



UNIVERSIDAD DE CÓRDOBA

Programa de Doctorado:

COMPUTACIÓN AVANZADA, ENERGÍA Y PLASMAS

MODELLING EFFICIENT DATA
TRANSMISSION FOR MULTIVARIATE
THRESHOLD-BASED EVENTS



MODELADO DE TRANSMISIÓN EFICIENTE
DE DATOS PARA EVENTOS
MULTIVARIANTES BASADOS EN UMBRAL

Directores:

DR. JOAQUÍN OLIVARES BUENO
DR. JOSÉ MANUEL PALOMARES MUÑOZ

Autor

D. FERNANDO LEÓN GARCÍA

Fecha de depósito de la tesis

3 OCTUBRE 2019

TITULO: *MODELLING EFFICIENT DATA TRANSMISSION FOR
MULTIVARIATE THRESHOLD-BASED EVENTS*

AUTOR: *Fernando León García*

© Edita: UCOPress. 2019
Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

[https://www.uco.es/ucopress/index.php/es/
ucopress@uco.es](https://www.uco.es/ucopress/index.php/es/ucopress@uco.es)



TÍTULO DE LA TESIS: Modelling efficient data transmission for multi-variate threshold-based events

DOCTORANDO: Fernando León García

INFORME RAZONADO DE LOS DIRECTORES DE LA TESIS:

El doctorando ha realizado un trabajo de investigación novedoso en el que ha mostrado la evolución personal y académica desde ingeniero hasta investigador. El doctorando ha realizado numerosas propuestas y discusiones con los directores que han llevado a obtener las aportaciones científicas incluidas en esta tesis doctoral. El doctorando ha realizado una revisión bibliográfica de antecedentes científico-técnicos relativos al campo de investigación de la tesis. Posteriormente, ha propuesto un modelo matemático generalizado para poder aplicar mecanismos de reinterpretación de datos para optimizar el envío de mensajes. Este modelo presenta numerosas novedades respecto a la literatura científica, ya que, entre otras novedades, es capaz de utilizar diferentes tipos de datos, no necesariamente numéricos; permite alterar la función de reinterpretación para realizar transformaciones entre diferentes espacios de datos y manejar múltiples conjuntos de parámetros para aplicar a la reinterpretación a diferentes niveles. Ninguna de estas características está presente en otras propuestas.

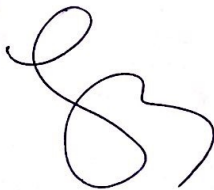
Para poder obtener un conjunto de datos más amplio, el doctorando ha desarrollado un sistema automatizado y parametrizable que permite la generación de datasets pseudo-realísticos con tantos datos como se deseen. Para finalizar, el doctorando ha desarrollado un algoritmo concreto basado en el modelo propuesto en el que los parámetros de reducción se seleccionan automáticamente y se ajustan adaptativamente a las diferentes situaciones puntuales de varias señales que se combinan entre sí, respecto al objetivo de activación de regla booleana que se desea alcanzar. De esta manera, el algoritmo adaptativo propuesto consigue alcanzar reducciones de tráfico muy elevadas sin incurrir en errores de activación de la regla booleana, ni en valor ni en el instante temporal de disparo.

Todas estas aportaciones científicas son novedosas y han producido 3 artículos indexados en JCR. Y con ello demuestran la capacidad del doctorando en el ámbito investigador.

Por todo ello, se autoriza la presentación de la tesis doctoral.

En Córdoba, a 4 de octubre 2019.

Los Directores de la tesis doctoral:



Fdo. Joaquín Olivares Bueno



Fdo. José Manuel Palomares Muñoz

A Ana,
por hacer la vida fácil
con su empatía y cariño.

RESUMEN

Esta tesis doctoral profundiza en la optimización de las comunicaciones en redes de sensores con un propósito específico: evaluar eventos basados en umbral que dependen de múltiples variables distribuidas. Con esta motivación se desarrolla la investigación detallada aquí en forma compendio de artículos. El trabajo desarrollado se estructura en 3 aportaciones científicas en artículos. De esas 3 aportaciones, el trabajo en su vertiente más teórica se desarrolla en 2 de ellas, quedando el tercer artículo para la presentación de una herramienta de soporte metodológico con gran impacto científico y de relevancia en esta tesis doctoral.

Gracias a las dos aportaciones teóricas y de gran calado en el ámbito propuesto se propone una solución que se plantea en forma de hipótesis. La primera aportación son los fundamentos matemáticos para modelar la reducción de datos en la red de sensores y medir su incidencia en la calidad de la evaluación del evento. Para ello define una serie de funciones y parámetros que alteran la cardinalidad de los dominios matemáticos en los que se define la información, así como métricas específicas que tienen en cuenta los desfases temporales en los cambios de estado de la condición evaluada. La segunda aportación es un algoritmo adaptativo que, considerando el contexto lógico de la información del sistema, parametriza el modelo propuesto en tiempo de ejecución. Como resultado, esta técnica maximiza la reducción de tráfico y minimiza el error en la evaluación del evento simultáneamente, obteniendo resultados prometedoros.

Como tercera aportación se describe también un procedimiento para generar señales aleatorias pseudo-realistas, una herramienta útil para disponer fácilmente de grandes conjuntos de datos adecuados para experimentación, que ha sido utilizada en las aportaciones descritas.

ABSTRACT

This doctoral thesis delves into the optimization of communications in sensor networks for a specific purpose: to evaluate threshold-based events that depend on multiple distributed variables. This motivation is behind the detailed research presented here in the form of a compendium of papers. The developed work is structured in 3 scientific contributions in articles. Out of those 3 contributions, the most theoretical work has been described in 2 of them, leaving the third article for the presentation of a methodological support tool with great scientific impact and relevance in this doctoral thesis.

Due to the two theoretical and large-scale contributions in the proposed field, a solution is proposed which is stated as an hypotheses. The first contribution is the mathematical foundations for modelling data reduction in the sensor network and measuring its influence on the quality of the event evaluation. For this purpose, a set of functions and parameters is defined. This logic modifies the cardinality of the mathematical domains in which information is defined in order to save traffic. Specific metrics that consider the time delays in the state changes of the evaluated condition are also defined. The second contribution is an adaptive algorithm that, taking into account the logical context of the system information, parameterizes the proposed model at runtime. As

a result, this technique maximizes traffic reduction and minimizes error in the evaluation of the event simultaneously, obtaining promising results.

As a methodological contribution, a procedure for generating pseudo-realistic random signals is also described, a useful tool for easily obtaining large datasets suitable for experimentation, which has been applied in the described contributions.

PUBLICACIONES CIENTÍFICAS

Las ideas, imágenes y datos que sustentan esta tesis doctoral han sido previamente publicadas en los siguientes trabajos:

- Fernando León-García, Francisco J. Rodríguez-Lozano, Joaquín Olivares y José M. Palomares *Data Communication Optimization for the Evaluation of Multivariate Conditions in Distributed Scenarios*. *IEEE Access*, 7: 123473–123489, 2019.
DOI: [10.1109/ACCESS.2019.2936918](https://doi.org/10.1109/ACCESS.2019.2936918). Índice de Impacto: 4.098
- Fernando León-García, Francisco J. Rodríguez-Lozano, José M. Palomares y Joaquín Olivares *SysGpr: Sistema de Generación de Señales Sintéticas Pseudo-realistas*. *Revista Iberoamericana de Automática e Informática Industrial*, 16(3):369–379, 2019. DOI: [10.4995/riai.2019.10025](https://doi.org/10.4995/riai.2019.10025).
Índice de Impacto: 1.313
- Fernando León-García, José M. Palomares y Joaquín Olivares *D2R-TED: Data-Domain reduction model for threshold-based event detection in sensor networks*. *Sensors*, 18(11):3806–3832, 2018.
DOI: [10.3390/s18113806](https://doi.org/10.3390/s18113806). Índice de Impacto: 3.031
- Fernando León-García, Antonio Cubero-Fernandez, Francisco J. Rodríguez-Lozano, José M. Palomares y Joaquín Olivares *Generación de señales sintéticas para IoT*. *Jornadas SARTECO (JCER), Málaga (España), 20–22 septiembre 2017*. DOI:[10.5281/zenodo.1025964](https://doi.org/10.5281/zenodo.1025964)
- Fernando León-García, José M. Palomares, Joaquín Olivares y Antonio Cubero-Fernandez *Reconocimiento de Patrones para Reglas Distribuidas en Sistemas de Bajas Prestaciones*. *Jornadas SARTECO (JCER), Salamanca (España), 14–16 septiembre 2016*. DOI:[10.5281/zenodo.1025964](https://doi.org/10.5281/zenodo.1025964)

Busca la simplicidad, pero desconfía de ella.

— Alfred North Whitehead

AGRADECIMIENTOS

Quisiera agradecer a todas las personas que me han ayudado en este proyecto personal, por su apoyo y su paciencia, especialmente en los momentos más difíciles.

Un agradecimiento muy especial a mis padres. Como siempre, incluso sin entender muy bien el destino, han adornado mi camino con sus mejores deseos.

Esta Tesis Doctoral no hubiera sido posible sin haber contado con el apoyo personal y profesional de mis directores a los que dedico unas palabras de agradecimiento. A José Manuel, porque sin contar las horas invertidas delante de la pizarra ha logrado convertir ingeniería en ciencia en mi forma de pensar. Y a Joaquín, por su visión pragmática y optimista que hace todo más sencillo. Ambos, que han creído en mí incluso cuando yo no he sido capaz, son más mis amigos que mis directores.

MUCHAS GRACIAS.

LISTA DE CONTENIDOS

i	INTRODUCCIÓN	1
1	INTRODUCCIÓN	3
1.1	Antecedentes en computación ubicua	4
1.2	Antecedentes tecnológicos	4
1.3	Enfoque de la tesis	7
1.4	Estructura de la tesis	9
2	MARCO TEÓRICO	11
2.1	La computación ubicua	11
2.1.1	Consciencia del entorno	12
2.2	Internet de las cosas (IoT)	14
2.2.1	Arquitecturas de IoT	15
2.2.2	Tecnología en el borde de la red: las cosas	17
2.3	Redes de sensores	18
2.3.1	Detección de eventos	18
2.3.2	Reducción de tráfico	20
3	MOTIVACIÓN Y OBJETIVOS	23
3.1	Motivación	24
3.2	Objetivos	25
4	HIPÓTESIS Y METODOLOGÍA	27
ii	APORTACIONES	29
5	DESCRIPCIÓN DE LAS APORTACIONES	31
5.1	Artículo 1. SysGPR: Generador de Señales Pseudo-realísticas	32
5.2	Artículo 2. Modelo D2R-TED	33
5.3	Artículo 3. Algoritmo DRACE	33
6	ARTÍCULO 1: GENERADOR SYSGPR	35
6.1	Abstract	35
6.2	Introducción	36
6.3	Trabajos relacionados	37
6.4	Método	39
6.5	Resultados experimentales	41
6.5.1	Herramienta de generación de señales	41
6.5.2	Muestras generadas	41
6.5.3	Validación	43
6.6	Conclusiones	49
6.7	Trabajo futuro	50
6.8	Apéndice A. Código simplificado del método	50
6.8.1	Función generateSignal()	50
6.8.2	Función generateIncrements()	50
6.9	Apéndice B. Diagramas de clases e instancias	51
7	ARTÍCULO 2: MODELO D2R-TED	59
7.1	Abstract	60
7.2	Introduction	60
7.2.1	Specifying the Scope of Interest	61
7.3	Background	62
7.3.1	Event Detection in WSN	63
7.3.2	Aggregation and Compression Techniques in WSNs	63
7.3.3	Discussion and Hypothesis	65
7.4	Method	66
7.4.1	Data Reduction	67
7.4.2	Metrics	69

7.5	Case Study	74
7.5.1	Case Study I	75
7.5.2	Case Study II	78
7.6	Comparative Test	79
7.6.1	Increasing Linear Threshold 'K'	80
7.6.2	Send-on-Delta and Predictive Sampling	80
7.6.3	Adaptive Versions of SoD and PS	81
7.6.4	Experiment Configuration and Results	82
7.7	Conclusions and Future Work	84
8	ARTÍCULO 3: ALGORITMO DRACE	91
8.1	Abstract	92
8.2	Introduction and Motivation	92
8.2.1	The Problem of a Proper Defense	93
8.2.2	Contextualization	94
8.3	Background	95
8.3.1	Distributed event detection	96
8.3.2	Sampling schemes	97
8.3.3	Traffic aware strategies in WSNs	97
8.3.4	Discussion about the background	98
8.4	Mathematical Approach and Algorithmic Proposal	98
8.4.1	Notation for pseudocode and equations	98
8.4.2	Mathematical approach to The Problem of a Proper Defense	100
8.4.3	The council solution	102
8.4.4	Algorithmic notation	103
8.4.5	Generalisation: DRACE algorithm	106
8.5	Case Study	107
8.6	Experiments	111
8.6.1	DRACE configuration parameters	111
8.6.2	Randomly generated pseudo-realistic signals	111
8.6.3	Metrics	112
8.6.4	Results and discussion	116
8.7	Conclusions	118
8.8	Future Work	119
iii	DISCUSIÓN Y CONCLUSIONES	125
9	DISCUSIÓN	127
9.0.1	Fundamentos matemáticos de la reducción de tráfico: el modelo D2R-TED	128
9.0.2	Aplicación adaptativa de la reducción de tráfico: Algoritmo DRACE	129
9.0.3	Generación de señales pseudo aleatorias: SysGpr	130
10	CONCLUSIONES	131
11	TRABAJO FUTURO	133
12	CONCLUSIONS	135
	BIBLIOGRAFÍA	137

LISTA DE FIGURAS

- Figura 6.1 Diagrama de flujo del método propuesto 39
- Figura 6.2 Interfaz para la generación de señales 41
- Figura 6.3 Generador de repositorios de señales 42
- Figura 6.4 Señal original y señales generadas aleatoriamente utilizando solo el nivel 0 con distribución gaussiana 43
- Figura 6.5 Señal original y señales generadas aleatoriamente utilizando solo el nivel 0 con distribución gamma 43
- Figura 6.6 Señal original y señales generadas con distribución gaussiana de nivel 0 y nivel 1 de tipo gamma 44
- Figura 6.7 Señal original y señales generadas con distribuciones gaussianas en nivel 0 y en nivel 1 44
- Figura 6.8 Diagrama de clases. 52
- Figura 6.9 Diagrama de instancias para dos niveles de generación. 53
- Figura 7.1 Tree structure. 67
- Figura 7.2 Φ domain reduced to a set of $\Phi_{r\delta}$ domains ($j = k = 1$). 68
- Figura 7.3 Example of RGB color domain reduction. Components of color coded with 5, 3 and 1 bit. 70
- Figura 7.4 Upper left: Original signal $F(x)$, in black; and a constant threshold (Threshold), in red. Bottom left: Boolean response of the condition $(F(x) > \text{Threshold})$ applied to the original signal and threshold. Upper right: Reduced signal $F'(x)$, in black; and a constant threshold (Threshold), in red. Bottom left: Boolean response of the condition $(F'(x) > \text{Threshold})$ applied to the reduced signal and threshold. In red, Boolean variations with the obtained in the bottom left image. 72
- Figura 7.5 Illustration of the metrics analysis process. (A) Transition recall analysis with $\tau = 0$. In red, unmatched transitions; (B) Transition recall analysis with $\tau = 1$. In green, matched transitions; (C) Extended transition recall analysis with $\tau = 1$. In green, matched transitions; in red, unmatched transitions; in purple, unmatched samples. 73
- Figura 7.6 Case study II. LP, TP, and TR metrics for training and experimental data. 79
- Figura 7.7 Representation of increasing linear threshold 'K', defined for the temperature signal extracted from the Oropa station. 80
- Figura 7.8 Case study III. With event-based sampling techniques SoD, PS, A-SoD, and A-PS previously detailed, the condition $T_i > K_i$ has been evaluated for each temperature signal of each station of the Arpa Piemonte repository [7]. The resulting Boolean signals are analyzed through metrics described in Section 7.4.2 for comparison purposes. The numeric identifier of each station in the repository is as follows: Oropa 0, Cameri 1, Alessandria 2, Vercelli 3, Pallanza 4, Montaldo Scarampi 5, and Bobes 6. 83

- Figura 8.1 Artistic representation of the Problem of a Proper Defense, designed using Inkarnate® web tool (<http://www.inkarnate.com>). 101
- Figura 8.2 Filling process of the queues. The diagram represents an example of the process of filling 3 queues of 5 elements, corresponding to $\delta = 1$, $\delta = 5$ and $\delta = 10$. From left to right and from top to bottom the incorporation of successive data is illustrated. Green elements belong to a complete queue, red elements belong to an incomplete queue, grey elements symbolize absence of value. The black circle indicates the current δ value. 106
- Figura 8.3 Evolution of the number of soldiers in enemy camps ($AS[e][d] = M_e \cdot d + A_e \cdot \sin(\frac{2\pi}{T_e} \cdot d)$). The total is represented in black, and the threshold of the attack condition, in red. For this configuration, the battle takes place on day 602, in which the total reaches 10009 soldiers. 108
- Figura 8.4 Evolution of the $AS'[e]$ signals, which represent the most up-to-date information known at the castle on a daily basis. These signals are coloured on a scale from green to red, representing the δ value of the corresponding explorer at the time of sending. The total is represented in black and the threshold of the condition in red. As a reference, the original signals are shown in dotted lines. In gray, the cumulative number of pigeons that would have been sent if the DRACE algorithm were not applied. The blue area represents the cumulative number of pigeons sent from explorers, and the purple area the cumulative number of pigeons sent from the castle, both add up to the number of pigeons sent using the DRACE algorithm. 109
- Figura 8.5 Evolution of the defense preparations carried out in the castle. $DD[e]$ signals correspond to the number of defensive devices present in each gate, and evolve according to Algorithm 5. The marks represent the final number of soldiers ($DS[e][X]$) and devices $DD[e][X]$ defending each door. It is important to note that soldiers are not distributed at the gates until the day of battle. The blue line represents the total number of soldiers (TDS) and devices (TDD) available for defence (on both axes). 110
- Figura 8.6 Number of pigeons received from each enemy camp ($AS'[e]$ signals) differentiated according to δ value of the corresponding explorer at the time of sending. The diagram has been generated with the R library *Circlize* [11]. 110
- Figura 8.7 SysGpr configuration for the generation of a repository of 10000 pseudo-realistic signals of 1000 samples in the range 0-100. Each level delimits the configuration parameters of the statistical distributions available to generate increases in the previous level. 112
- Figura 8.8 Set of signals randomly selected from the R repository. 113

- Figura 8.9 Example of execution with $K = 2$ and $Q = 10$. The signals of S are represented by a thin stroke, and the sum, by a thick stroke. The threshold is represented by a red dotted line. B) and A) show the evaluation of the condition with and without DRACE, respectively. 114
- Figura 8.10 Matrix of scatter diagrams with the results of the experimentation. The results of the experiments for increasing Q values are shown from top to bottom. From left to right for increasing K values. In each diagram, 10^3 executions of DRACE are represented by semi-transparent blue marks, proportionally sized to the number of transitions detected in that execution. In red tones, the mean population ($\mu_{x,y}$) and iso-contours of the Gaussian distribution for 1σ , 2σ , and 3σ are represented, covering the 68, 95 and 99 percent of the points, respectively. 117

LISTA DE TABLAS

Tabla 6.1	Resultado algoritmo clasificación C4.5 para las diferentes señales	46
Tabla 6.2	Resultado de análisis de autocorrelaciones cruzadas y correlaciones cruzadas.	47
Tabla 6.3	Tabla de resultados extraídos de MOS. Un nivel modificador (L1), Dos niveles modificadores (L2).	48
Tabla 6.4	Tablas de contingencia de MOS. Un nivel modificador (L1), Dos niveles modificadores (L2). En filas, tipo de señales presentadas a los expertos. En columnas, número de señales según la respuesta de los expertos para cada tipo.	49
Tabla 7.1	Case study I. Traffic Saving Ratio (TSR).	76
Tabla 7.2	Case study I. Level Precision (LP).	76
Tabla 7.3	Case study I. Transition Precision (TP).	77
Tabla 7.4	Case study I. Transition Recall (TR).	77
Tabla 7.5	Case study II. Traffic Saving Ratio (TSR).	79
Tabla 8.1	Summary of notations	104
Tabla 8.2	Parameter values M , A , and T for each AS[e] signal.	108

ACRÓNIMOS

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ARPANET	Advanced Research Projects Agency NETWORK
CC	Communication Compression
CDG	Compressive Data Gathering
CEP	Complex Event Processing
COP	Context Oriented Programming
CPS	Cyber Physical System

CS	Compressive Sensing
CSV	Comma-Separated Values
DA	Data Aggregation
DARPA	Defense Advanced Research Projects Agency
DC	Data Compression
DDASA	Data-Driven Adaptive Sampling Algorithm
DG	Data Gathering
DPM	Data Per Message
EEUU	Estados Unidos
FCC	Federal Communication Commission
FP	False Positives
GPS	Global Positioning System
GPU	Graphical Processing Unit
GSTD	Generate Spacio Temporal Data
ID ₃	Iterative Dichotomiser 3
IEEE	Institute of Electrical and Electronics Engineers
IIC	Industrial Internet Consortium
IOE	Internet Of Everything
IOT	Internet Of Things
ISM	Industrial, Scientific and Medical
ITU	International Telecommunication Union
LZW	Lempel Ziv Welch (coding)
MAC	Medium Access Control
MIT	Massachusetts Institute of Technology
MOS	Mean Opinion Score
NFC	Near Field Communication
OFC	Open Fog Consortium
OSI	Open System Interconnection
PCA	Principal Component Analysis
PMML	Predictive Model Markup Language
PS	Predictive Sampling
PSDG	Parallel Synthetic Data Generation
QoS	Quality of Service
RFID	Radio Frequency Identification
RGB	Red Green Blue
ROC	Receiver Operating Characteristic
SC	Sampling Compression
SDDL	Synthetic Data Description Language
SN	Sensor Network
SoD	Send on Delta
TCP/IP	Transmission Control Protocol / Internet Protocol
TIC	Tecnologías de la Información y la Comunicación

WECA	Wireless Ethernet Compatibility Alliance
WGKS	White Gaussians K-Simplex
WIFI	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSAN	Wireless Sensor Actuator Network
WSN	Wireless Sensor Network
XML	EXtensible Markup Language

Bloque I

INTRODUCCIÓN

INTRODUCCIÓN

ÍNDICE

1.1	Antecedentes en computación ubicua	4
1.2	Antecedentes tecnológicos	4
1.3	Enfoque de la tesis	7
1.4	Estructura de la tesis	9

Las sociedades actuales, inmersas en un proceso de globalización vertebrado en torno a la tecnología, propician entornos humanos cada vez más digitalizados. En pro de la calidad, experiencia y oportunidades vitales; una extensa red de servicios, dispositivos e interfaces se despliega a nuestro alrededor, generando y procesando grandes cantidades de información con el fin último de otorgarnos la capacidad de aprovecharla de formas sin precedentes. Las tecnologías de información y comunicación (TICs), como parte fundamental de su desarrollo, profundizan en su integración en nuestra cotidianidad, reduciendo así las limitaciones del ser humano para dar lugar a una comunidad globalmente conectada, donde el tiempo y la distancia no son un problema.

