lecturas realizadas.

6. Conclusiones

En este trabajo, explotando las características y facilidades de los vehículos modernos y la información que podemos obtener de ellos, se realizó un estudio basado en simulación con el fin de analizar la eficacia de este tipo de soluciones en la estimación de un patrón de precipitación en tiempo real, analizando la sensibilidad del error de estimación cuando variamos distintos parámetros del sistema. El estudio se realizó para un área específica de la ciudad de Valencia, para la cual se generó un patrón de precipitación sintético. Cada vehículo fue considerado como un sensor de lluvia, y la estimación se realizó en base a la información obtenida, de preferencia en tiempo real, con los diferentes vehículos integrantes del sistema. Se utilizó una herramienta geoestadística (kriging) para la predicción del patrón de precipitación a partir de las lecturas de los sensores. Los resultados experimentales muestran que, con un número reducido de vehículos implicados, es posible tener una buena estimación de estos fenómenos meteorológicos aún cuando su variabilidad espacial es especialmente alta. Además, incrementar el número de vehículos que participan en el proceso de sensorización favorece la mitigación del error del sensor, alcanzando valores bastante bajos si el número de vehículos es superior a 200, independientemente de la tasa de muestreo. Como trabajo futuro se puede extender el presente trabajo de distintas maneras: (i) estudiando el error de estimación con distintos patrones, incluyendo patrones de precipitación variables en el tiempo; (ii) analizar el impacto de las prestaciones de la red inalámbrica en el tiempo necesario para recabar información; y (iii) estudiar algoritmos de agregación de datos para reducir la sobrecarga de la red cuando el número de vehículos en el sistema es muy elevado.

Referencias

- [1] Kajioaka, H., Fujimura K., and Fujita Y. "Automatic Wiper Controller Using Optical Rain Sensor." FUJITSU TEN Tech. J., N 2, 1989.
- [2] Kulkarni, Tapan S., and Harsh S. Holalad. "Semi-Automatic Rain Wiper System." International Journal of Emerging Technology and Advanced Engineering ISSN: 2250-2459.
- [3] Haberlandt, U. and Sester, M., "A real rainfall estimation using moving cars as rain gauges - a modeling study," Hydrol. Earth Syst. Sci., 14, 1139-1151, 2010.
- [4] de Jong, S., "Low cost disdrometer," Master thesis report, TU Delft, Delft, the Netherlands, 2010.
- [5] Upton, G. J. G., Holt, A. R., Cummings, R. J., Rahimi, A. R., and Goddard, J. W. F., "Microwave links: The future for urban rainfall measurement?", Atmos. Res., 77, 300-312, 2005.
- [6] Mitra, Gaurav, Chandreyee Chowdhury, and Sarmistha Neogy. "Application of mobile agent in VANET for measuring environmental data." Applications and Innovations in Mobile Computing (AIMoC), 2014. IEEE, 2014.
- [7] Rabiei, E., et al. "Rainfall estimation using moving cars as rain gauges-laboratory experiments." Hydrology and Earth System Sciences 17.11 (2013): 4701-4712.
- [8] Urrea, Oscar, et al. "Using hitchhiker mobile agents for environment monitoring." 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009). Springer Berlin Heidelberg, 2009.
- [9] Urrea, Oscar, et al. "Mobile agents in vehicular networks: Taking a first ride." Advances in Practical Applications of Agents and Multiagent Systems. Springer Berlin Heidelberg, 2010. 119-124.
- [10] Ali, A., Abteu, W., Van Horn, S. and Khanal, N., "Temporal and spatial characterization of rainfall over central and south Florida." Journal of the American Water Resources Association, Vol. 36, IS 4, Blackwell Publishing Ltd. 2000, 833-848.
- [11] AEMET: Interpretación de Predicciones, [Online]
<http://www.aemet.es/es/eltiempo/prediccion/provincias/ayuda>

- [12] Gnuplot, [Online]
<http://www.gnuplot.info/>
- [13] Kriging, [Online]
<https://en.wikipedia.org/wiki/Kriging>
- [14] Aryal, Santosh K., et al. Characterizing and modeling temporal and spatial trends in rainfall extremes." Journal of Hydrometeorology 10.1 (2009): 241-253.
- [15] Rabiei, E., et al. Rainfall estimation using moving cars as rain gauges-laboratory experiments." Hydrology and Earth System Sciences 17.11 (2013): 4701-4712.
- [16] Wikipedia: Lluvia, [Online]
<http://es.wikipedia.org/wiki/Lluvia>
- [17] Tiempo Severo: Precipitación, [Online]
<http://www.tiemposevero.es/ver-reportaje.php?id=103>
- [18] The R Project for Statical Computing, [Online]
<https://www.r-project.org/>
- [19] RStudio, [Online]
<https://www.rstudio.com/>
- [20] OMNeT++, [Online]
<https://omnetpp.org/>
- [21] SUMO: Simulation of Urban MObility, [Online]
http://sumo.dlr.de/wiki/Main_Page
- [22] OSM: OpenStreetMap, [Online]
<https://www.openstreetmap.org/relation/344953>
- [23] Wikipedia: Precipitación (Meteoreología), [Online]
[https://es.wikipedia.org/wiki/Precipitaci3n_\(meteorolog3a\)](https://es.wikipedia.org/wiki/Precipitaci3n_(meteorolog3a))
- [24] ReviewCivilPE: Steel's formula, [Online]
<http://www.reviewcivilpe.com/rainfall-intensity-analysis/>

-
- [25] Santosh K. Aryal, Bryson C. Bates, Edward P. Campbell, Yun Li, Mark J. Palmer, and Neil R. Viney, "Characterizing and modeling temporal and spatial trends in rainfall extremes." *J. Hydrometeor*, 10. 2009, 241-253.
- [26] Waymire, Ed., Gupta, Vijay K., "The mathematical structure of rainfall representations: A review of the stochastic rainfall models." *Water Resources Research*, Vol. 17, IS 5. 1261-1272.
- [27] Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., and Hubaux, JP., 2008, "TraCI: an interface for coupling road traffic and network simulators." In *Proceedings of the 11th communications and networking simulation symposium (CNS '08)*. ACM, New York, NY, USA, 155-163. DOI=10.1145/1400713.1400740 <http://doi.acm.org/10.1145/1400713.1400740>