Esta realidad en constante desarrollo conlleva un gran esfuerzo de investigación en múltiples ámbitos. La terminología científica adoptada es amplia, al tratarse de un paradigma tecnológico aún por definir en muchos aspectos. Desde el punto de vista de la infraestructura, en el ámbito global se adoptan los conceptos *Internet of Things (IoT)*, *Internet of Everything (IoE)*, *Cloud Computing* y *Fog Computing* entre otros; en el ámbito local se desarrollan otros términos como *Cyber Physical System (CPS)*, *Edge Computing* o *[Wireless] Sensor Networks (WWSN)*. Este crisol de conceptos representa un frente de vanguardia tecnológica que, como piezas de un mismo rompecabezas, se integran en la visión de una infraestructura global que todo interconecta.

El camino de datos desde los sensores y terminales hacia los centros de la infraestructura es un proceso en el que la información trasciende a su entorno local para ser parte de una abstracción de mayor perspectiva. En este proceso, el conocimiento es extraído mediante el análisis de grandes volúmenes de datos heterogéneos: *Big Data*; y suministrado por los diferentes servicios y aplicaciones de acuerdo con sus propósitos. Un buen ejemplo de ello son las modernas aplicaciones de navegación, conscientes del estado del tráfico de las vías de circulación en tiempo real.

Según las previsiones de CISCO y otras compañías importantes del sector, el número de dispositivos conectados a Internet crecerá exponencialmente en la próxima década. Disponer de infraestructura de red eficiente para dar servicio al volumen de datos previsto es uno de los grandes desafíos tecnológicos que afronta la comunidad científica. En relación a este tema, se han publicado numerosas propuestas que replantean las redes de comunicaciones a todos los niveles. Una línea de investigación en relación a este problema busca desarrollar modelos de inteligencia para gestionar los recursos de red, dando lugar a infraestructuras adaptativas en pro de la eficiencia. Por otra parte, surge la necesidad de acercar el procesamiento a los generadores de información en la

medida de lo posible, evitando así la centralización del procesamiento en los centros de datos, con el fin de evitar colapsos distribuyendo la carga de la red. Los sensores y nodos terminales juegan un papel fundamental en el desarrollo de estas tendencias de diseño, al disponer de capacidad de cómputo aplicable en las fases más tempranas del camino de datos.

1.1 Antecedentes en computación ubicua

Mark Weiser comenzó su artículo *The computer for the 21st Century*, publicado en 1991, escribiendo: “Las tecnologías más profundas son las que desaparecen. Se entretienen en el tejido de la vida cotidiana hasta que son indistinguibles de ella” [83]. Inspirado en la distopía de ciencia ficción *Ubik*, de Philip K. Dick, en este trabajo sentó las bases de lo que él denominó *Ubiquitous Computing* y más tarde recibió otros nombres como *Pervasive Computing*, *Calm Technology* o *Everyware* [37].

La computación ubicua de Weiser puso sobre la mesa la necesidad de replantear la interfaz hombre-máquina. El desarrollo de la sociedad digital debía construirse desde la simplicidad de lo gestual y lo verbal, acercando los modelos de computación al lenguaje humano, en contraposición a la complejidad característica de los terminales de sistema operativo. Incluso las interfaces basadas en ventanas, concebidas sobre la metáfora espacial de un escritorio, fueron puestas en duda ante la idea de futuros dispositivos sin pantalla o de tamaño reducido.

En cuanto al hardware, las hipótesis originales de la computación ubicua consideran tres clases de dispositivos en relación a su tamaño: *Tags*, de escasos centímetros; *Pads*, del tamaño de una mano; y *Boards*, de incluso metros; compartiendo los tres la característica de ser generalmente planos y con salida visual.

La incuestionable certeza con la que Mark Weiser vaticinó el futuro tecnológico revela la influencia de su trabajo en el desarrollo de las TICs, sentando los principios de diseño que rigen los avances científicos y tecnológicos hoy en día. La computación ubicua establece los cimientos teóricos del Internet de las cosas, si bien no empezó a desarrollarse hasta que el estado de la tecnología lo hizo posible años después.

1.2 Antecedentes tecnológicos

En la actualidad, el concepto *Internet of things* recoge el testigo de la computación ubicua en su dimensión técnica. A lo largo de las últimas décadas, el desarrollo científico en áreas del conocimiento relacionadas con las TICs han permitido dar los primeros pasos hacia la sociedad digital descrita por Mark Weiser. Con el propósito de construir una perspectiva más amplia del presente de este desarrollo, se recogen a continuación una serie de tecnologías que han sido imprescindibles para la aparición de los conceptos en los que se basa esta tesis.

Tecnologías de integración a gran escala (VLSI)

En cuanto a la interfaz hombre-máquina desde el punto de vista del hardware, la constante mejora del proceso de fabricación de circuitos integrados ha dado lugar a la miniaturización de los dispositivos. En 1965, Gordon Moore publicó

un trabajo en el que estudia las tendencias de dos parámetros relacionados con los circuitos integrados: el número de componentes por circuito y el precio relativo por componente [59]; revelando dos marcadas tendencias: la capacidad técnica de integrar cada vez más componentes por chip, y la consecuente reducción del precio relativo por componente. Sus conclusiones dieron lugar a la famosa ley de Moore, que usualmente se interpreta en clave incremental en relación a la capacidad de cómputo, afirmando que *cada 2 años se duplica el número de transistores de los procesadores*. Si bien esta interpretación es cierta, también lo es a la inversa: *cada 2 años se puede reducir a la mitad el tamaño de un procesador*. Esta interpretación en clave de miniaturización de la electrónica no es únicamente una cuestión de espacio. Componentes semiconductores más reducidos implican menores consumos de energía para la conmutación. Este aspecto no ha sido desaprovechado por la industria, que ha construido un nutrido mercado de productos digitales alimentados con baterías.

Una interesante visión del panorama de la computación actual surge de concebir las consecuencias de ambas interpretaciones de la ley de Moore. En una hipotética escala de potencia de cómputo en la que el ordenador personal promedio se sitúa en el centro, la capacidad de fabricación chips más complejos da lugar a la computación de altas prestaciones, potentes procesadores y GPUs organizados en **CLÚSTER** en grandes centros de procesamiento: *la nube*. En el otro extremo de la escala se ubica la electrónica miniaturizada, los dispositivos de bajas prestaciones y escaso tamaño y consumo: *las cosas*. Las propuestas actuales de arquitecturas para la red global de Internet de las cosas integran esta escala, normalmente a tres niveles de computación: *Edge* (los dispositivos miniaturizados), *Fog* (elementos de infraestructura intermedios que permiten acercar el procesamiento al entorno), y *Cloud* (los centros de procesamiento y almacenamiento de grandes volúmenes de información).

GPU: *Procesador gráfico, de las siglas en inglés de Graphical Processing Unit*

CLÚSTER: *Conjunto de procesadores intercomunicados con redes de alta velocidad que se comportan como uno solo*

Internet y WEB

En 1969 tuvo lugar la primera conexión remota entre dos computadores. En el marco del proyecto **ARPANET**, originalmente concebido con propósitos militares, la Universidad de California en Los Ángeles y el Augmentation Research Center, un instituto de investigación de Stanford, intercambiaron el primer paquete de red de la historia. El concepto de computador como interfaz de comunicación se debe a Joseph C. R. Licklider, que en 1968 describió las ideas subyacentes del Internet moderno con asombrosa precisión, esbozando incluso los principios del modelo cliente-servidor [52].

ARPANET: *Advanced Research Projects Agency NETWORK*

ARPANET supone los inicios de la infraestructura, el demostrador funcional que evidencia la factibilidad de las ideas de Licklider. Durante los siguientes años más computadoras se unen a la red, poniendo de manifiesto un problema de escalabilidad que se resolvió con el protocolo **TCP/IP**, desarrollado por Vinton Cerf y Robert E. Kahn en la década de los 70 e incorporado en ARPANET en 1982, del cual toma el nombre *Internet* tras escindir del proyecto a la parte militar, orientando la red a uso exclusivamente civil.

TCP/IP: *Transmission Control Protocol / Internet Protocol*

En sus orígenes, la red ARPANET/Internet es utilizada para el intercambio de información, y con tal propósito se desarrollan el correo electrónico y la transferencia de archivos en sus primeros años. Sin embargo, no es hasta la década de los 90 que se desarrolla el potencial comunicador de Internet mediante el desarrollo de la *World Wide Web* (o simplemente Web) por parte de Tim Berners-Lee [11]. La Web introduce una nueva forma de comunicación a través de Internet al permitir la transferencia de hipertexto desde los servidores de contenidos. Este hipertexto se organiza visualmente en páginas en el

navegador de cualquier terminal que acceda a este servicio.

En un principio, los estándares Web fueron concebidos para desplegar contenidos estáticos accesibles globalmente, lo que dio lugar a la aparición de grandes portales de contenido digital. Más adelante la Web evoluciona, permitiendo a los usuarios participar activamente del proceso de publicación; los blogs y las secciones de comentarios de las páginas webs son herramientas producto de esta evolución. Las actuales redes sociales representan la culminación de este modelo en el que el usuario acapara el foco de interés, suministrando información recibida por los interesados: el modelo de publicación-suscripción.

Desde el comienzo de la Web hasta la actualidad, la constante incorporación de innovaciones técnicas tienen lugar en un proceso evolutivo cuyos hitos clave han sido catalogados mediante una sucesión de *generaciones Web*, siendo la generación *Web 4.0* objeto de actual debate. Si bien no existe un consenso en torno a la definición de la futura Web, se aprecia en la literatura científica la tendencia a compartir los mismos principios propuestos por Mark Weiser para la computación ubicua: interfaces con abstracción suficiente para la interacción humana e inteligencia.

El desarrollo de Internet/Web es un antecedente tecnológico clave, es la herramienta fundamental sobre la que se construyen las propuestas teóricas resumidas en este capítulo. No tiene sentido plantear la digitalización global sin Internet.

Redes inalámbricas de bajo consumo

La utilización del medio inalámbrico para transmitir información se remonta a finales del 1800, cuando se establecen las bases teóricas de la generación, propagación y detección de ondas electromagnéticas de la mano de Heinrich Rudolf Hertz [39]. Sus investigaciones fueron la base para la construcción de las primeras emisoras de radio. Tal fue la capacidad disruptiva del descubrimiento que en la primera década de 1900 ya se establecieron los primeros enlaces de radio entre América y Europa a través del Atlántico.

La transmisión de paquetes digitales mediante ondas electromagnéticas tiene lugar como hito importante en 1970, cuando un equipo de investigación de la Universidad de Hawaii bajo la dirección de Norman Abramson publica los detalles de su sistema experimental ALOHA [3]. El sistema ALOHA es una red de 7 computadores comunicados mediante enlaces de radio, una alternativa inalámbrica para la comunicación digital. El sistema de Abramson incorpora los primeros métodos para resolver desafíos propios del acceso a un medio compartido, que posteriormente se recogen en el nivel **MAC** del modelo **OSI**, revelando un desafío científico que sigue siendo objeto de investigación. Posteriormente ALOHA se conecta a ARPANET, apareciendo así el primer enrutador inalámbrico de la historia de Internet.

En 1985 la **FCC** lleva a cabo la primera regulación del espectro radioeléctrico, en la cual se liberan de licencia ciertas bandas de frecuencia para propósitos industriales, médicos y científicos; las bandas **ISM**. Hasta este momento, los experimentos con tecnologías inalámbricas tuvieron lugar en laboratorios utilizando generalmente altas frecuencias. Especialmente reseñable es el experimento que publicó IBM en 1979 en una de sus fábricas en Suiza, en la que desarrollan una red local mediante tecnología de infrarrojos que consigue velocidades de transmisión considerables [35]. Una vez reguladas las bandas ISM, la comunidad científica y la industria se volcaron en la investigación y desa-

MAC: *Medium Access Control*

OSI: *Open System Interconnection*

FCC: *Federal Communication Commission*

ISM: *Industrial, Scientific and Medical*

rollo de tecnologías inalámbricas. En un principio aparecieron en el mercado soluciones comerciales independientes de la mano de grandes empresas, pero en 1999 se asociaron conformando la alianza **WECA**, que 4 años más tarde pasó a llamarse **Wi-Fi**, con el objetivo de aunar esfuerzos de desarrollo en un estándar para comunicaciones inalámbricas de área local. Esta implicación de la industria en la estandarización mediante grandes consorcios, en contraposición con el modelo de iniciativa únicamente pública como fue el caso de ARPANET/Internet, constituye el modelo actual de estandarización.

WECA: *Wireless Ethernet Compatibility Alliance*

Wi-Fi: *Nombre comercial que proviene de Wireless Fidelity alliance Inc.*

Al margen de los nombres comerciales como Wi-Fi, la organización internacional **IEEE**, líder en estándares de ingeniería eléctrica y electrónica, engloba los principales grupos de trabajo que publican los documentos técnicos subyacentes a las tecnologías de comunicación de área local y metropolitana bajo la numeración 802. Por ejemplo, 802.11 corresponde al estándar de redes locales inalámbricas **WLAN**, correspondiente al consorcio Wi-Fi, del cual hay publicadas 18 versiones.

IEEE: *Institute of Electrical and Electronics Engineers*

WLAN: *Wireless Local Area Network*

En relación al interés de esta tesis, un hito tecnológico importante se dio en el año 1998 con la publicación de la especificación *Bluetooth* por parte de un grupo de empresas encabezadas por Ericsson, quien desarrolló la tecnología inicialmente. Esta publicación inicia el desarrollo de las tecnologías de redes inalámbricas de área personal (**WPAN**), que son el objeto de trabajo del grupo 802.15 del IEEE.

WPAN: *Wireless Personal Area Network*

El grupo de trabajo *IEEE 802.15.4* se encarga de elaborar estándares de red de área personal con un perfil de requerimientos que se ajusta a las necesidades de la electrónica miniaturizada: bajas tasas de datos a cambio de consumos muy reducidos. Este estándar, que define las capas física y de acceso al medio, es la base técnica sobre la que se construyen soluciones comerciales como *ZigBee*, de la mano del consorcio *ZigBee Alliance*, cuya relación con *IEEE 802.15.4* es parecida a la relación entre *Wi-Fi* e *IEEE 802.11*. En la actualidad, está en proceso de desarrollo la versión *IEEE 802.15.4w*, que incluye la posibilidad de establecer enlaces kilométricos mediante una alternativa sub-GHz en capa física manteniendo la máxima del reducido consumo. Si bien existen alternativas que comparten los criterios de consumo del estándar *IEEE 802.15.4*, su publicación puso el foco de interés de la comunidad científica sobre el desarrollo de conceptos como *WSN* e *IoT*. Interés que perdura en la actualidad.

En el entorno cercano, el presente y futuro tecnológico es esencialmente inalámbrico. El ideal de ubicuidad es difícilmente sostenible sin las tecnologías inalámbricas que lo hacen posible; no tanto por las características intrínsecas del medio radioeléctrico, sino por la libertad espacial que otorga la ausencia de infraestructura cableada. El desarrollo de las tecnologías inalámbricas orientadas al bajo consumo y la miniaturización de la electrónica han hecho posible que dispositivos muy pequeños y con escasos recursos de cómputo y energía puedan comunicarse de forma inalámbrica. Es decir, han hecho posible la aparición de *las cosas*.

1.3 Enfoque de la tesis

Los antecedentes teóricos y tecnológicos introducidos ayudan a comprender el origen y la tendencia de la revolución digital. En cuanto al futuro las previsiones son claras: el entramado de dispositivos de uso cotidiano que generan información y requieren de infraestructura para canalizarla va a extenderse penetrando en todos nuestros entornos; va a introducirse en nuestros hogares, industrias, ciudades, vehículos, prendas de ropa, e incluso en nuestros propios

cuerpos. La conjunción de la inteligencia artificial, el Big Data, y la humanización de las interfaces con la tecnología van a producir un cambio en nuestro estilo de vida sin precedentes. Los retos derivados de tal cambio de paradigma requieren de análisis, visión y decisión en todos los aspectos del saber y hacer humano: política, ciencia, ética, sociedad, economía, etc.

Uno de los numerosos frentes de investigación de la dimensión técnica del desafío tiene que ver con el ingente volumen de información que se espera canalizar desde el creciente conglomerado de dispositivos distribuidos en el entorno. La implicación científica en esta problemática abarca todas las ramas del conocimiento relacionadas con las TICs, proponiendo enfoques orientados a las diferentes fases del camino de datos de la infraestructura: captación de datos, compresión, estrategias de envío, protocolos de comunicación, enrutamiento, procesamiento temprano, fusión de información, estrategias de clústering, esquemas de prioridad, etc. Asimismo, la heterogeneidad de las fuentes de información y del propósito de las mismas hace que sea difícil plantear soluciones de propósito general, lo que fomenta la aparición de soluciones de eficiencia para propósitos específicos. Esta extensa riqueza del panorama científico conlleva la imposibilidad de prever el futuro tecnológico con precisión; parece que la futura infraestructura de Internet de las cosas está llamada a hacer frente a la heterogeneidad diversificando su infraestructura intermedia, con pasarelas multiprotocolo con inteligencia para adaptarse al contexto.

Por lo abstracto de su definición, el concepto de adaptación al contexto está sujeto a interpretación. La palabra contexto hace referencia a toda suerte de circunstancia que caracteriza el entorno de un hecho. Es por ello que abordar tal propósito desde el rigor de la ciencia requiere la concreción de los parámetros fundamentales que modelan el supuesto contexto al que la propuesta se refiere. En el caso de las red de cosas, proponemos tres niveles de contexto:

Contexto tecnológico El contexto tecnológico hace referencia a los detalles técnicos de la red objeto de adaptación. En este nivel nos referimos a las características de los dispositivos desplegados, las tecnologías de comunicación empleadas, la topología de la red, la movilidad de los nodos, etc.

Contexto de red El contexto de red tiene que ver con el estado de la misma. Aquí entran en juego los parámetros de calidad de servicio ampliamente explorados por la comunidad científica.

Contexto lógico El contexto lógico está relacionado con el significado y uso de los datos que fluyen en la red. La cuestión del propósito de la información no es accesible si se pretende una estrategia de adaptación de propósito general. Sin embargo, en redes de propósito específico, conocer la importancia y objetivo de cada envío posibilita incorporar esta lógica en los mecanismos de adaptación.

Entendiendo que el contexto es una realidad que evoluciona, la adaptabilidad de un sistema es un proceso que parte como información de entrada la caracterización de éste, y como salida requiere disponer de recursos para actuar en consecuencia, con el objetivo final de optimizar su funcionamiento. Por otra parte, el proceso requiere de cierta inteligencia para gestionar acertadamente diagnósticos, acciones y repercusiones.

Una de las tecnologías clave de este marco tecnológico son las WSNs, parte fundamental de la infraestructura al tratarse de los dispositivos que nutren la red con información extraída del entorno físico. Las WSNs implementan la digitalización de datos y el primer frente de procesamiento de la infraestructura global. Los sensores se caracterizan por ser dispositivos pequeños, baratos, con

escasa capacidad de cómputo y un reducido consumo para poder funcionar con batería. Estas características han servido a los científicos para establecer una nutrida línea de investigación en la que se han propuesto toda suerte de técnicas de colaboración, auto-organización y agrupación entre otras; además de sistemas operativos, librerías y middlewares específicos para estos sistemas. En cuanto a las aplicaciones de las WSNs, una de las más comunes consiste en la detección de eventos, existiendo propuestas desde multitud de enfoques. Un enfoque clásico y muy extendido es el de evento basado en umbral, según el cual el evento tiene lugar cuando una magnitud sobrepasa un determinado umbral.

En este marco conceptual se desarrolla esta tesis doctoral, en la que se abordan cuestiones relacionadas con el envío eficiente de datos generados por sensores, acotando el contexto lógico a la evaluación de eventos basados en umbral mediante expresiones multivariantes. En la investigación descrita en esta tesis se aportan modelos y análisis matemáticos relacionados con el procesamiento y envío de datos, proponiendo técnicas adaptativas que tienen cabida en las fases previas de la obtención de conocimiento de mayor abstracción, basado en el resultado de los eventos multivariantes.

1.4 Estructura de la tesis

El contenido de esta tesis se organiza en tres bloques. El bloque I se divide en 4 capítulos y contiene la introducción (que finaliza en esta descripción de la estructura del documento), la descripción de los antecedentes de este trabajo en un marco teórico (Capítulo 2), la motivación y objetivos de las propuestas desarrolladas (Capítulo 3), y las hipótesis y metodología propuestas para su demostración (Capítulo 4). El bloque II contiene 4 capítulos, el primero es una descripción de las 3 publicaciones aportadas como sustento a esta tesis doctoral (Capítulo 5), mientras que los tres capítulos restantes contienen dichas aportaciones por separado (Capítulos 6, 7 y 8). Finalmente el bloque III concluye con 4 capítulos, abordando la discusión de los resultados obtenidos (Capítulo 9), conclusiones (Capítulo 10) y trabajo futuro (Capítulo 11). El último capítulo presenta las conclusiones del trabajo en inglés (Capítulo 12).

ÍNDICE

2.1	La computación ubicua	11
2.1.1	Consciencia del entorno	12
2.2	Internet de las cosas (IoT)	14
2.2.1	Arquitecturas de IoT	15
2.2.2	Tecnología en el borde de la red: las cosas	17
2.3	Redes de sensores	18
2.3.1	Detección de eventos	18
2.3.2	Reducción de tráfico	20

2.1 La computación ubicua

La computación ubicua tiene su origen en el artículo que publicó Mark Weiser en 1991 [83], considerado como el inicio de este área de investigación, que abarca en la actualidad un amplio abanico de tecnologías. La computación ubicua y otros conceptos que surgieron a raíz de ésta, como *pervasive computing*, *calm technology*, o *everyware*; viene a describir la tercera era de la computación moderna. La primera era se definió por la computadora centralizada, sistemas únicamente al alcance de empresas y organismos que eran compartidos por usuarios especializados (un computador para múltiples usuarios). La segunda era corresponde al concepto de ordenador personal, fundamentalmente perteneciente a una persona (un computador por usuario). La tercera era se caracteriza por la explosión de la computación miniaturizada e interconectada, es la era de los teléfonos y relojes inteligentes, los libros electrónicos, y otros dispositivos con procesador empotrado que utilizamos a diario (múltiples computadores por usuario). Si bien los teléfonos inteligentes se han convertido en el computador más ubicuo de la historia, aún son poco sofisticados en comparación con los propósitos de la computación ubicua, al implementar sistemas operativos de uso similar a los clásicos aunque con interfaces táctiles para adaptarlos al tamaño del dispositivo. Por contra, la computación ubicua pretende que, en lugar de virtualizar el mundo en una abstracción simplificada en pantalla, éste sea revestido con la tecnología necesaria para poder hacer un uso intuitivo de una versión computerizada del mismo. Por tanto, podríamos decir que la tercera era de la computación moderna se encuentra aún en fase de desarrollo [81].

El trabajo de investigación científica en computación ubicua puede categorizarse en tres áreas de interés, que corresponden con las características clave de la propuesta de Weiser [84]: *computación*, relacionada con las arquitecturas software para dar soporte a aplicaciones ubicuas; *invisibilidad*, en la que se desarrollan los modelos de interacción de la tecnología ubicua con el ser humano; y *consciencia del entorno*, centrada en los desafíos tecnológicos de los dispositivos y cómo estos interactúan con el entorno de manera inteligente para hacer realizable el concepto de ubicuidad.

2.1.1 Consciencia del entorno

La inteligencia humana nos ha hecho eficientes a la hora de compartir ideas y entender cómo funciona el mundo. Aunque no seamos conscientes de la complejidad subyacente, tenemos una extraordinaria capacidad de reunir y relacionar información contextual mediante incontables estímulos que dan lugar a nuestra percepción de la realidad. En una conversación entre humanos, además de la información compartida gracias a la riqueza de nuestros lenguajes, nuestro cerebro relaciona también información situacional sin que seamos necesariamente conscientes de ello. Las computadoras empiezan a entender nuestros lenguajes, pero carecen de capacidad de comprensión de la situación, no entienden cómo funciona el mundo, no pueden entender el contexto de la información; al menos, no tan fácilmente como los humanos. Un ordenador personal clásico cuenta con un ratón y un teclado para ser provisto de información mediante interfaces extremadamente limitantes para el usuario, un escenario muy alejado del principio de transparencia de la computación ubicua. Reducir las limitaciones en las comunicaciones humano-computadora es un asunto de actual interés para la ciencia, que aborda el tema desde dos perspectivas fundamentales:

- Mejorar los lenguajes interfaz humano-computadora.
- Incrementar la capacidad de las computadoras de percibir y reaccionar a la información situacional, o contexto.

La primera trata de mejorar la forma en la que el humano interactúa con la máquina, habilitando formas de comunicación más naturales. Bajo este enfoque se desarrollan técnicas como el procesamiento del habla o el análisis gestual, formas explícitas de comunicación. Existe otra información relevante que se comparte de manera implícita en una comunicación humano-humano, como la expresión de las emociones mediante expresiones faciales; eventos pasados y futuros; otras personas presentes en el momento de la conversación; o sus relaciones con los interlocutores. Habilitar esta información para los computadores de forma implícita para enriquecer la comunicación humano-máquina es el objetivo del segundo enfoque. Los investigadores en esta materia tratan de hacer este proceso natural, evitando la necesidad de transmitir este conocimiento mediante interfaces prediseñadas, tratando de abstraer al interlocutor humano del proceso de extracción de información contextual. Esta segunda forma de abordar la manera en que nos comunicamos con los computadores es un desarrollo clave para la computación ubicua, que ofrece al usuario la expectativa de acceso a información y servicios que quiera, donde quiera y cuando quiera [26].

El concepto de computación contextual (*context-aware computing* en inglés) surge a mediados de los años 90, apareciendo por primera vez en la literatura científica en [72], trabajo en el cual los autores consideran el contexto como la localización, la identidad de las personas y objetos cercanos, y los cambios en los mismos. De manera similar pero unos años más tarde, en [17] se define el contexto como la localización, la identidad de las personas cercanas al usuario, la hora, estación del año, temperatura, etc. Posteriormente, [27] enriquece el concepto con aspectos físicos, emocionales, sociales y mentales del usuario; como el foco de atención, estado de ánimo y orientación entre otros. Existen muchas definiciones alternativas de contexto en esa década, esencialmente centradas en el entorno del usuario [16, 31] o en de la aplicación [71, 82]. De los múltiples enfoques que propició el debate científico, es especialmente reseñable la definición propuesta por Abowd et al. [2], en la que se define contexto como *“cualquier información que puede ser utilizada para caracterizar la situación de una entidad. Siendo una entidad una persona, lugar u objeto considerado relevante pa-*

ra la interacción entre un usuario y una aplicación, incluyendo al propio usuario y la propia aplicación". Más recientemente la computación contextual ha adquirido enfoques más pragmáticos. Los interrogantes *cuándo* (tiempo), *dónde* (localización), *quién* (identidad), *qué* (actividad) y *por qué* (uso) vertebran la dinámica de los algoritmos de decisión que se desarrollan en la materia [32].

Los sistemas de computación contextual abarcan una extensa variedad de áreas de aplicación, como sistemas distribuidos, aplicaciones web, aplicaciones móviles, robótica y sistemas críticos entre otras. Algunas de las técnicas más utilizadas para modelar y gestionar el conocimiento del contexto se describen a continuación [54].

Ontologías Algunos enfoques han utilizado ontologías para modelar datos de contexto provenientes de una variedad de fuentes relacionadas, y para mapear y desambiguar conceptos similares teniendo en cuenta la reutilización [18, 63, 66].

Modelos de características Los modelos de características representan *aspectos, calidades o atributos* característicos del sistema y sus dependencias. Estos modelos se han aplicado para representar la variación del conocimiento del contexto [19, 25, 63].

Controladores en tiempo de ejecución La reconfiguración basada en la información de contexto desempeña un papel clave para proporcionar un comportamiento inteligente a los sistemas de computación contextual. Existe trabajo previo en diferentes campos, utilizando datos del contexto como entrada para programas gestores que, en tiempo de ejecución, desencadenan reconfiguraciones de manera oportuna. La toma de decisiones es un proceso complejo que requiere procesar abundante información implícita y explícita [15]. En este marco se desarrollan métodos computacionales eficientes que razonan sobre flujos de datos de sensores en tiempo de ejecución, modelando así el escenario de un contexto cambiante [20, 33, 60].

Lenguajes de programación orientados al contexto El paradigma de la programación orientada al contexto (COP) surgió como una solución de ingeniería del lenguaje para diseñar software sensible a contextos variable. En [6], los autores comparan 11 lenguajes orientados a contexto, señalando diferencias entre sus características y estrategias de implementación. La idea principal detrás de la mayoría de los lenguajes COP consiste en añadir nociones de contexto (a veces llamados capas) como clases principales, y construir sobre ellas ciertas abstracciones de comportamiento prediseñadas para escenarios particulares (típicamente clases o métodos). Algunos ejemplos de propuestas de lenguajes son [23, 36, 40].

COP:
Context-Oriented
Programming

Actualmente, la computación ubicua y, más concretamente, la computación contextual, se desarrollan en el marco tecnológico de internet de las cosas, concepto más reciente que aborda desafíos relacionados con la computación miniaturizada y su interconexión. Muchos de los retos de la computación contextual son herencia directa de las limitaciones intrínsecas a los pequeños dispositivos que componen el universo de las cosas. Estas limitaciones están relacionadas con la escasez de recursos de memoria, capacidad de cómputo y energía del hardware. Así, la capacidad de procesar el contexto espacial en un entorno depende en gran medida de la precisión de las mediciones de los sensores, tanto si se utilizan tecnologías de geolocalización global o cotejando mediciones propias de la señal inalámbrica para mapear posiciones relativas entre los dispositivos. De igual manera, la calidad del modelo de contexto temporal dependerá en gran medida de la capacidad de sincronización de los elementos presentes. En general, la escasez de recursos supone un factor limitante a la hora de implementar mecanismos precisos de medición en todos los ámbitos.

2.2 Internet de las cosas (IoT)

El origen del término se atribuye a Kevin Ashton, pionero de las tecnologías RFID en el instituto tecnológico de Massachusetts (MIT). En 1999 expuso una presentación titulada “*Internet of things*” en la que se vinculaba el uso de motas RFID aplicadas a una cadena de suministro con Internet. En su visión, Ashton describe cómo los computadores serían capaces de coleccionar datos sin asistencia humana para procesar información útil, lo cual sería posible mediante el uso de tecnologías como sensores miniaturizados y RFID, que permitirían a los computadores observar, identificar y comprender cómo funciona el mundo [7]. A principios de los 2000 ya surgieron trabajos científicos haciendo uso del nombre con el que Ashton bautizó lo que hoy en día es un extenso y prolífico tema de investigación. En el 2002, la revista Forbes se hizo eco del concepto publicando un artículo con el mismo nombre [73]. Citando a Ashton, el trabajo consideró IoT como la semilla de un estándar para que los ordenadores entendieran el mundo real, describiendo aplicaciones RFID integradas en los sistemas de inventario y mejorando la experiencia del cliente. Dos años más tarde, el trabajo científico [34] profundiza en una propuesta más completa, analizando los potenciales usos y beneficios de la interconexión de objetos cotidianos mediante una serie de demostradores con sistemas empotrados y sensores. Había nacido un concepto nuevo, cuya idea original no consistía en redefinir Internet, sino extender la interconexión global de redes informáticas a toda suerte de dispositivo, añadiendo las capas de compatibilidad necesarias. Actualmente existen revistas científicas de gran impacto especializadas en esta temática, y un grupo de trabajo del IEEE para la estandarización de la arquitectura de IoT: IEEE P2413 [22].

A diferencia de la computación ubicua, de origen esencialmente teórico, IoT tiene sus raíces en las experiencias empíricas del equipo de Ashton con RFID, caracterizándose desde el principio por integrar enfoques más técnicos. No obstante, ambos conceptos representan paradigmas tecnológicos sujetos a interpretación, cada cual objeto de un nutrido debate científico que, en muchos aspectos, permite intuir ciertas similitudes en sus planteamientos. La distribución de dispositivos autónomos, miniaturizados y capaces de procesar información del entorno de manera inteligente con el fin último de asistir al ser humano es, sin duda, un escenario compartido de ambos conceptos. En opinión del autor de esta tesis doctoral y sin pretender afirmar categóricamente este extremo, IoT recoge el testigo de la dimensión técnica de los desafíos que plantea la computación ubicua, yendo esta última un paso más allá en abstracción al plantearse la manera en que el ser humano y la tecnología deberían interactuar.

IoT es un tema de investigación muy amplio que ha sido abordado desde muy diferentes perspectivas, dando lugar a una extensa literatura científica pretendiendo acotar el concepto y clasificar los diferentes enfoques. En este aspecto es especialmente reseñable el artículo científico [9], en el que los autores proponen tres perspectivas del internet de las cosas que han sido objeto de extenso debate.

Perspectiva de las cosas Orientada a los dispositivos, en la que se engloba la investigación de tecnologías como RFID, NFC, WSN, entre otras.

Perspectiva de internet Orientada a la tecnología de la infraestructura de compatibilidad para la conexión global. En esta perspectiva tienen cabida los desarrollos de estándares como 6LoWPAN[41].

RFID: Radio
Frequency
Identification

MIT: Massachusetts
Institute of
Technology

NFC: Near Field
Communication

WSN: Wireless
Sensor Networks

6LoWPAN: IPv6 over
Low power Wireless
Personal Area
Networks

Perspectiva semántica Orientada a investigar y desarrollar métodos para representar, almacenar, buscar y organizar la ingente cantidad de información generada en el marco de IoT.

2.2.1 Arquitecturas de IoT

La arquitectura de un sistema es la descripción detallada de su organización fundamental, incluyendo sus componentes y la relación que guardan entre sí, así como los principios que gobiernan su diseño y evolución. IoT abarca ámbitos tan diversos que establecer una arquitectura única es un desafío en el que, tanto la industria como la comunidad científica, se encuentran inmersas. En [68], el autor detalla diferentes propuestas de arquitectura aplicadas a escenarios como salud, urbanismo, logística y agricultura entre otros. La lógica intrínseca a una arquitectura para IoT debe considerar aspectos clave que abarcan diferentes niveles de abstracción. A este respecto el autor considera de manera genérica los siguientes bloques funcionales:

- Dispositivo
- Comunicación
- Servicios
- Control
- Seguridad
- Aplicación

Estos bloques funcionales de la arquitectura se diseñan para cubrir características funcionales como:

- Adaptación dinámica
- Auto configuración
- Comunicación multi-protocolo
- Identificación y descubrimiento
- Integración en red de información común
- Conciencia del contexto
- Capacidad de tomar decisiones

El desarrollo de una arquitectura para IoT que integre estas características funcionales requiere diseñar los bloques funcionales en consecuencia, teniendo en cuenta aspectos como el hardware de los dispositivos, los sensores y actuadores a integrar, las tecnologías de comunicación implicadas, los servicios de Internet a acceder y ofrecer, o los propósitos finales del desarrollo.

Una forma bastante extendida de propuesta de arquitectura para IoT consiste en el desarrollo y publicación de software intermedio o middleware que, mediante especificaciones de clases, módulos y librerías, asisten a la implementación de cualquier aplicación bajo los criterios de diseño de la arquitectura en cuestión. Estos códigos se implementan para ciertos hardwares compatibles y utilizando comúnmente sistemas operativos orientados a IoT, y suponen una abstracción del hardware y sistema operativo para el programador. El artículo científico [69] resume y analiza más de 60 middlewares para IoT, aportando un criterio de clasificación según su diseño, diferenciando enfoques basados en eventos, servicios, agentes, máquinas virtuales, espacios de tuplas, bases de datos o aplicaciones específicas; y analizando aspectos funcionales, no funcionales y de la arquitectura. El artículo pone de manifiesto la complejidad del

debate científico, retratando la magnitud de la variedad y cantidad de enfoques que tienen cabida en la materia.

Con enfoque de propósito general, la primera propuesta de arquitectura para IoT se publicó en el informe anual ITU de 2005 [44], en la que se diferencian 3 capas: *Perception* (percepción), *Network* (red) y *Application* (aplicación). La capa de percepción representa la piel del sistema, sus órganos sensoriales. Básicamente se encarga de identificar objetos y transmitir datos. La capa de percepción integra lectores de códigos de barras, etiquetas RFID, antenas GPS, cámaras, y todo dispositivo terminal de red capaz de interactuar con el entorno como sensor o actuador. Si la capa de percepción es la piel, la capa de red es el sistema nervioso, y se encarga de canalizar y procesar los datos recibidos de la capa de percepción. En esta capa tiene lugar la convergencia entre la red de cosas e Internet, considerando servidores de procesamiento inteligente para obtener información. Por último, la capa de aplicación se encarga de elevar la información a conocimiento útil, que es puesto al servicio del ser humano en muy diferentes ámbitos sociales e industriales.

A raíz de esta arquitectura, otros autores han propuesto otros modelos con diferentes niveles. Es el caso de [85], en el que se considera un modelo con 5 capas en el que la capa de red es desdoblada en las capas *Transport* (transporte) y *Processing* (procesamiento), diferenciando claramente las funciones de transmisión y enrutamiento de las de análisis de datos, procesamiento y almacenamiento. También añade la capa *Business* (negocio) sobre aplicación, en la que se lleva a cabo una gestión de las aplicaciones IoT según el criterio de una abstracción superior, un modelo de negocio. Otro ejemplo es el trabajo científico [78], en el que los autores consideran una arquitectura de 7 capas con un enfoque especialmente orientado a la interoperabilidad. Así, entre la capa de red y de aplicación se introducen las capas *Coordination* (coordinación) y *Middleware* (software intermedio), para propiciar la adaptación de los formatos de representación de la información y el hardware. En este modelo, también introducen un bloque de compatibilidad con sistemas existentes en el nivel inferior, y desdoblan la capa de percepción en *Edge technology* (Tecnología de borde) y *Access* (Acceso), diferenciando así la lógica del sensor con los mecanismos de compatibilidad de acceso a la red.

En años recientes el protagonismo de la industria en la definición y estandarización de IoT ha ido tomando relevancia. En 2011 Flavio Bonomi, responsable de investigación de arquitecturas avanzadas de Cisco Systems, introdujo el término *Fog Computing* (computación en la niebla) referido a la necesidad de distribuir geográficamente la carga computacional de los servidores en la nube para satisfacer los requerimientos técnicos de IoT (baja latencia, movilidad, localización en entornos cercanos, etc.). Tal y como el autor lo expresa, *Fog* es esencialmente una nube cercana al suelo, para la cual propone una plataforma de servicios de cómputo, almacenamiento y red entre los dispositivos terminales y la nube [13, 14]. El nuevo concepto introducido por Bonomi tuvo tal repercusión que, en 2015, Cisco Systems, ARM, Dell, Intel, Microsoft y la Universidad de Princeton (Nueva Jersey) fundaron *OpenFog Consortium* (OFC) para desarrollarlo. En 2019 el consorcio pasó a formar parte del *Industrial Internet Consortium* (IIC), tras lograr integrar 57 miembros entre empresas y universidades de todo el mundo. Independientemente, IIC fue fundada en 2014 por AT&T, Cisco, General Electric, IBM e Intel para acelerar el desarrollo y penetración de tecnologías IoT en la industria: *Industrial Internet of Things* (IIoT). Tras la integración de OFC, cuenta actualmente con más de 300 miembros, lo que supone la mayor iniciativa privada relacionada con IoT hasta la fecha. De la documentación técnica publicada por IIC, cabe destacar la arquitectura de

referencia para IIoT [43], en el que se detallan 4 perspectivas: *Implementation* (implementación), *Functional* (funcional), *Usage* (uso) y *Business* (negocio). Bajo la perspectiva de implementación proponen una arquitectura de 3 capas: *Edge* (borde), que integra los detalles de los dispositivos terminales y la red próxima; *Platform* (plataforma), que se encarga de la transformación, análisis y transmisión de los datos; y *Enterprise* (empresa), relacionada con los modelos de negocio de los diferentes ámbitos de aplicación.

El hecho de que la iniciativa privada acelere su implicación en el desarrollo de IoT es una muestra de la madurez del concepto, que adquiere cada vez más precisión en sus definiciones y ámbitos. Si bien actualmente no existe una arquitectura estándar para IoT, es necesario que surja como fruto de la colaboración entre las instituciones de estandarización y los grandes consorcios empresariales, para que la interoperabilidad de los dispositivos en la red de cosas sea efectivamente global, y no se organice en redes autárquicas sujetas a patentes comerciales independientes.

2.2.2 Tecnología en el borde de la red: las cosas

En el extremo de la red, esparcidas en el entorno cercano, se encuentran las cosas. Es en esta parte del macrosistema IoT donde la heterogeneidad, dinamismo y distribución geográfica de los recursos se maximizan, plantenando retos que eran desconocidos cuando se diseñaron las redes informáticas clásicas. Técnicas colaborativas, organizativas, adaptativas, de replicación de información, o de auto-descubrimiento entre otras surgen de estas necesidades. Por esta razón, todas las propuestas de arquitectura para IoT consideran uno o más bloques funcionales específicos para este nivel, aunque con diferentes nombres como *Perception*, *Edge* o *Device* [43, 44, 68, 78, 85]. Actualmente, diferentes enfoques con respecto a las cosas coexisten en el debate científico. Los dos más importantes han dado lugar a dos tópicos ampliamente abordados: *Edge Computing* (computación en el borde), y *[Wireless] Sensor Networks (IWSN)* (Redes de sensores [inalámbricas]).

Edge Computing es el término más reciente, proviene del mundo de los servicios de Internet y está enfocado a la operación y optimización de la infraestructura. *Edge Computing* se origina a principios de los 2000, cuando la empresa Akamai introduce el concepto de *redes de distribución de contenidos* o *CDNs*, cuya finalidad era acelerar el rendimiento de los servicios web mediante dispositivos cercanos al usuario que, a modo de caché, aceleran la carga mejorando la experiencia de uso del navegador [28]. Se puede considerar que *Edge Computing* es la extensión del concepto *Fog Computing* de Bonomi ([13, 14]) aplicado a dispositivos del entorno cercano.

CDN: Content Delivery Network

Las redes de sensores (SNs), por su parte, están más orientadas a la aplicación que al servicio. Su origen se remonta a la década de los 80, en el marco del proyecto *Distributed Sensor Networks* de la agencia de proyectos de investigación avanzada en defensa de EEUU, *DARPA*. El informe final del proyecto, del Instituto Tecnológico de Massachusetts (MIT), aborda experimentalmente el uso de técnicas de procesamiento distribuido para aplicaciones de posicionamiento, seguimiento de objetivo, procesamiento de imágenes para vigilancia o análisis de señales entre otras [76]. A pesar de su antigüedad, el informe describe desafíos tecnológicos de vigente actualidad científica, abordados hoy en día con tecnologías modernas no disponibles entonces, como las comunicaciones inalámbricas de bajo consumo.

DARPA: Defense Advanced Research Projects Agency

2.3 Redes de sensores

El proyecto de DARPA que da origen al concepto de red de sensores tenía el objetivo de desarrollar métodos distribuidos de vigilancia y seguimiento de blancos para sistemas que emplean múltiples recursos de procesamiento y sensorica distribuidos espacialmente [76]. Una aplicación demostrativa del proyecto consistió en un sistema de seguimiento para helicópteros basado en una red de micrófonos [64]. Sin embargo, estos conceptos no fueron abordados por el mundo académico hasta finales de los 90, cuando se creó el grupo de trabajo *IEEE 802.15.4* para crear un estándar de comunicaciones a nivel físico y de acceso al medio para redes personales de baja transferencia de datos y consumo [1]. En 1999 la revista *Business Week* publicó *21 ideas for the 21st century*, en el que esta tecnología fue considerada como una de las más importantes del siglo que comenzaba [77]. La primera versión del estándar de *IEEE 802.15.4* se publicó en 2003, y sentó la base para el desarrollo de protocolos como *ZigBee* [29], *WirelessHART* [21], *6LoWPAN* [41] y *Threads* [75] entre otros. Estos hitos tecnológicos fueron sucediendo en un periodo de investigación con creciente interés por las WSNs, para la que actualmente *IEEE 802.15.4* sigue siendo el marco de desarrollo más importante sobre el que se han implementado sistemas operativos [70], middlewares [80], arquitecturas[24], etc.

Los sensores son dispositivos generalmente pequeños, baratos y muy limitados en cuanto a procesador y energía. A diferencia de las redes informáticas clásicas, las redes de sensores son extremadamente dinámicas, debiendo asegurar la robustez mediante la adaptación, atendiendo a toda suerte de circunstancia debida a factores como por ejemplo la movilidad de los sensores, su escasez de recursos o eventos externos que incrementen inesperadamente el tráfico de datos. Estas circunstancias han hecho que las redes de sensores hayan supuesto un idóneo laboratorio de investigación para técnicas de procesamiento distribuido, como por ejemplo agrupación en clústeres [4], auto organización [55], auto optimización [79] o colaboración [51] entre otras. La complejidad y diversidad de cuestiones que hay que abordar a la hora de diseñar una red de sensores, como por ejemplo el hardware, sistema operativo, arquitectura software, modelo de datos, aplicación o topología; da lugar a una extensa variedad de tipos de WSNs, que pueden ser clasificadas mediante numerosos criterios [61].

Las redes de sensores son utilizadas en entornos domésticos, urbanos, industriales, médicos, militares y otros; principalmente con dos propósitos: recolectar datos del entorno y detectar eventos. En ambos casos, dadas las limitaciones de los sensores y de los protocolos de comunicaciones empleados, se aplican técnicas de optimización para reducir ahorrar energía y reducir el tráfico de red (ambos objetivos estrechamente relacionados). En las siguientes subsecciones se aborda la detección de eventos y las técnicas de reducción de tráfico, ambas en el foco de interés de esta tesis doctoral.

2.3.1 Detección de eventos

En términos generales, un evento es un suceso significativo que es inusual en relación a la normalidad [5]. Al encontrarse esparcida por el entorno, una red de sensores supone una herramienta idónea para monitorizar ciertos eventos cuya ocurrencia conlleva efectos no limitados a una ubicación concreta. Con las técnicas adecuadas, la computación distribuida habilita modelos de detección de patrones distribuidos. La red de sensores, como una piel que reviste el entorno, colabora en pro de un propósito compartido combinando los datos de cada elemento para nutrir un modelo común que a menudo involucra espacio

y tiempo en la síntesis de sus resultados.

La detección de eventos es un tema amplio que requiere abordar algunos aspectos que, independientemente de la variedad de las propuestas, son transversales e intrínsecos al problema. Estos aspectos son generalidades extraídas del análisis de años de aportaciones en la materia [48]:

Dependencia del contexto En una aplicación de detección de eventos los parámetros, variables y métricas involucrados en el problema conforman un modelo tan específico que difícilmente puede ser aplicado en otro contexto. Este aspecto complica la síntesis de soluciones generales.

Críticidad de la aplicación A menudo, la detección de un evento está sujeta a los requerimientos de una aplicación crítica. La monitorización de constantes vitales de pacientes, o la detección de objetos inesperados en el perímetro de seguridad de un vehículo son ejemplos de aplicaciones críticas, en las que el evento en cuestión debe ser detectado a tiempo para abordar las contramedidas oportunas. En estos escenarios, el método de detección debe responder con precisión y rapidez para identificar eventos sin errores y en cortos periodos de tiempo.

Fuentes de datos numerosas y diversas Un evento puede considerarse datos de diferentes fuentes y con diferente estructura, como por ejemplo documentos de texto, imágenes, audio, registros compuestos o datos espacio-temporales. El problema de la detección del evento adquiere complejidad de esta heterogeneidad al tener que determinar qué datos son realmente relevantes, además de administrar los procedimientos adecuados a cada tipo de información.

Topología de la red La cuestión de la topología es especialmente importante cuando el evento en cuestión relaciona datos espacio-temporales y los nodos tienen movilidad. En una red de sensores móviles la conectividad varía constantemente, y con ella la topología del despliegue. Esta situación requiere de métodos adaptativos de ajuste para el enrutamiento.

Algoritmo de detección El método escogido para la detección y el consecuente algoritmo deben abordar todos los aspectos mencionados anteriormente entre otros. Este proceso determinará aspectos de la detección que deben ajustarse a los requisitos del problema, como la tolerancia a errores y retrasos, o la disponibilidad de datos o periodos de entrenamiento. En un sistema de detección de eventos los errores pueden ser de dos tipos: falsos positivos y falsos negativos. La relevancia de estos no siempre es igual, y este hecho abre la puerta a calibraciones iniciales muy específicas para el escenario en cuestión.

De todas las cuestiones a abordar descritas anteriormente, el algoritmo de detección es la más importante, al integrar en su implementación todas las decisiones tomadas para el escenario de aplicación. La lógica del modelo matemático subyacente para detectar eventos se suele clasificar en los siguientes tipos de métodos [65]:

Métodos estadísticos A partir de muestras, la estadística permite analizar correlaciones y dependencias que suponen conocimiento útil a la hora de detectar anomalías o patrones en las señales de los sensores. Combinaciones de técnicas simples como regresiones lineales o medias móviles pueden sustentar un modelo sólido para muchas aplicaciones [38, 47, 88].

Métodos probabilísticos Cuando la ocurrencia del evento depende de sucesos de naturaleza estocástica pueden ser modelados en términos de probabilidad. En redes de sensores con redundancia de información, los métodos probabilísticos pueden ser eficaces a la hora de predecir comportamientos emergentes a nivel de sistema [42, 62].

Métodos basados en IA y aprendizaje automático Los algoritmos de inteligencia artificial computan flujos de información para extraer relaciones y dependencias profundas, difíciles de advertir con percepción humana. Cuando se dispone de capacidad de cómputo suficiente estas técnicas pueden resultar útiles para detectar eventos, especialmente cuando se carece de un modelo eficiente de lo que se busca detectar [10, 53, 67].

Cuando el sistema conoce a priori el evento a detectar los métodos estadísticos son los más utilizados. En este contexto podemos encontrar soluciones algorítmicas que utilizan umbrales [12, 49], lógica difusa [45, 46] o detección de patrones [87, 90] entre otras.

La detección de eventos basada en umbrales se ha utilizado desde los primeros sistemas declarativos, que procesaban variables booleanas conocidas como *hechos* como entrada de sistemas expertos basados en reglas [4]. La detección de eventos basados en umbral es considerada la forma más básica de detección, aunque también la más extendida [48]. En su forma más simple, se notifica el evento cuando una variable relevante traspasa un valor o rango conocido. Sin embargo, este procedimiento puede adquirir complejidad al introducir umbrales dinámicos, introducir expresiones booleanas más complejas o requerir de cierta colaboración entre los sensores. Los eventos basados en umbral son particularmente adecuados para aplicaciones industriales y control de alarmas, como por ejemplo detección de incendios, detección de inundaciones u otras aplicaciones en las que un sensor puede detectar un límite crítico del valor medido [14].

2.3.2 Reducción de tráfico

Uno de los desafíos más importantes de las redes de sensores es el aprovechamiento máximo de sus recursos energéticos. Las tecnologías de comunicación que hicieron posible esta tecnología fueron diseñadas con la perspectiva de un consumo muy reducido [1, 21, 29, 41, 75, 77], ya que los procesos de la interfaz inalámbrica existentes eran energéticamente inasumibles para dispositivos miniaturizados.

Para reducir el tráfico es necesario evitar enviar paquetes de red en la medida de lo posible. Este planteamiento conlleva inevitablemente que la información se vea reducida o comprimida en algún aspecto. Poniendo en perspectiva una red de sensores como un sistema, el camino de los datos comienza en los módulos de adquisición de los sensores. Cada sensor controla sus módulos de adquisición y los primeros procesos sobre los datos digitales extraídos localmente, así como las funciones de acceso a la red. Más allá de la interfaz local de red tienen lugar otros procesos de lógica distribuida, como fusión de información, estadísticas de consenso, establecimiento de prioridades, etc. Finalmente el trabajo colaborativo del despliegue se materializa en un resultado común, que suele ser enviado fuera de la red o centralizado en un nodo coordinador. Salvando la generalidad de esta descripción, en este camino de datos hay multitud de procesos y fases en las que la información puede ser reducida, retenida o descartada. En [30] se profundiza este hecho, aportando una interesante clasificación que agrupa las técnicas de compresión en tres tipos: compresión de muestreo, compresión de datos y compresión de comunicaciones.

Compresión de muestreo En inglés *Sampling Compression (SC)*, se refiere a aquellas técnicas que reducen el número de lecturas de una señal para su digitalización. Una de las técnicas más representativas de este tipo es el denominado *Compressive Sensing*, un método que permite extraer la

información de una señal mediante muestreo de frecuencia inferior a la establecida por el teorema de Shannon/Nyquist [30]

Compresión de datos También llamada *Data Compression (DC)*, engloba a aquellas técnicas que convierten flujos de datos de entrada en otros que requieren menos bits para su representación. Un ejemplo de este tipo de compresión es el uso del análisis de componentes principales (PCA por sus siglas en inglés), que aprovecha la dispersión de las muestras de una señal para transformar el espacio de representación de la información de manera que se pueda codificar con pocos valores representativos [50, 86].

PCA: *Principal Component Analysis*

Compresión de comunicaciones En la literatura científica es conocida por *Communication Compression (CC)*, se refiere a aquellas técnicas que reducen el tiempo de uso de red de los nodos. Este tipo de compresión suele plantearse desde enfoques más próximos a la infraestructura, a menudo relacionados con la gestión del enrutamiento. Ejemplos de compresión de comunicaciones son las técnicas de control de congestión, una nutrida línea de investigación que desarrolla métodos adaptativos que evitan el colapso de las redes [74].

Aunque únicamente el último tipo de compresión hace alusión directa al uso de red, todos los tipos descritos tienen influencia en el volumen de tráfico, ya que existe una relación jerárquica entre ellos. Así, una reducción de muestras induce una reducción de número de bits, lo que a su vez requiere menos paquetes de datos para su envío.

Un importante aporte a la reducción del tráfico en las redes de sensores viene de la disciplina del control automático, donde se estudian los criterios de estabilidad de un sistema de control en relación al régimen de envío de muestras de una señal. En contraposición al clásico envío periódico de información, *Lebesgue sampling* [8] o *Magnitude-based sampling* [56] son denominaciones que hacen referencia al mismo enfoque: estrategias de envío de muestras basadas en un criterio relacionado con la magnitud de la señal, en lugar del tiempo. Existe una extensa literatura al respecto, con propuestas que integran métodos predictivos para minimizar el envío de información necesaria [58, 89].

ÍNDICE

3.1	Motivación	24
3.2	Objetivos	25

En años recientes hemos asistido a la eclosión de la tercera era de la informática. Desde la semilla conceptual de la computación ubicua hasta las actuales propuestas arquitectónicas para Internet de las cosas, un crisol de técnicas y conceptos surgidos desde diferentes ámbitos de la tecnología se combinan para dar forma a las visiones de un futuro ya palpable. La reciente pero firme implicación de la industria es un claro indicio de madurez del desarrollo, camino de la estandarización de los servicios y productos que están llamados a implementar esta realidad. Por otra parte, la comunidad científica ha participado muy activamente en el desarrollo desde el inicio de las ideas, diversificada en múltiples áreas de investigación que han aportado la perspectiva de su ámbito de estudio. Con el propósito de precisar y acotar el ámbito de las aportaciones de esta tesis doctoral, este capítulo aborda la motivación y los objetivos que la sustentan.

3.1 Motivación

La dimensión técnica de IoT enfrenta los problemas derivados de extender la red global a un gran número de dispositivos. Ante este desafío, la máxima de la optimización se repite como un mantra en la literatura científica, independientemente del nivel de abstracción, aplicación o enfoque. Por una parte, para optimizar la red en el entorno cercano se trata de minimizar los accesos al medio inalámbrico en la medida de lo posible. Por otra parte, evitar el colapso de los servidores requiere deslocalizar el procesamiento, acercar el servicio a las fuentes de datos distribuyendo así geográficamente la carga computacional. En ambos casos se requiere computar cierta inteligencia que permita a la red incorporar modelos de optimización adaptativa: cierta consciencia del contexto de red.

En cuanto al entorno local, las redes inalámbricas de sensores son el escenario en el que se desarrolla el procesamiento temprano, fuentes de datos con escasa memoria y capacidad de cómputo interconectadas mediante tecnologías de ancho de banda limitado. Un enfoque predominante aquí es la coordinación de estrategias sensibles a los recursos de red. Generalmente, la literatura científica recoge propuestas de optimización del tráfico basadas en el principio de red como servicio, dando como resultado técnicas aplicables en un amplio rango de casos de uso, sin tener en cuenta el propósito del sistema: el contexto lógico.

Una de las aplicaciones más comunes de las redes inalámbricas de sensores es la detección de eventos; y, de entre las técnicas utilizadas, la más extendida es la detección de eventos basados en umbral. Los eventos basados en umbral computan la ocurrencia de una expresión booleana que generalmente es simple, una variable cuyo valor sobrepasa unos límites preestablecidos. Esta manera de procesar eventos es aplicable en procesos industriales, sistemas de alarma (detección de intrusos, incendios, etc.), domótica, y en general toda circunstancia cuya lógica pueda modelarse mediante rangos de variables. Los eventos basados en umbral siguen siendo de interés para la computación distribuida, aunque la lógica subyacente sigue siendo la evaluación en el sensor de una inequación simple, cuyo resultado es posteriormente la entidad de información que se pone a disposición de una lógica distribuida de mayor abstracción. Así, podemos encontrar propuestas en las que los eventos basados en umbral son consensuados en redes de sensores donde existe redundancia en su evaluación, o donde un coordinador de cluster establece umbrales dinámicos para los modelos de evaluación de los sensores subordinados.

Un enfoque diferente de evento basado en umbral parte de la idea de expresiones multivariante cuyas variables están distribuidas en la red. Este concepto habilita el despliegue de redes en las que un mismo nodo participa de la evaluación de diferentes eventos, bien sea por cuestiones de optimización, precio, o por la distribución de las fuentes de datos en el entorno. Concretamente para estos escenarios no se tiene constancia de literatura científica que analice las potenciales optimizaciones aprovechando las particularidades de la aplicación.

3.2 Objetivos

En este marco conceptual se plantea la investigación de esta tesis doctoral, que pretende profundizar en la optimización del tráfico en redes de sensores que evalúan eventos basados en umbral de expresión multivariante, centrándose en el siguiente objetivo principal.

- Analizar y desarrollar la computación del contexto lógico de la evaluación de condiciones multivariantes en red para la optimización del tráfico.

De este objetivo principal se derivan los siguientes objetivos secundarios.

- Establecer un modelo matemático representativo del contexto lógico de la evaluación de condiciones multivariantes distribuidas, centrado en el uso de la red.
- Proporcionar métricas que permitan evaluar el desempeño del modelo con diferentes parametrizaciones.
- Desarrollar una técnica adaptativa basada en el modelo propuesto para reducir el tráfico en redes de detección de eventos basados en umbral de condición multivariante, maximizando la reducción de tráfico mientras se minimizan los errores en el resultado booleano de respuesta del sistema.

Conocer el estado de una condición que depende de variables distribuidas en red requiere conocer el estado de esas variables en el instante de evaluación. Suponiendo que la sincronización no es un problema, la solución más simple es plantear esta aplicación desde un enfoque síncrono, funcionando la red como un proceso iterativo en el que cada iteración supone una consulta remota de cada fuente de datos. Si bien esta estrategia de muestreo periódico es sencilla de implementar, el volumen de tráfico de red necesario está estrechamente vinculado con la criticidad temporal de la aplicación para la que se evalúa la condición. En otras palabras, si la detección de un cambio de estado debe ser detectada antes de que transcurra un determinado intervalo de tiempo, la frecuencia de muestreo debe incrementarse de acuerdo con este requerimiento, lo que aumenta el volumen de datos enviados proporcionalmente. Este hecho parece invalidar la solución propuesta para aplicaciones críticas, sin embargo cabe cuestionar la utilidad de enviar información desde todas las fuentes en todos los periodos de evaluación, es decir, si el envío de información de cada sensor puede estar vinculado a otros criterios además del temporal.

En la disciplina de control automático analizan los criterios de estabilidad y convergencia de los algoritmos de control cuando el envío de señales se vincula a criterios de magnitud. Esta interesante línea de investigación se conoce por diferentes nombres, como muestreo aperiódico, muestreo basado en eventos o muestreo basado en magnitud, y tiene como objetivo evitar el envío de datos irrelevantes. Aunque aspectos como la estabilidad o la convergencia no tienen traslación directa al problema que motiva esta tesis doctoral, el concepto subyacente puede ser aplicado y analizado en el escenario propuesto.

Por otra parte, como resultado final de la evaluación de una condición se obtiene un valor que representa si la condición se cumple o no. Matemáticamente, una condición de expresión multivariante combina una serie de valores definidos en su dominio matemático para obtener un resultado en el dominio Booleano. Este cambio de dominio en la relación inyectiva es una particularidad interesante de la evaluación de condiciones que puede ser tenida en cuenta para el criterio de envío de señales. Para evitar tráfico innecesario se puede vincular el criterio de magnitud al resultado Booleano del sistema. De esta manera se condiciona la importancia de un dato con su potencial efecto de hacer conmutar el estado de la condición.

Estas ideas fundamentan el enfoque con el que se aborda esta tesis doctoral. El trabajo de investigación realizado se construye sobre la lógica de combinar ambos conceptos para lograr los objetivos propuestos, partiendo de las siguientes hipótesis.

***Hipótesis 1** En la evaluación de una condición de expresión multivariante cuyas variables son sometidas a un muestreo basado en la diferencia con la muestra anterior, se puede establecer un modelo que relacione la magnitud de la diferencia entre muestras con el error introducido en términos de precisión y exhaustividad de la detección de conmutaciones en la señal Booleana resultante.*

Hipótesis 2 Conociendo este modelo, se puede diseñar un algoritmo que ajuste la magnitud diferencial de muestreo para reducir sensiblemente el número de muestras necesarias para evaluar correctamente el estado de la condición.

Para la demostración de estas hipótesis se proponen los siguientes objetivos metodológicos:

1. Realizar un estudio del estado de la vanguardia en tres líneas de investigación en el campo de IoT: a) detección de eventos en redes de sensores, b) técnicas de reducción de tráfico en redes de sensores, y c) técnicas de compresión de información en sistemas distribuidos.
2. Proponer unos fundamentos matemáticos que modelen: a) el escenario de una red de sensores que evalúan eventos basados en umbral de expresión multivariante, b) un esquema de reducción de información basado en técnicas de muestreo basado en magnitud, y c) métricas que cuantifican la reducción de tráfico conseguida y la calidad del resultado en términos de precisión y exhaustividad.
3. Implementar modelo y métricas para construir un simulador específico para el escenario en cuestión. Este simulador permitirá experimentar con volúmenes de datos extensos almacenados en ficheros.
4. Validar el modelo mediante una extensa experimentación que determine el impacto de los parámetros en el resultado, evaluado con las métricas propuestas. Esta prueba de concepto estará orientada a un escenario de propósito general, utilizando ficheros de datos tanto de señales reales como generadas computacionalmente.
5. Comparar el desempeño del modelo con otras técnicas existentes que puedan ser aplicadas al escenario simulado en base a las métricas implementadas.
6. Para disponer de conjuntos de datos de calidad generados computacionalmente, en el marco de la metodología se requiere el diseño e implementación de un software generador de señales, con el que obtener grandes volúmenes de datos de calidad que sean adecuados para sustituir a señales obtenidas de sensores reales en experimentación.
7. Con el conocimiento obtenido de la experimentación con el modelo propuesto, diseñar un algoritmo adaptativo que establezca los parámetros del modelo en tiempo de ejecución, con el objetivo de vincular la reducción de datos a la calidad de la detección del evento, tratando de maximizar ambos extremos.
8. Someter al algoritmo a experimentación utilizando un extenso y variado conjunto de datos creados artificialmente con el software generador implementado.
9. Evaluar los resultados experimentales para comprobar el cumplimiento de las hipótesis de partida.

La realización de los hitos metodológicos descritos ha dado lugar a tres aportaciones científicas publicadas en revistas indexadas que se describen y recogen en el siguiente bloque de esta tesis.

Bloque II

APORTACIONES

DESCRIPCIÓN DE LAS APORTACIONES

ÍNDICE

5.1	Artículo 1. SysGPR: Generador de Señales Pseudo-realísticas	32
5.2	Artículo 2. Modelo D2R-TED	33
5.3	Artículo 3. Algoritmo DRACE	33

Este bloque compila el trabajo científico realizado para dar cobertura a la metodología propuesta. Los resultados de esta investigación han sido publicados en tres artículos que, puestos en común, recogen en sus conclusiones la demostración de las hipótesis planteadas. Como introducción a esta parte, se describe a continuación el contenido de cada artículo y su implicación en la consecución de los objetivos generales y metodológicos de esta tesis doctoral.

5.1 Artículo 1. *SysGpr: System of Generation of Pseudo-realistic Synthetic Signals*

SysGpr es un software generador de señales aleatorias pseudo-realistas que se detalla en este trabajo. Su nombre proviene de su descripción en inglés y título del artículo: *SYStem of Generation of Pseudo-Realistic syntetic signals*, y su desarrollo cubre el objetivo metodológico 6.

Las metodologías de investigación con redes de sensores utilizan a menudo entornos de simulación. Hay varias razones que pueden justificar este hecho, el despliegue de redes reales puede ser temporal y/o económicamente costoso, incluso puede ser tecnológicamente inasumible si se trata del desarrollo de algoritmos o técnicas sobre supuestos de despliegue desafiantes para la tecnología disponible, como por ejemplo una sincronización ideal en topologías multisalto, o despliegues super masivos que cubren enormes áreas geográficas. En estos casos, la simulación ofrece un entorno virtual del que extraer resultados sólidos que validen una propuesta.

En estos casos, nutrir el sistema de simulación de datos para los sensores virtualizados puede ser un problema por diferentes circunstancias, como por ejemplo:

- Si se requiere que el escenario de simulación contemple largos periodos de ejecución, para los que se necesita una cantidad proporcional de datos.
- Si la simple generación aleatoria no es admisible, porque se necesiten señales continuas con un comportamiento coherente.
- Si se requieren datos en un formato específico, sin ausencia de información o de una longitud determinada.

Existen bancos de datos reales y accesibles que, procesados adecuadamente, pueden ser utilizados para este propósito. Sin embargo, a menudo la búsqueda, selección y preprocesamiento de esta información se convierte en una actividad tediosa que requiere mucho tiempo. Como propuesta de solución a este problema se desarrolla *SysGpr*, un software que genera señales aleatorias continuas que presentan patrones de variabilidad coherentes con lo observable en datos de sensores reales.

El algoritmo subyacente utiliza distribuciones de probabilidad Gaussiana y Gamma, en un proceso recursivo que construye las señales muestra a muestra, variando la parametrización de estas distribuciones en cada iteración. La lógica de este proceso está planteada como una jerarquía de bloques anidados, donde cada bloque genera una señal aleatoria cuyas muestras son alteraciones a los parámetros del bloque superior. De esta forma, el incremento en magnitud de una muestra generada con respecto a la anterior obedece a una distribución de probabilidad cuyos parámetros son variaciones de los implicados en el proceso de la iteración anterior. Como resultado se obtienen señales continuas que, analizadas estadísticamente, no siguen una distribución estadística ideal sino ajustes parciales, tal y como las señales del mundo real.

Esta técnica ha sido validada mediante tres métodos independientes: un algoritmo de aprendizaje automático, demostrándose incapaz de clasificar señales reales y generadas; correlaciones cruzadas, obteniendo una correlación elevada entre señales reales y generadas; y encuestas, en la que observadores humanos no han sido capaces de diferenciar señales reales de generadas.

5.2 Artículo 2. *D2R-TED: Data-Domain Reduction Model for Threshold-Based Event Detection in Sensor Networks*

Este trabajo cubre los objetivos metodológicos 1, 2, 3, 4, 5 y 9; sentando los fundamentos matemáticos que modelan la reducción de envíos en una red de sensores para la evaluación de condiciones multivariantes mediante técnicas de muestreo basado en magnitud. También propone tres métricas que evalúan la reducción de tráfico, y la precisión y exhaustividad de las conmutaciones de la señal booleana resultante.

Como se describe en el bloque anterior, las expresiones condicionales multivariantes conllevan un cambio de dominio de la información de entrada, ofreciendo un resultado en el dominio Booleano. Esta transición de dominios matemáticos divide conceptualmente la propuesta en dos partes: a) la técnica de reducción de datos en el dominio de entrada, y b) la evaluación del resultado en el dominio de salida (Booleano).

El modelo proporciona esencialmente dos cosas: una técnica paramétrica para reducir envíos en la red de sensores, y una forma de relacionar la reducción de información de entrada con la calidad del resultado de la evaluación. La reducción de envíos se fundamenta en la técnica de muestreo aperiódico *Send-on-Delta*[57], un criterio de muestreo basado en la diferencia en magnitud con respecto a la muestra anterior. El trabajo tiene dos aportaciones originales en sus planteamientos:

- El modelo matemático está orientado al dominio de la información, lo cual lo hace aplicable a escenarios en los que la información de entrada se define en estructuras de datos abstractas, no únicamente valores reales.
- La lógica de las métricas está centrada en las transiciones del estado de la señal, no en su nivel lógico. Esto permite introducir un parámetro que modela la tolerancia al desfase temporal en la detección del evento, aportando riqueza a la interpretación del desempeño del sistema.

El artículo incluye un análisis de la influencia de la variación de sus parámetros en el resultado utilizando un conjunto de datos de sensores reales. Aunque en la primera versión remitida a la revista se replicaba el estudio sobre señales generadas, en su versión final este análisis no se presenta por motivos de espacio. Finalmente incluye un estudio comparativo con otras técnicas aplicables al escenario de experimentación. Los resultados de esta experimentación demuestran la primera hipótesis planteada en esta tesis doctoral.

5.3 Artículo 3. *Data Communication Optimization for the Evaluation of Multivariate Conditions in Distributed Scenarios*

Este trabajo tiene un objetivo muy concreto: desarrollar un algoritmo que aplique el modelo D2R-TED ajustando sus parámetros en tiempo de ejecución para conseguir evitar el máximo volumen de tráfico posible tratando de no alterar el resultado de la evaluación de condiciones multivariantes en redes de sensores. El algoritmo se llama DRACE por las siglas en inglés de *Data Retaining Algorithm for Condition Evaluation*, y su desarrollo cubre los objetivos metodológicos del 7 al 9, planteados para demostrar la segunda hipótesis de esta tesis doctoral.

DRACE es descrito sobre la base de un símil bélico que, en un contexto lógico, integra todos los aspectos relevantes del envío de información para la

evaluación de una condición que depende de diferentes fuentes. Con este enfoque se presenta el problema del precio de la información, que en esencia es el necesario balance entre el tráfico generado y la precisión temporal de la evaluación de la condición. Como solución de compromiso, DRACE define una serie de procedimientos sujetos a parámetros que inciden en el equilibrio entre ambos conceptos, como se demuestra en la extensa experimentación presentada en el artículo.

El método de reducción y ajuste dinámico de la misma están basados en el modelo D2R-TED con señales de entrada definidas en el dominio de los números reales. La lógica del algoritmo se basa en aprovechar oportunamente los periodos temporales en los que una señal tiene poca influencia en el estado de la condición. Estas ventanas de oportunidad, detectadas mediante un método predictivo, son utilizadas para modificar el criterio de muestreo basado en magnitud de la señal, lo que reduce el tráfico consecuentemente.

La experimentación utiliza un conjunto de 3000 señales generadas con SysGpr, iterando 9000 ejecuciones sobre combinaciones aleatorias de estas señales para dar lugar a un análisis sobre 9 combinaciones paramétricas que representan distintos puntos de equilibrio entre ahorro de tráfico y precisión. Los prometedores resultados del trabajo demuestran satisfactoriamente la segunda hipótesis planteada.

ÍNDICE

6.1	Abstract	35
6.2	Introducción	36
6.3	Trabajos relacionados	37
6.4	Método	39
6.5	Resultados experimentales	41
6.5.1	Herramienta de generación de señales	41
6.5.2	Muestras generadas	41
6.5.3	Validación	43
6.6	Conclusiones	49
6.7	Trabajo futuro	50
6.8	Apéndice A. Código simplificado del método	50
6.8.1	Función generateSignal()	50
6.8.2	Función generateIncrements()	50
6.9	Apéndice B. Diagramas de clases e instancias	51

SysGpr: System of Generation of Pseudo-realistic Synthetic Signals

F. León, Fco. J. Rodríguez-Lozano, A. Cubero-Fernández, José M. Palomares, J. Olivares

6.1 Abstract

Signals obtained from sensors are widely used in different scientific fields. However, the resources to obtain the data are not always available due to structural constraints, physical, economic, environmental, and data collection failures, etc. It is in this scenario that the generation of synthetic data is established. The generation of synthetic data has several benefits, such as, reducing waiting times compared to the long periods required by some sensors to obtain large volumes of samples. In addition, the generated data can be as robust as users need it to be. For this reason, this paper presents a pseudo-realistic synthetic signal generation system for use in the validation of methods and design of experiments. The proposed signal generation method makes use of statistical models and the gradient of the signal to generate new data.

6.2 Introducción

En las última décadas se ha podido observar un gran avance de la tecnología y un aumento de la capacidad de cómputo así como la miniaturización de los sensores [2, 7, 8, 28]. Todo esto ha dado como resultado el nacimiento del paradigma del Internet de las Cosas (IoT) [1]. El gran auge del IoT ha hecho posible que hoy día se desplieguen redes de sensores donde antes era inviable [23].

Sin embargo, el hecho de que hoy día se disponga de enormes cantidades de sensores y datos asociados a los mismos, no supone una ventaja en determinadas situaciones. En especial, los investigadores encuentran limitaciones a la hora de testear sus métodos debido a que en muchas ocasiones necesitan datos para validar sus experimentos. Y aunque se disponga de información procedente de sensores o redes de sensores, éstas pueden no ser accesibles por motivos de legalidad como ley de protección de datos al estar asociadas a usuarios, ser propiedad de empresas privadas, etc.

Además, obtener datos de una red de sensores lleva asociado un doble coste, debido a que desplegar una red de sensores puede ser costosa en términos económicos, y en términos temporales, dado que los sensores necesitan realizar un muestreo que en determinados casos conllevan una prolongación temporal de semanas o incluso meses, para poder disponer de datos y llevar a cabo una experimentación.

Como solución a los problemas comentados anteriormente, se dispone de diversos repositorios de datos tanto públicos como privados. Sin embargo, no siempre se adaptan a las necesidades del problema que se pretende tratar.

La solución a todas estas limitaciones es la generación de datos o señales sintéticas que emulen el comportamiento de la realidad del problema que se pretende abordar. Este tipo de señales tiene múltiples ventajas que ayudan a solventar las limitaciones citadas anteriormente [18]. Por un lado destaca la robustez, debido a que los sensores del mundo real que componen las redes de sensores, pueden proporcionar datos erróneos en determinados casos. Los datos obtenidos por un generador de datos sintéticos carecen de este problema.

Otra característica destacable es la seguridad, puesto que los datos sintéticos pueden generarse con un nivel de detalle y realismo tal, que no es necesario asumir ningún tipo de riesgo, frente a lo que puede suceder en algunas disciplinas de la ciencia como en medicina.

Los datos sintéticos como herramienta para poner a prueba métodos y modelos desarrollados se utilizan en diversos campos científicos, tales como reconocimiento y generación de patrones [21], minería de datos [30], en aprendizaje automático [11], etc.

Este trabajo presenta un método para la generación de señales sintéticas basado en funciones de distribuciones estadísticas. Este trabajo se enmarca dentro del Proyecto ALCOR [12] para proporcionar conjuntos de datos con suficiente amplitud para poder realizar experimentaciones con un volumen elevado de datos. Al generar dichas señales de datos en base a distribuciones estadísticas se pueden conseguir datos que tengan un comportamiento similar al resultado que pueden proporcionar diversos sensores que existen en la actualidad tales como sensores con el fin de comprobar métodos y modelos como los trabajos realizados por [10, 16].

Una generación de señales de carácter pseudo-realistas, no puede consistir en obtener muestras aleatorias de distribución de probabilidad uniforme, pues el resultado sería una sucesión de valores carentes de toda coherencia. Debido a esto, se necesitan diferentes mecanismos para modificar la señal.

Junto con el modelo de generación de datos sintéticos con carácter pseudo-realistas que se propone en el presente trabajo, se proporciona una herramienta web disponible en [25] que implementa el modelo propuesto. Los datos generados pueden ser almacenados en el formato más usual de las bases de datos para experimentación (comma-separated values “CSV”).

El presente documento se organiza de la siguiente forma: En la sección. 6.3 se muestran las propuestas de otros autores en la generación de señales sintéticas. La sección. 6.4 describe las restricciones de las señales que generará el modelo y el método de generación de señales propuesto. Los resultados y su análisis se muestran en la sección. 6.5. Finalmente, en la sección. 6.6 se muestran las conclusiones obtenidas de los experimentos realizados y en la sección. 6.7 se muestran las posibles mejoras aplicables al método desarrollado.

6.3 Trabajos relacionados

La generación de señales sintéticas ha sido utilizadas en muchos campos de la ciencia. Por ejemplo en [24], los autores proponen un modelo llamado *WGENK* para la generación de datos sintéticos orientados a agricultura. *WGENK* es una variación del modelo *WGEN* [9]. En el trabajo de Kuchar et al. se generan datos tales como la radiación solar diaria, temperaturas mínimas y máximas precipitaciones, etc. Los autores logran alcanzar un modelo de generación que se aproxima a la realidad y realizan un contraste con datos reales para validar sus resultados.

En [5] los autores realizan una modificación de la herramienta *open-source Generator* [6] y hacen uso de una base de datos que contiene el censo poblacional de Irlanda. Los autores logran demostrar en sus experimentos que haciendo uso de herramientas de generación de datos sintéticos, y con las restricciones adecuadas se pueden conseguir datos que contengan las mismas métricas estadísticas que los datos del mundo real.

Observando las propuestas de los diferentes autores, se observa que existe cierta tendencia en la literatura científica de usar lenguajes específicos para etiquetar los datos [18]. En [19] los autores proponen un método para la generación de grandes conjuntos de datos de forma paralela. Utilizan el lenguaje *SDDL* (*Synthetic Data Description Language*) dado que los datos tienen que ser generados con diferentes restricciones. Este lenguaje está basado en *XML* (*Extensible Markup Language*) y es utilizado por muchos generadores de datos sintéticos cuando se necesita etiquetar datos y agregar restricciones a la generación.

Otro ejemplo del uso de lenguajes de etiquetado, lo encontramos en [4], donde los autores desarrollan un *framework* que hace uso de estructuras basadas en *XML* para generar grandes volúmenes de datos. El modelo de generación de los datos sintéticos propuestos por los autores se compone de dos fases. La primera es la generación de los ficheros *XML* junto con la extracción de características de los datos. Y la segunda fase es la generación de los datos basándose en diferentes distribuciones estadísticas tales como distribuciones de Poisson, normales y geométricas. De los resultados experimentales los auto-

res concluyen que los datos generados tienen un comportamiento similar a los datos reales bajo un coeficiente de confianza del 95 %.

En [22] los autores realizan la generación sintética de datos haciendo uso del estándar abierto *PMML (Predictive Model Markup Language)* como puente entre la base de datos original y el fichero *SDDL* generado. Una vez han conseguido el fichero *SDDL* utilizan un método *PSDG (Parallel synthetic data generation)* para obtener el nuevo conjunto de datos. En la experimentación llevada a cabo demuestran que utilizando la base de datos *Iris* [13], los datos sintéticos comparten las mismas características que los datos originales y que no existen diferencias significativas entre ellos.

Donde ha tenido una gran acogida la generación de datos sintéticos ha sido en minería de datos, en reconocimiento de patrones y en aprendizaje automático. Por ejemplo en [14] los autores utilizan la generación de datos sintéticos para validar métodos de aprendizaje automático y de minería de datos. En el trabajo de los autores se emplea un método denominado *WGKS (White Gaussians on k-simplex)*, que genera datos mediante distribuciones gaussianas. Al estar enfocado a generación de datos para aprendizaje automático se controlan factores como el número de clases y el error bayesiano.

En [30] los autores generan nuevos conjuntos de datos sintéticos por medio de árboles de decisión mediante una modificación del algoritmo *ID₃ (Iterative Dichotomiser 3)*. Mediante el uso de los árboles de decisión los autores consiguen crear interdependencia entre los datos de los conjuntos de datos generados con la intención de obtener conjuntos de datos genéricos con los que testear cualquier aplicación de aprendizaje automático.

En métodos de minería de datos tales como agrupamiento y detección de *outliers* no existen en ocasiones conjuntos de datos que sean útiles para probar la eficacia de dichos métodos [29]. Este es el hecho que motiva a los autores a crear un método que genera datos de forma sintética en base a diferentes distribuciones estadísticas, con un determinado número de *clusters*, un nivel de dificultad concreto y la capacidad de incorporar un determinado ruido en la generación, para simular aquellos patrones que serán *outliers*.

En el estudio de series espacio-temporales encontramos que en [34], los autores proponen el uso del algoritmo *GSTD (Generate Spatio Temporal Data)*. Este método ha sido desarrollado por los autores para la generación sintética de datos con carácter espacio-temporal en dos dimensiones. Dicho algoritmo es capaz de modificar los parámetros asociados a un objeto y modificar su posición y tamaño a lo largo de un determinado intervalo de tiempo. Los atributos asociados al objeto como el intervalo, pueden generarse mediante una función de probabilidad estadística normal o sesgada.

Con un objetivo similar al propuesto en el trabajo anterior, [17] proponen una generación de conjuntos de datos de series espacio-temporales centrada en la generación de datos sintéticos de topologías de redes de sensores irregulares. Mediante los experimentos realizados y los casos de estudio en los que se ha utilizando el sistema *DIMENSIONS* [15], los autores demuestran que los datos sintéticos poseen características similares a los datos reales.

Tras analizar los trabajos propuestos por otros autores, se detecta que en general no existe un sistema de generación de señales sintéticas de propósito general, sino que se centran en proporcionar soluciones para problemas específicos. Aunque en el presente trabajo se compartan aspectos en común con

los trabajos analizados, como el uso de distribuciones estadísticas, la principal ventaja del método de generación propuesto es que consigue realizar la generación mediante el uso del gradiente de la señal y la combinación de diferentes niveles modificadores permitiendo controlar el comportamiento de la señal.

6.4 Método

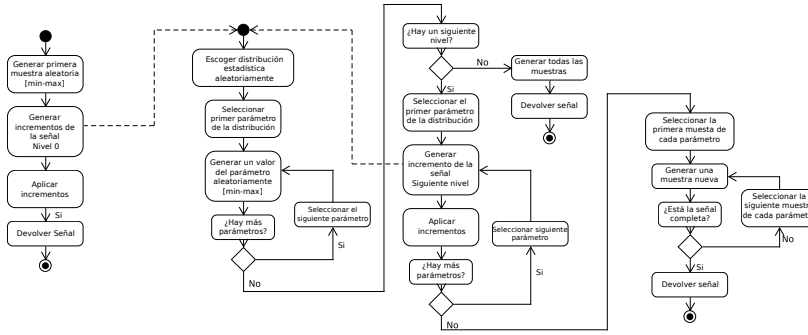


Figura 6.1
Diagrama de flujo del método propuesto

El método propuesto utiliza funciones de generación de números aleatorios para construir una señal con un número determinado de muestras a partir de un rango acotado. Es necesario que el método de generación de señales sintéticas sea capaz de generar muestras digitales que sean verosímiles desde un punto de vista cualitativo. Además, las señales resultantes deben cumplir el teorema de [27] y [32].

Como parámetros de entrada al modelo se consideran el número de muestras N , y el rango de valores posibles, representados por los valores frontera mínimo y máximo: s_m, s_M .

De acuerdo con (6.1) y (6.2), una señal de N muestras puede expresarse mediante la primera muestra y la señal correspondiente a los incrementos muestra a muestra de la señal original.

$$\forall s \in \mathbb{R}^n \exists s' \in \mathbb{R}^{n-1} / s'[i] = s[i+1] - s[i] \quad (6.1)$$

$$s[i] = s[0] + \sum_{j=0}^{i-1} s'[j] \quad (6.2)$$

Se introduce también un conjunto de distribuciones de probabilidad, cada una de ellas definida mediante un conjunto de parámetros reales cuya instanciación da lugar a una distribución de probabilidad concreta, Gamma y Gaussiana (Normal).

La distribución de probabilidad gamma consta de dos parámetros siempre positivos, α y β . El primer parámetro, es el que representa la máxima intensidad de probabilidad y por tanto la forma de la distribución. Y el valor de β representa el alcance de la asimetría positiva hacia la derecha. Esta distribución dada sus características, permite moldear en determinados casos [26]

otros tipos de distribuciones.

La distribución de probabilidad normal cuenta con los parámetros media (μ) y desviación típica (σ), y cada par de estos parámetros da lugar a una distribución de probabilidad diferente.

De manera general, la generación de N muestras acotadas entre s_m y s_M es un proceso que consta de las siguientes fases:

1. Se genera el valor inicial de la señal de manera aleatoria siguiendo una distribución de probabilidad uniforme.
2. Del conjunto de distribuciones de probabilidad, se escoge una aleatoriamente.
3. Cada parámetro que caracteriza la distribución de probabilidad resultante se genera aleatoriamente dentro de unos márgenes preconfigurados y, de nuevo, mediante una distribución de probabilidad uniforme.
4. Utilizando la distribución de probabilidad ya generada, se extraen $N-1$ muestras aleatorias, y se construye la señal con (6.2).

Con este método se obtiene una señal con coherencia (ya que sus incrementos siguen una distribución de probabilidad concreta) pero monótona. La naturaleza presenta variaciones que difícilmente se van a modelar satisfactoriamente utilizando una distribución de probabilidad constante. Para obtener señales cualitativamente más reales, la propuesta que se presenta es utilizar el procedimiento expuesto anteriormente para generar no solo variaciones en la señal, sino variaciones de los parámetros de la distribución de probabilidad que genera esta señal.

Este concepto da lugar a un procedimiento recursivo que se puede visualizar como un sistema de generación de señales por niveles, en el que el nivel 0 corresponde al nivel de la señal (el fin último del proceso), el nivel 1 corresponde a las distribuciones de probabilidad que generarán los incrementos de la señal, el nivel 2 corresponde a las distribuciones de probabilidad que generarán los incrementos que harán cambiar los parámetros de la distribución de probabilidad del nivel anterior, etc.

De manera genérica, el nivel de generación x es invocado para generar una señal aleatoria en el nivel anterior, para lo cual genera aleatoriamente la configuración inicial de una distribución de probabilidad escogida al azar y solicita al nivel de generación $x + 1$ (si lo hubiere) que genere los cambios dinámicos de sus propios parámetros.

Un aspecto importante a tener en cuenta es la configuración necesaria a la hora de generar la señal. Como se ha mencionado anteriormente, cada tipo de distribución de probabilidad necesita unos parámetros de configuración que deben ser acotados uno a uno y cada nivel. Esta configuración permite controlar la aleatoriedad del comportamiento de la señal sin que se los sucesivos incrementos generados se descontrolen.

Por ilustrar este aspecto, considérese que se desea generar una señal cuyas muestras estén acotadas entre s_m y s_M ; lo lógico es diseñar el primer nivel con unos parámetros de configuración para generar cambios que estén proporcionados con el margen dinámico $s_M - s_m$. Este procedimiento debe extrapolarse a todos los niveles, para que cada nivel aporte variaciones en una escala controlada al nivel anterior. Para una comprensión más detallada sobre el funcionamiento del método, la Figura 6.1 muestra el diagrama de flujo del mismo.

Además, en el Apéndice A se proporciona un ejemplo simplificado del código del método propuesto. Adicionalmente en el Apéndice B en las Figura 6.8 y Figura 6.9 se encuentran un diagrama de las clases de la implementación del método desarrollado y un diagrama de instancias que muestran un ejemplo de como son distribuidos los parámetros por las clases y las funciones utilizadas en el caso de dos niveles de generación.

6.5 Resultados experimentales

6.5.1 Herramienta de generación de señales

Con el fin de comprobar la utilidad del método, se ha realizado la implementación del mismo en una herramienta web llamada “SysGpr” la cual permite configurar y generar señales, disponible en [25].

En la Figura 6.2 se puede observar la pantalla principal de la aplicación. En la parte izquierda, la cabecera “Configuration” permite realizar la configuración básica de la señal modificando parámetros como el rango de valores y el número de muestras.

Las cabeceras “Level 0” y “Level 1” permiten configurar los diferentes niveles (la barra de herramientas “Levels” permite añadir o eliminar niveles) de distribuciones estadísticas y habilitar o deshabilitar las distribuciones. En caso de tener seleccionadas dos distribuciones, para cada señal nueva que se genere se escoge una de las dos de forma aleatoria.

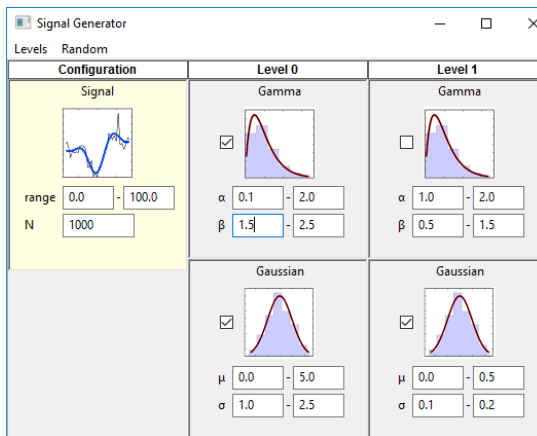


Figura 6.2
Interfaz para la generación de señales

Una vez establecidos los parámetros deseados para generar una señal, se puede tal y como se muestra en la Figura 6.3, generar tantas señales como se desee en una sola ejecución (esta característica se encuentra dentro de “Random” en la barra de herramientas). Esta característica agiliza enormemente la tarea en caso de necesitar un gran número de señales para trabajar con ellas posteriormente.

6.5.2 Muestras generadas

Para mostrar el funcionamiento del modelo, se han llevado a cabo cuatro generaciones de conjuntos de señales para que se puedan apreciar visualmente

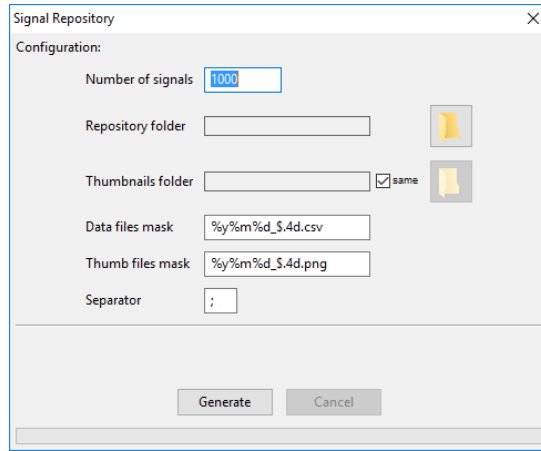


Figura 6.3
Generador de repositorios de señales

los resultados.

En el primer conjunto de señales sintéticas (T') se generan datos similares a los obtenidos por un sensor de temperatura *DHT22* (T). Los resultados de la generación de dichos datos se pueden observar en la Figura 6.4. La primera señal de esta figura se corresponde con los datos obtenidos desde el sensor de temperatura durante 200 min en un recinto climatizado y con fluctuaciones de aire que modificaban la temperatura. Los parámetros de generación para obtener datos similares a los obtenidos por dicho sensor, corresponden a un nivel de distribuciones gaussianas con valores comprendidos entre 20°C y 22°C con un total de 200 muestras y $\mu = [0 - 0.1]$ y $\sigma = [0.01 - 0.1]$. Dichos valores han sido escogidos analizando la naturaleza de las señales de temperatura debido a que los cambios en temperatura en las condiciones del experimento son moderados.

El segundo conjunto (V') de experimentos se corresponde con la Figura 6.5. En dicha figura se representa en la gráfica de la izquierda la señal de velocidad obtenida por un anemómetro *NRG40* (V). En este caso las señales sintéticas se han generado con una función de distribución de nivel 0 gamma con un rango de valores entre 0 km/h y 1.5 km/h y un total de 200 muestras. Dado que la velocidad del viento puede cambiar drásticamente y tener un comportamiento muy pronunciado en algunos instantes la señal se ha generado con unos valores de $\alpha = [0.1 - 0.5]$ y $\beta = [0.1 - 0.5]$.

Para demostrar el uso de diferentes niveles de modificación en la Figura 6.6 se muestran el conjunto (P') generado con un nivel modificador de distribución gamma y un nivel inicial gaussiano. Al igual que sucede con los casos anteriores, en la parte izquierda de la figura se encuentra la señal original, procedente de un sensor de presión atmosférica *BMP180* (P). El resto de figuras corresponden con algunas señales extraídas del conjunto de señales generadas con parámetros $\mu_{L_0} = [0.01 - 0.05]$, $\sigma_{L_0} = [0.01 - 0.03]$ y $\alpha_{L_1} = [0.01 - 0.03]$, $\beta_{L_1} = [0.01 - 0.05]$. La señal sintética tiene un rango de valores entre 1015 hPA y 1016 hPA y un total de 200 muestras. Estos parámetros han sido seleccionados debido a que la presión atmosférica raramente suele cambiar dependiendo del periodo de muestreo muy bruscamente. El segundo nivel añade un control de grado fino para ajustar el comportamiento de la señal.

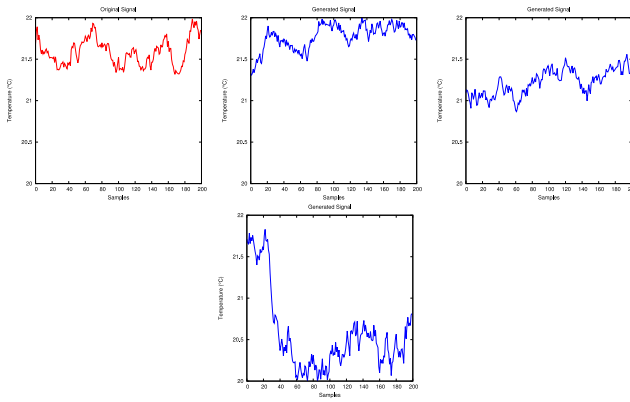


Figura 6.4
Señal original y señales generadas aleatoriamente utilizando solo el nivel 0 con distribución gaussiana

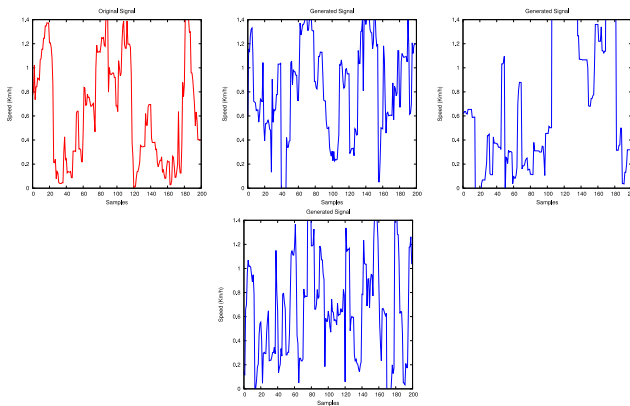


Figura 6.5
Señal original y señales generadas aleatoriamente utilizando solo el nivel 0 con distribución gamma

La Figura 6.7 representa el último conjunto de señales sintéticas (I'), dedicado en esta ocasión a la generación de datos de irradiación solar. La gráfica de la izquierda corresponde con los datos reales obtenidos de un pirómetro *SP-215* (I) durante el mediodía en el mes de febrero con presencia de nubes en la azotea de uno de los edificios de la Universidad de Córdoba, España. Para modelar señales con un carácter realista que representen datos similares a los que proporciona el sensor, se ha propuesto utilizar dos niveles modificadores de la señal. Ambos niveles corresponden con señales gaussianas y tienen como valores de sus parámetros característicos: $\mu_{L_0} = [0 - 5]$, $\sigma_{L_0} = [1 - 2.5]$ y $\mu_{L_1} = [0 - 2]$, $\sigma_{L_1} = [0.5 - 1.5]$. El rango de valores de la señal está comprendido entre 200 w/m^2 y 300 w/m^2 con un total de 200 muestras.

6.5.3 Validación

La validación de los resultados del método propuesto se ha llevado a cabo mediante tres métricas diferentes.

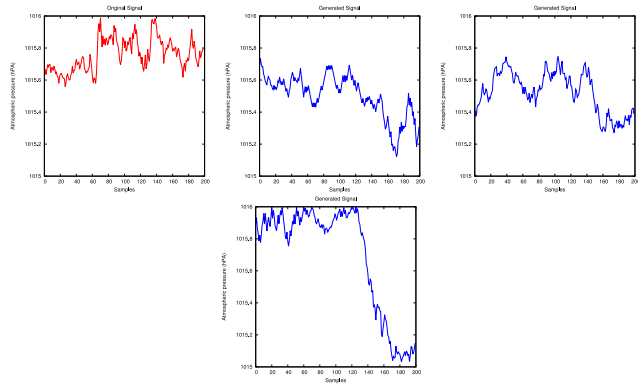


Figura 6.6
Señal original y señales generadas con distribución gaussiana de nivel 0 y nivel 1 de tipo gamma

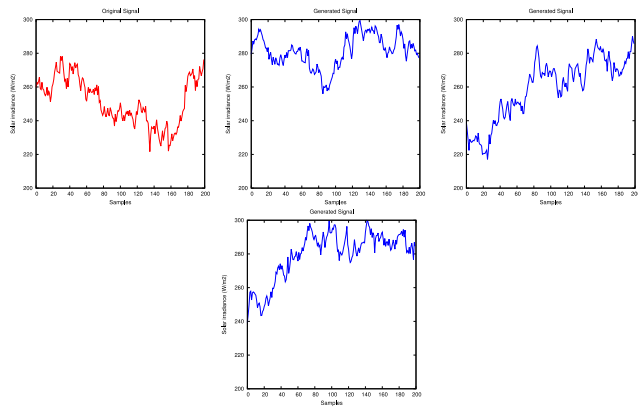


Figura 6.7
Señal original y señales generadas con distribuciones gaussianas en nivel 0 y en nivel 1

Validación mediante algoritmo de aprendizaje C4.5

La primera validación se ha llevado a cabo utilizando el algoritmo de aprendizaje automático de árboles de decisión C4.5 [31]. Se han realizado cuatro pruebas diferentes que corresponden a los diferentes conjuntos de datos generados en el apartado anterior (T', V', P', I'). Dado que se emplea un algoritmo de aprendizaje automático, es necesario entrenar el modelo para posteriormente llevar a cabo una validación o etapa de test. Los datos de entrenamiento para cada modelo incorporan 200 señales reales procedentes de cada sensor (T, V, P o I) y 200 señales procedentes de los otros sensores para que el modelo aprenda que tipo de señales no corresponden a los datos sensor que se pretende clasificar.

Para el conjunto de test se han utilizado 30 señales catalogadas como T, V, P e I , pero que realmente se tratan de señales pseudo-naturales de los conjuntos T', V', P' e I' . Por otro lado se han escogido otras 30 señales catalogadas como $\bar{T}, \bar{V}, \bar{P}, \bar{I}$ que también son señales pseudo-naturales que emulan el resto de sensores.

El motivo de utilizar 400 ($200 + 200$) y 60 ($30 + 30$) señales en las fases de entrenamiento y test, es que tal y como estipula el teorema central del límite, estas cifras se consideran suficientes y representativas de las poblaciones de datos de las que proceden.

De este modo, el primer modelo de clasificación dispondrá para su entrenamiento de un total de 400 señales. La mitad de ellas señales procedentes del sensor de temperatura (T) y el resto que forman el conjunto de entrenamiento etiquetadas como \bar{T} son señales procedentes de V, P o I , que corresponden con los sensores restantes. En la fase de test, el conjunto estará formado por 30 señales de temperatura generadas con el método propuesto T' pero etiquetadas como T y 30 señales sintéticas de tipo V', P' o I' catalogadas como clase \bar{T} .

Hay que destacar que todas las señales, que se han utilizado para la fase de aprendizaje y Test, han sido normalizadas a valores en el rango $[0 - 1]$ para eliminar el efecto de sesgo aditivo o cambios de escala, dado que el objetivo es comprobar que las señales tienen un comportamiento similar.

La siguiente lista detalla el tipo de señales que componen los conjuntos de entrenamiento y test para los diferentes modelos y casos estudiados:

- Modelo para señales de temperatura:
 - Datos para etapa de aprendizaje:
 - * T : 200 señales obtenidas del sensor de temperatura *DHT22*.
 - * \bar{T} : 200 señales procedentes de V, P e I .
 - Datos para etapa de Test:
 - * T : 30 señales pseudo-naturales del conjunto T' .
 - * \bar{T} : 30 señales pseudo-naturales procedentes de V', P' e I' .
- Modelo para señales de velocidad:
 - Datos para etapa de aprendizaje:
 - * V : 200 señales obtenidas del anemómetro *NRG40*.
 - * \bar{V} : 200 señales procedentes de T, P e I .
 - Datos para etapa de Test:
 - * V : 30 señales sintéticas del conjunto V' .
 - * \bar{V} : 30 señales pseudo-naturales procedentes de T', P' e I' .
- Modelo para señales de presión:

- Datos para etapa de aprendizaje:
 - * P: 200 señales obtenidas del sensor de presión atmosférica *BMP180*.
 - * \bar{P} : 200 señales procedentes de T, V e I.
- Datos para etapa de Test:
 - * P: 30 señales pseudo-naturales del conjunto P' .
 - * \bar{P} : 30 señales pseudo-naturales procedentes de T' , V' e I' .
- Modelo para señales de irradiancia solar:
 - Datos para etapa de aprendizaje:
 - * I: 200 señales obtenidas del pirómetro *SP-215*.
 - * \bar{I} : 200 señales procedentes de T, V y P.
 - Datos para etapa de Test:
 - * I: 30 señales pseudo-naturales del conjunto I' .
 - * \bar{I} : 30 señales pseudo-naturales procedentes de T' , V' y P' .

En la Tabla 6.1 se pueden observar los resultados del algoritmo de clasificación C4.5. En dicha tabla se muestra resumida toda la información de los datos obtenidos en la etapa de test del clasificador. En la parte de la izquierda se muestra la tabla de contingencia de cada uno de los modelos estudiados. Además, a la izquierda de cada tabla de contingencia, se proporcionan métricas como son el ratio de verdaderos positivos (TP) y Falsos Positivos (FP) y el área bajo la curva ROC.

Tabla 6.1
Resultado algoritmo clasificación C4.5 para las diferentes señales

	T	\bar{T}	TP	FP	ROC
T	22	8	0,733	0,167	0,783
\bar{T}	5	25	0,833	0,267	0,783
	V	\bar{V}			
V	22	8	0,733	0,367	0,693
\bar{V}	11	19	0,633	0,267	0,693
	P	\bar{P}			
P	25	5	0,833	0,233	0,789
\bar{P}	7	23	0,767	0,167	0,789
	I	\bar{I}			
I	23	7	0,767	0,233	0,802
\bar{I}	7	23	0,767	0,233	0,802

Validación mediante autocorrelaciones y correlaciones cruzadas

En segundo lugar se han realizado pruebas de autocorrelaciones cruzadas y correlaciones cruzadas entre las diferentes señales. Los resultados de las autocorrelaciones y correlaciones medias, pueden observarse en la Tabla 6.2.

Las autocorrelaciones cruzadas se han llevado a cabo tomando únicamente una muestra de 30 señales de un tipo en concreto de señales reales de temperatura (T), velocidad (V), presión (P) o Irradiancia (I), y se han calculado

las correlaciones señal a señal con ellas mismas para comprobar el grado de similitud entre las señales originales.

Para las correlaciones cruzadas se han tomado un total de 30 muestras de señales aleatorias de cada conjunto T' , V' , P' o I' , y se han comparado una a una con las señales escogidas para las autocorrelaciones de la misma naturaleza. Por ejemplo en el caso de las señales de temperatura se ha obtenido la correlación media entre T y T' para observar la similitud de las señales generadas con las señales reales.

En ambos casos se han escogido muestras aleatorias simples cumpliendo el teorema central del límite obteniendo así un número de muestras suficiente para que sean representativas de las poblaciones de los diferentes conjuntos. Además al igual que ocurría en la validación mediante el algoritmo de clasificación, las señales se han normalizado al rango $[0 - 1]$, para mitigar el efecto cambios de escala y poder comparar las señales en forma.

Puede observarse de los resultados obtenidos en la Tabla 6.2, que las señales pseudo-naturales guardan en media una similitud en torno al 80% con las señales obtenidas de sensores reales, y que el 20% se debe a la propia variación entre las señales originales. Debido a estas variaciones, en el caso de las señales pseudo-naturales de temperaturas se obtiene una correlación mayor que las propias autocorrelaciones cruzadas de las señales reales, lo que significa que hay más cantidad de señales generadas con una forma similar, existiendo una mayor homogeneidad que en las señales obtenidas de los sensores reales.

Tabla 6.2
Resultado de análisis de autocorrelaciones cruzadas y correlaciones cruzadas.

Tipo de señales	Autocorrelaciones	Correlaciones
Temperatura	$T/T' : 0.84056$	$T/T' : 0.84565$
Velocidad	$V/V' : 0.81264$	$V/V' : 0.80436$
Presión	$P/P' : 0.83740$	$P/P' : 0.83284$
Irradiancia	$I/I' : 0.86898$	$I/I' : 0.85313$

Validación mediante MOS (Mean Opinion Score)

Adicionalmente, se ha utilizado *Mean Opinion Score (MOS)* [20], para realizar una evaluación de calidad subjetiva de las señales pseudo-realistas con el fin de contrastar los resultados proporcionados por el generador de señales con señales reales proporcionadas por sensores.

Para llevar a cabo la validación mediante *MOS*, se han realizado una serie de encuestas a diferentes usuarios. Las encuestas pretenden evaluar de manera empírica la calidad de las señales generadas, éstas fueron realizadas por seis investigadores los cuales tienen conocimientos relacionados con el tratamiento de señales (el número de expertos escogidos, cumple los requisitos mínimos exigidos por *MOS* [33] para obtener resultados concluyentes). En dichas encuestas, a los expertos se les pedía que clasificasen las diferentes señales que se les mostraban para comprobar si los científicos eran capaces de diferenciar las señales reales y las generadas por el método propuesto. A cada investigador se le mostraron dos bloques de noventa preguntas con cuatro imágenes de señales. Cada bloque correspondía con un tipo de señal generada: señales

pseudo-naturales de un nivel modificador y señales pseudo-naturales de dos niveles modificadores. Dentro de las cuatro señales se encontraba una señal original y tres señales pseudo-naturales en orden aleatorio (dentro de cada uno de los grupos anteriormente descritos). A cada señal mostrada, cada experto debía etiquetarla como “real” o “sintética”, aunque podría dejarla sin responder en caso de no ser capaz de decidirse. Un ejemplo de los diferentes grupos de señales que analizaron los expertos, se pueden observar en las Figuras 6.4, 6.5, 6.6 y 6.7. Hay que notar, que en el presente documento las señales aparecen etiquetadas como *Original signal* (señales rojas) y *Generated signal* (señales azules), para que el lector pueda observar las diferencias entre señales generadas y originales. Sin embargo, los expertos no disponían de dicha información, ni de colores que las diferenciasen, para evitar dar información de la naturaleza de los datos que analizaban. Los únicos datos que los expertos conocían, eran la señal con la información de los ejes X e Y, que mostraba el tipo de señal, temperatura, velocidad del viento, presión atmosférica y radiación solar.

En la Tabla 6.3 se pueden observar los resultados obtenidos mediante MOS. Estos resultados han sido obtenidos comparando las respuestas de los diferentes expertos al clasificar los diferentes conjuntos de datos. Se han comprobado las etiquetas determinadas por los expertos, con las etiquetas reales que deben de tener cada señal pseudo-natural. El resultado fue que los expertos identificaron erróneamente o no pudieron distinguir en un 71.25 % de los casos si una señal procedía de una captura de datos real o de las proporcionadas por el generador de datos pseudo-naturales. Se observó que el 30 % de las señales pseudo-naturales de un único nivel modificador fueron clasificadas correctamente por los expertos. Asimismo, en el caso de las señales pseudo-naturales con dos niveles de modificadores, el 72.5 % señales pasaban desapercibidas a ojos de los expertos. En la Tabla 6.4 se pueden observar los resultados acumulados de las respuestas de la evaluación MOS. En dicha tabla, se pueden ver, por filas, el tipo de señales presentadas a los expertos, mientras que por columnas se puede ver la clasificación de cada señal por parte de los expertos. Se consideran aciertos aquellas señales reales identificadas como reales y aquellas señales sintéticas identificadas como sintéticas. Todas las demás combinaciones (incluidas todas aquellas señales que los expertos no sabían si eran reales o sintéticas y por tanto no contestaron) se consideran fallos.

Tabla 6.3

Tabla de resultados extraídos de MOS. Un nivel modificador (L1), Dos niveles modificadores (L2).

Tipo de señales	Fallos y No Sabe		Aciertos	
	Señales	%	Señales	%
L1	1512	70 %	648	30 %
L2	1566	72.5 %	594	27.5 %
Media (L1,L2)	71.25 %		28.75 %	

Tabla 6.4

Tablas de contingencia de MOS. Un nivel modificador (L1), Dos niveles modificadores (L2). En filas, tipo de señales presentadas a los expertos. En columnas, número de señales según la respuesta de los expertos para cada tipo.

Tipo de señales	Reales	Sintéticas	No sabe
L1			
Reales	48	66	426
Sintéticas	282	708	738
L2			
Reales	66	114	360
Sintéticas	306	528	786

6.6 Conclusiones

En el presente trabajo se ha propuesto un método para generar señales sintéticas que tengan cierta similitud a las que podrían obtenerse desde un sensor. Dicho método tiene la capacidad de generar datos en los que el comportamiento de la señal es capaz de seguir un comportamiento basado en distribuciones gaussianas y gamma, proporcionando un carácter realista.

El método propuesto aporta un enfoque distinto en el campo de generación de señales mediante la aleatorización y determinación del comportamiento del gradiente de la señal que se pretende generar. Esta característica permite añadir niveles para modificar la señal de forma que al añadir sucesivos niveles de modificación se puede controlar el comportamiento de la señal de una forma concreta para ajustarla a las características deseadas.

Además se ha desarrollado una interfaz visual que hace uso del método propuesto y proporciona a los científicos una manera simple de poder generar grandes repositorios de datos pseudo-naturales para que puedan testear sus propios métodos en los diferentes campos científicos, sin que tengan que esperar para obtener los datos desde sensores reales o realizar etapas previas de tratamiento de datos para eliminar datos erróneos o incompletos.

Respecto a los resultados obtenidos se han realizado validaciones mediante tres enfoques distintos a diferentes conjunto de datos procedentes de sensores y a señales pseudo-naturales generadas con el método propuesto.

Para la primera validación se ha empleado el algoritmo de aprendizaje C4.5. Dicho algoritmo ha utilizado señales reales en la fase de entrenamiento, pero en la fase de test las señales han sido sustituidas por señales pseudo-naturales para comprobar la similitud de los datos generados. De los resultados se puede concluir que las señales sintéticas pseudo-naturales han sido clasificadas en media como señales reales en más de un 75 % (tomando los resultados de patrones bien clasificados) de los casos y que únicamente en aproximadamente un 25 % de los casos son detectadas como señales sintéticas. En cuando a las métricas para evaluar la calidad de los clasificadores se ha escogido el área bajo la curva ROC que en media es superior a 76 %.

Para la segunda validación se han calculado las correlaciones cruzadas para los cuatro conjuntos de datos detallados en la sección 6.5.2. Por un lado se han obtenido las autocorrelaciones cruzadas de los datos procedentes de los

sensores reales obteniendo un 20 % de diferencia en media, que se debe principalmente a la variabilidad de las propias señales. Al calcular las correlaciones cruzadas entre los datos reales y los generados por el método propuesto, se observa que la relación en media entre ambas poblaciones es mayor al 80 %.

Por último se ha realizado una validación mediante *MOS* para determinar con ayuda de expertos en el campo de tratamiento de señales la eficacia del método propuesto. Dichos expertos han etiquetado los diferentes conjuntos de señales sin información previa que delatasen a las señales pseudo-naturales, como reales o sintéticas. De los resultados medios de dichas pruebas se ha obtenido que los expertos sólo son capaces de diferenciar las señales pseudo-naturales de las originales en un 28.75 % de los casos.

6.7 Trabajo futuro

Una de las principales mejoras consiste en añadir una variedad de distribuciones estadísticas nuevas a la aplicación y la posibilidad de crear señales con componentes periódicos, con lo que se ampliaría el abanico de posibilidades de la aplicación.

Por otro lado, la inclusión de métodos metaheurísticos y de aprendizaje automático permitiría obtener los parámetros estadísticos característicos de una señal o conjunto de señales dados, para generar nuevos conjuntos de datos sintéticos de forma automática, preservando la misma naturaleza de los datos de entrada [3].

6.8 Apéndice A. Código simplificado del método

6.8.1 Función *generateSignal()*

```

Generator::generateSignal()
Begin

    signal = []
    increments = first_level.generateIncrements(samples-1)

    previous_sample = random_number(max = maximum, min = minimum)
    signal.append(previous_sample)

    for inc in increments:
        sample = previous_sample + random_choice(-1,1)*inc
        if sample > maximum: sample -= 2*inc
        elif sample < minimum: sample += 2*inc
        signal.append(sample)
        previous_sample = sample
    return signal

End

```

6.8.2 Función *generateIncrements()*

```

Generator_Level::generateIncrements(N)
Begin

```

```
st_distribution = random_choice(GetDistributions())

for parameter in st_distribution.GetParameters():
    parameter.SetRandomValue()

signal = []
if next_level == Null:
    while len(signal) < N:
        signal.append(st_distribution.generateSample())

    else:
        increments = []
        parameters = st_distribution.GetParameters()

        for i in len(parameters)
            increments.append(next_level.generateIncrements(N-1))
            j = 0
            signal.append(st_distribution.GenerateSample())

        while len(signal) < N:
            for i in len(parameters):
                parameter = parameters[i]
                inc = increments[i][j]
                previous_value = parameter.value
                parameter.value = previous_value + random_choice(-1,1)*
                    inc
                if parameter.value > parameter.maximum: parameter.value -=
                    2*inc
                elif parameter.value < parameter.minimum: parameter.value
                    += 2*inc

            signal.append(st_distribution.GenerateSample())
            j++
        return signal
```

End

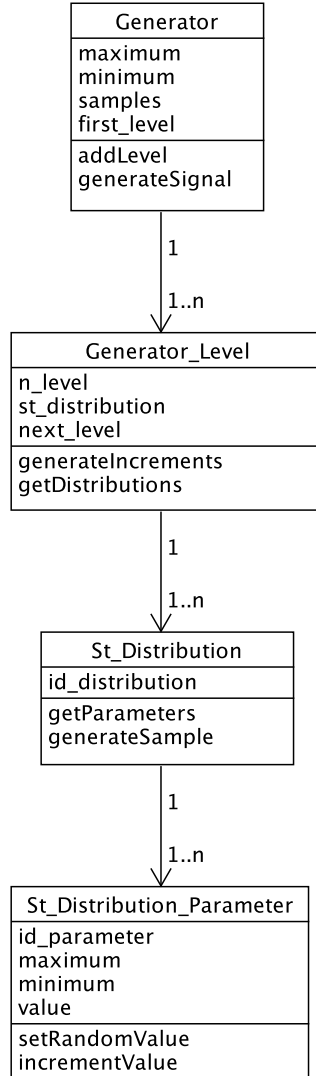


Figura 6.8
Diagrama de clases.

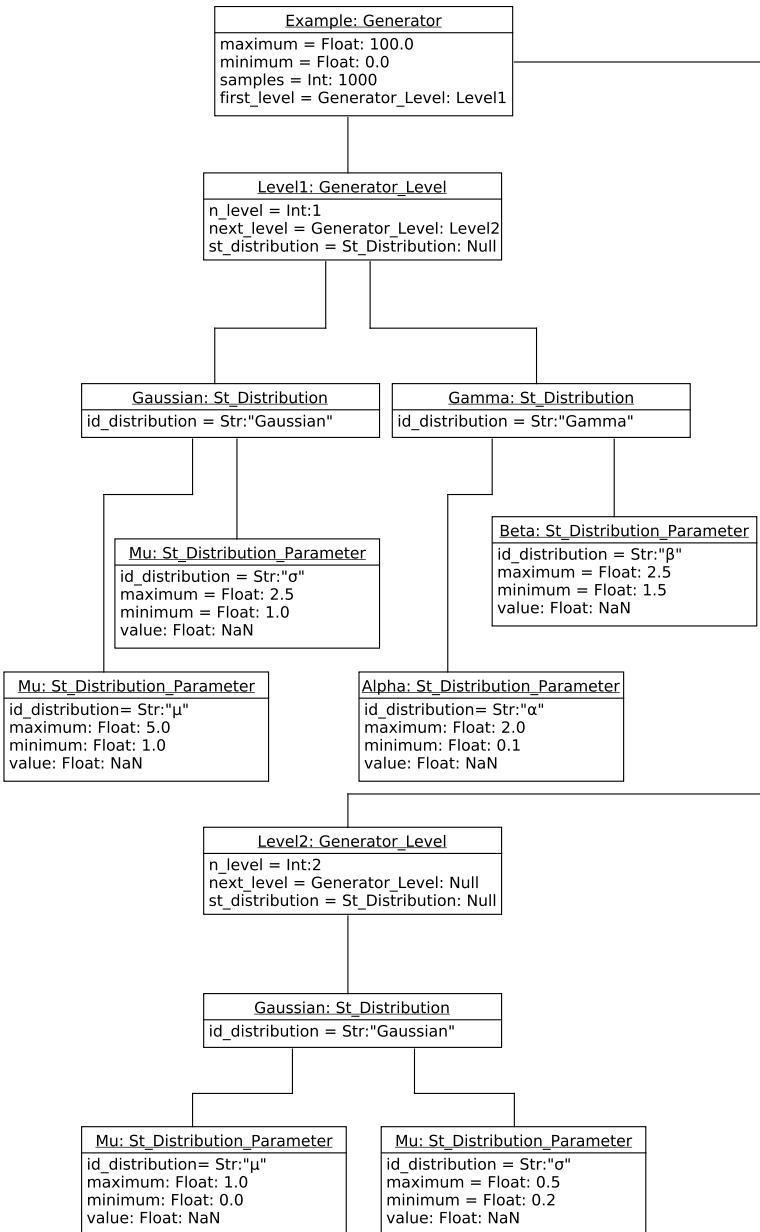


Figura 6.9
Diagrama de instancias para dos niveles de generación.

REFERENCIAS

- [1] R. Minerva A. Biru and D. Rotondi: *Towards a definition of the internet of things (iot)*. Technical report, IEEE Tech. Rep., 2015.
- [2] Naufal Alee, Phaklen Ehkan, L. M. Kamarudin, and A. Harun: *Size efficiency for sensor node with embedded processing unit*. In *2016 3rd International Conference on Electronic Design (ICED)*. IEEE, aug 2016.
- [3] Moustafa Alzantot, Supriyo Chakraborty, and Mani Srivastava: *SenseGen: A deep learning architecture for synthetic sensor data generation*. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, mar 2017.
- [4] Jason W. Anderson, K. E. Kennedy, Linh B. Ngo, Andre Luckow, and Amy W. Apon: *Synthetic data generation for the internet of things*. In *2014 IEEE International Conference on Big Data (Big Data)*. IEEE, oct 2014.
- [5] Vanessa Ayala-Rivera, Patrick McDonagh, Thomas Cerqueus, and Liam Murphy: *Synthetic data generation using generator tool*. CoRR, abs/1311.3312, 2013.
- [6] Volker Bergmann: *Data generator tool*, 2013. [OnLine] Available: <http://databene.org/databene-generator>. [Accessed: 23-may-2017].
- [7] M. Castrillón-Santan, J. Lorenzo-Navarro, and D. Hernández-Sosa: *Conteo de personas con un sensor RGBD comercial*. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 11(3):348–357, jul 2014.
- [8] Hongyu Chen, Mengru Xue, Zhenning Mei, Sidarto Bambang Oetomo, and Wei Chen: *A review of wearable sensor systems for monitoring body movements of neonates*. *Sensors*, 16(12):2134, dec 2016.
- [9] Richardson CW and Wright DA: *Wgen: A model for generating daily weather variables*, 1984. US Department of Agriculture, Agricultural Research Service, ARS-8.USDA, Washington, DC. 1984.
- [10] S. Dormido, J. Sánchez, and E. Kofman: *Muestreo, control y comunicación basados en eventos*. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(1):5–26, jan 2008.
- [11] Hadi Keivan Ekbatani, Oriol Pujol, and Santi Seguí: *Synthetic data generation for deep learning in counting pedestrians*. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, pages 318–323, 2017, ISBN 978-989-758-222-6.
- [12] J. Olivares F. Espinosa, J.L. Lázaro: *Proyecto alcor: Contribuciones a la optimización del guiado remoto de robots en espacios inteligentes*. *Revista Iberoamericana de Automática e Informática industrial*, 15(4):416–426, 2018, ISSN 1697-7920.
- [13] Ronald A. Fisher: *UCI Machine Learning Repository: Iris Data Set*. <http://archive.ics.uci.edu/ml/datasets/Iris>, January 2011.
- [14] Janick V. Frasch, Aleksander Lodwich, Faisal Shafait, and Thomas M. Breuel: *A bayes-true data generator for evaluation of supervised and unsupervised learning methods*. *Pattern Recognition Letters*, 32(11):1523–1531, aug 2011.

- [15] Deepak Ganesan, Deborah Estrin, and John Heidemann: *Dimensions: Why do we need a new data handling architecture for sensor networks?* SIGCOMM Comput. Commun. Rev., 33(1):143–148, January 2003, ISSN 0146-4833.
- [16] D. Garcia-Alvarez and M.J. Fuente: *Estudio comparativo de técnicas de detección de fallos basadas en el análisis de componentes principales (PCA)*. Revista Iberoamericana de Automática e Informática Industrial RIAI, 8(3):182–195, jul 2011.
- [17] Lewis Girod, Ramesh Govindan, Deepak Ganesan, Deborah Estrin, and Yan Yu: *Synthetic data generation to support irregular sampling in sensor networks*. In *GeoSensor Networks*, pages 211–234. CRC Press, aug 2004.
- [18] Joseph E. Hoag: *Synthetic Data Generation: Theory, Techniques and Applications*. PhD thesis, University of Arkansas, Fayetteville, AR, USA, 2008, ISBN 978-0-549-66137-5. AAI3317844.
- [19] Joseph E. Hoag and Craig W. Thompson: *A parallel general-purpose synthetic data generator*. ACM SIGMOD Record, 36(1):19–24, mar 2007.
- [20] ITU-T: *Methods for subjective determination of transmissions quality*. Recommendation P.800, 1996.
- [21] Feng Jiang, Wen Gao, Hongxun Yao, Debin Zhao, and Xilin Chen: *Synthetic data generation technique in signer-independent sign language recognition*. Pattern Recognition Letters, 30(5):513–524, apr 2009.
- [22] Craig W. Thompson Josh Eno: *Generating synthetic data to match data mining patterns*. IEEE Internet Computing, 12(3):78–82, may 2008.
- [23] Narendra Kumar Kamila (editor): *Handbook of Research on Wireless Sensor Network Trends, Technologies, and Applications*. IGI Global, 2017.
- [24] Leszek Kuchar: *Using WGENK to generate synthetic daily weather data for modelling of agricultural processes*. Mathematics and Computers in Simulation, 65(1-2):69–75, apr 2004.
- [25] F. Leon, Fco. J. Rodriguez-Lozano, A. Cubero-Fernandez, Jose M. Palomares, and J. Olivares: *Sysgpr, servicio web para la generación de señales sintéticas*, 2018. [OnLine] Available: <https://www.uco.es/giia/sysgpr/>. [Accessed: 23-Apr-2018].
- [26] Fabián Muñoz: *Distribuciones poisson y gamma: Una discreta y continua relación*. Prospectiva, 12(1):99, aug 2014.
- [27] H. Nyquist: *Certain topics in telegraph transmission theory*. Transactions of the American Institute of Electrical Engineers, 47(2):617–644, April 1928, ISSN 0096-3860.
- [28] A. Ollero, I. Maza, A. Rodríguez-Castaño, J.R. Martínez de Dios, F. Caballero, and J. Capitán: *Proyecto AWARE. integración de vehículos aéreos no tripulados con redes inalámbricas de sensores y actuadores*. Revista Iberoamericana de Automática e Informática Industrial RIAI, 9(1):46–56, jan 2012.
- [29] Yaling Pei and Osmar Zaiane: *A synthetic data generator for clustering and outlier analysis*. Technical report, Department of computing Science, University of Alberta, 2006.
- [30] Taoxin Peng and Florian Hanke: *Towards a synthetic data generator for matching decision trees*. In *Proceedings of the 18th International Conference on Enterprise Information Systems*. SCITEPRESS - Science and Technology Publications, 2016.

-
- [31] Ross Quinlan: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [32] C. E. Shannon: *Communication in the presence of noise*. Proceedings of the IRE, 37(1):10–21, Jan 1949, ISSN 0096-8390.
- [33] Robert C. Streijl, Stefan Winkler, and David S. Hands: *Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives*. Multimedia Systems, 22(2):213–227, dec 2014.
- [34] Yannis Theodoridis, Jefferson R. O. Silva, and Mario A. Nascimento: *On the generation of spatiotemporal datasets*. In *Advances in Spatial Databases*, pages 147–164. Springer Berlin Heidelberg, 1999.

ÍNDICE

7.1	Abstract	60
7.2	Introduction	60
7.2.1	Specifying the Scope of Interest	61
7.3	Background	62
7.3.1	Event Detection in WSN	63
7.3.2	Aggregation and Compression Techniques in WSNs	63
7.3.3	Discussion and Hypothesis	65
7.4	Method	66
7.4.1	Data Reduction	67
7.4.2	Metrics	69
7.5	Case Study	74
7.5.1	Case Study I	75
7.5.2	Case Study II	78
7.6	Comparative Test	79
7.6.1	Increasing Linear Threshold 'K'	80
7.6.2	Send-on-Delta and Predictive Sampling	80
7.6.3	Adaptive Versions of SoD and PS	81
7.6.4	Experiment Configuration and Results	82
7.7	Conclusions and Future Work	84

D₂R-TED: Data-Domain Reduction Model for Threshold-Based Event Detection in Sensor Networks

Fernando Leon-Garcia, Jose Manuel Palomares and Joaquin Olivares

7.1 Abstract

The reduction of sensor network traffic has become a scientific challenge. Different compression techniques are applied for this purpose, offering general solutions which try to minimize the loss of information. Here, a new proposal for traffic reduction by redefining the domains of the sensor data is presented. A configurable data reduction model is proposed focused on periodic duty-cycled sensor networks with events triggered by threshold. The loss of information produced by the model is analyzed in this paper in the context of event detection, an unusual approach leading to a set of specific metrics that enable the evaluation of the model in terms of traffic savings, precision, and recall. Different model configurations are tested with two experimental cases, whose input data are extracted from an extensive set of real data. In particular, two new versions of Send-on-Delta (SoD) and Predictive Sampling (PS) have been designed and implemented in the proposed data-domain reduction for threshold-based event detection (D2R-TED) model. The obtained results illustrate the potential usefulness of analyzing different model configurations to obtain a cost-benefit curve, in terms of traffic savings and quality of the response. Experiments show an average reduction of 76% of network packages with an error of less than 1%. In addition, experiments show that the methods designed under the proposed D2R-TED model outperform the original event-triggered SoD and PS methods by 10% and 16% of the traffic savings, respectively. This model is useful to avoid network bottlenecks by applying the optimal configuration in each situation.

7.2 Introduction

Wireless sensor networks (WSN) have been widely used for monitoring environments. Nowadays, WSN are immersed in an exponential growth in many sectors of society such as e-health, industry, agriculture, or smart-cities, among others. Therefore, the amount of deployed devices is huge and increasing each day. However, as more nodes are included, larger data flows are produced. Networks get collapsed as they are not able to handle such amount of information. Thus, network traffic must be reduced.

One of the most common approaches to reduce traffic is by compression. According to Razzaque et al. [30], compression drastically reduces the energy costs, with benefits even greater than linear, since the congestion at the link-level is largely reduced, and, therefore, the effect is wider. In addition, these authors describe the compression in WSN in three levels: Sampling Compression (SC), Data Compression (DC), and Communication Compression (CC). The first type, SC, reduces the amount of sensing (or sampling) while keeping the loss of precision within an acceptable margin. Another strategy, DC, is to transform input data to a new data stream with fewer bits. They are mainly based on the fact that most data streams have redundant values and, thus, more efficient representations can be used. Finally, CC focuses on reducing the amount of communication messages (either transmissions or receptions). This last type is able to provide the largest energy savings, as communications are the most energy-consuming elements in most distributed systems.

Recent works in the state-of-the-art show that there are many lossless compression techniques used in WSN. These techniques are mainly combined with Data Gathering (DG) [21–23]. The goal of DG is to efficiently collect all the generated data from multiple deployed sensors in an environment. All of the collected data are usually sent to a central node, commonly called *head cluster*, where data are reconstructed. As identical sensors are placed to capture iden-

tical physical variables, there is a possibility of large data redundancy and, therefore, large savings can be achieved if compression techniques are applied. In fact, Data Gathering is combined with Compressed Sensing (CS) [3, 11] to produce Compressive Data Gathering (CDG) [17, 52, 53]. This hybrid scheme interconnects the well-established foundations of CS, based on the representation of a sparse dimensional signal according to fewer projections of the same signal, and DG to collect data efficiently. In order not to lose any data when reconstructing the original signal in the head cluster, the selected CS methods tend to be lossless.

The CS algorithms usually have large computational requirements. However, sensors and their hosting devices are highly constrained in terms of computational power, energy, and bandwidth, as shown in [29]. Therefore, further lightweight algorithms are included. In this sense, lossy compression methods have usually lower computational requirements. Nevertheless, a trade-off between precision, computing power, and the rest of the constraints must be taken into consideration in the selection of the compression technique.

However, WSNs are not only used to collect data, but to provide responses. These networks are called Wireless Sensor and Actuator Networks (WSAN) [5]. The responses to the actuators are generated automatically according to the system state and the information acquired by the sensors. In these cases, the sensed data can be compressed to reduce the amount of the data sent through the network, but the reconstructed data may not need such accuracy, as long as the obtained response is the same as the one that would be obtained with the original data prior to any compression. Thus, in these cases, lossy compression can be applied, with larger traffic reduction rates and very similar, or even equal, responses.

7.2.1 *Specifying the Scope of Interest*

This work considers that the data can be further compressed if it is contextualized, that is, it is known what the extracted data is used for. Therefore, in order to be able to take advantage of the compression methods, this work focuses on deployments where data is acquired and analyzed to provide responses based on the detection of evaluable conditions. Thus, this work is interested in duty cycle-based WSN for condition evaluations on the sensed data. This is quite a common scheme, for instance, WSAN deployments have this type of approach: Nodes that provide information on a given environment, some decisions are made based on all the data acquired and other internal/external variables, and a response is sent to an actuator. This actuator may be applied on the same environment or in other different ones.

In most WSN deployments, the data is sent periodically, in time-triggered schemes. However, there is the possibility of enhancing the transmission process by sending information only when a relevant fact occurs. These mechanisms are included within the event-triggered schemes. Currently, there is much interest in these types of schemes in several fields of interest, for instance, in industrial event-oriented control [8, 33], cooperative control of environmental monitoring [15], tracking trajectories of vehicles and robots [13, 25], rule-based control of Smart-Homes [4], etc. Currently, large advances have been obtained with diverse variations of the event-triggered schemes, for example, by dynamic adaptive delta mechanisms [10].

The events may provoke responses, which are Boolean in most cases. Therefore, it is possible to analyze the accuracy of the outputs both of the original

uncompressed and the event-triggered compressed streams of data. However, the most commonly used metrics in WSN [51] are not suitable by themselves alone for the described task, as they do not take into consideration the accuracy of the outputs in the value domain. In this case, two items must be taken into account: level and transitions. Level is related with the actual Boolean response, while transition deals with the instant where a change in level takes place.

Here, the scenario at hand uses the information to obtain Boolean results. Assuming the applicability of a data compression strategy, the question on its application is how it would affect the result. The analysis of these circumstances from a contextualized perspective, i.e., taking into account the system output, is the main motivation of this work. For example, if a temperature sensor is used to activate a ventilation system when it exceeds a certain threshold, any reduction of the amount of information about the temperature signal can change the triggering pattern of the fans, but it can also reduce the amount of data required for system operation. Therefore, any temperature value that will not change the system state may be ignored because they are not relevant for the output in most cases. This reduction of data is translated into a cost-benefit ratio, which emerges from the contextualized perspective. How to model it is a fundamental motivation of this work.

This paper is structured as follows: Section 7.3 provides an overview of event detection, compression techniques, and some related work. Section 7.4 lays the foundation for our proposal. Here, the mathematics behind the data reduction is described. It also describes the metrics for evaluating the data reduction, contextualized to the detection of threshold-based events. Section 7.5 addresses two case studies to exemplify data reduction with a simple configuration. The reduction process is analyzed in terms of traffic savings and response quality according to the proposed metrics. Section 7.6 proposes a comparative study to demonstrate the benefits of the model. The properties of the model are used to adapt two known event-based sampling techniques and improve them in the context of the evaluation of Boolean conditions. Finally, some conclusions and future work are pointed out in Section 7.7.

7.3 Background

WSNs are typically used to collect data from the environment on an ongoing basis. The simplest design is to provide all nodes with a cyclical routine of activation, reading, sending, and deactivation. Over time, the uses of WSNs have multiplied, and the solutions are increasingly sophisticated, generating a large variety of networks according to different criteria [29].

This work focuses on the use of WSNs for detecting threshold-based events, aiming to provide an analytic way to model data compression techniques in that scope, by contextualizing the effects of the compressed data on the Boolean result.

Thus, this work is based on two fundamental concepts: event detection and data compression techniques in WSNs. The following subsections delve into both in order to establish a more comprehensive framework. A third subsection describes the scope of this work as the combination of both concepts, specifying the cases of use and clarifying the specific goal.

7.3.1 Event Detection in WSN

An event is an occurrence or a significant activity that is unusual in relation to normal patterns of behavior [5, 48]. Applying the concept to a physical environment, the distributed nature of the sensors combined with their reduced cost and size makes WSNs ideal for monitoring environments in order to detect events.

In [65], the authors point out the challenges of this task in WSNs by organizing them into four groups: context dependency, application criticality, heterogeneity and overcrowding of data sources, and network topology. In addition, they classify the methods into three groups: statistical, probabilistic, and based on automatic learning techniques and artificial intelligence. According to these authors, the most widely used technique is the detection of events based on a static threshold. It consists of the systematic checking of the condition to know whether the parameter value is within known ranges.

Threshold-based event detection has been used since the earliest declarative systems, which processed Boolean entries known as “facts” [4]. However, the technological framework of the Internet of Things (IoT) has brought about the need to rethink the computation of these Boolean variables in systems where the computational load is distributed in restricted processing cores. On the other hand, it has also brought the possibility of detecting events which are difficult to evaluate with classic centralized systems, e.g., early warning systems for which multiple sensors are required [18] or hazardous environment [28].

Detecting events with WSNs is a problem that has multiple approaches. For example, in [34], they classify different proposals according to whether the notification decision of the event is made on a sink node or on the node itself. However, the authors of [35] take into consideration the network architecture, the scope of the decision, and the nature of the detection algorithm.

Within the framework of event detection, it is important to note the appearance of Complex Event Processing (CEP) in recent years. This discipline arises from the need to enrich the semantics of conditions in declarative systems, and introduces the concept of a temporal and orderly succession of simple events [43]. The author of [90] proposes various techniques for performing CEP techniques at WSN. As mentioned above, the threshold event detection technique is widely used, a clear example of which is the need for simple event calculation to assess complex events.

The development of concrete applications for the detection of events in WSN or IoT is becoming increasingly importance in recent years [6, 12, 14, 35, 44, 46, 47]. These systems optimize the data transmission related to events by setting filters under different criteria, obtaining large reductions of traffic. As tailor-made solutions, they are an excellent example of good use of the network. However, this type of proposal requires specific analysis of each environment and network deployment in order to establish the general measurement of benefits.

7.3.2 Aggregation and Compression Techniques in WSNs

As mentioned above, a key aspect of this work is to reduce the use of the network accordingly. Here, the most relevant scientific contributions in the field of traffic data reduction WSNs are summarized.

Razzaque et al. [30] provided a survey in which compression techniques were classified into three types: Sampling Compression (SC), Data Compression (DC), and Communication Compression (CC).

Many WSN devices are designed to provide data periodically. Moreover, redundant sensors are usually deployed in an environment. Devices may combine their own data with the data received from others, either providing a stream of modified data, or a value, as the result of a mathematical operation. This is called Data Aggregation (DA). This aggregation results in a reduction of the transmitted data. This strategy can be included within the DC type, as it modifies the data to reduce the amount of data in the network, although some kind of computation and processing is applied in-route on the data.

Communication Compression

An interesting taxonomy of congestion control techniques in WSNs is proposed in [74]. These techniques are based on the detection or prediction of network congestion conditions and the application of more or less generic strategies to control the situation. This type of proposal is closely linked to infrastructure, and offers routing solutions in most cases. While the purpose of this work is not to control congestion (but to reduce traffic by limiting the loss of response quality), some strategies end up implementing traffic reduction policies to control congestion [20, 31, 41, 42].

Data Compression

Solutions to the congestion problem consider the *network as a service* for data exchange, so they have no control over the sending data. However, there are data reduction techniques that work more closely with information. The authors of [8] delve deeper into predictive techniques for data reduction in WSNs, classifying models according to the prediction method and the network element where it is applied. The prediction models that are generated in the cluster head nodes, ([24, 48, 86], where correlation analysis techniques are used to assess the importance of the data, are particularly noteworthy.

Ikjune et al. [50] proposed an aggregation and compression data method to use in a solar-powered WSN. In it, nodes aggregate data to obtain maximum compression. Data are compressed using a lossless scheme (LZW coding). If the node has more energy coming from its solar panels than it can store in its batteries, aggregated data are sent. Otherwise, it just stops sending data, and just senses its environment, storing the values in its internal memory. This method makes use of the DC type, along with the DA mechanism. However, data are not processed in-route. In addition, the CC is connected to their internal energy status, and not to the Quality of Service (QoS) of the network.

Sampling Compression

As a reference work, the sampling techniques exposed in [39], a work that is framed in the discipline of control based on events, is worth highlighting. This article discusses the Send-on-Delta [57] technique, which is intended as a fixed increase in the magnitude of a signal whose consecutive samples must be tested in order to be taken into account. In the model proposed in this work, a similar mechanism is proposed but mathematically posed from the domain of system inputs definition, and not subject to static magnitudes.

An important line of research for compressing data is applying PCA to WSN [50, 86], in order to reduce the amount of data transferred. There are many sensors providing data that may be correlated. If a PCA is applied to the acquired data space, the dimensionality of the data can be reduced to only a few representative values. The main use for this mechanism is to get rid of several sensors providing similar readings that can be correlated.

Shu et al. [38] proposed DDASA to improve the power efficiency while ensuring the accuracy of the data and have recently stated that “Data reduction exploits the fact that depending on the characteristics of the sampled data within the environment, some data could be redundant”. However, that DDASA algorithm differs substantially from the work presented in this manuscript. In DDASA, the sampling frequency is dynamically and explicitly modified.

Event-Triggered Sampling

In recent years, non-periodic signal sampling techniques have been developed. These techniques propose an alternative approach to determine the sampling instant, instead of the strictly temporal one. For this purpose, strategies based on events intrinsic to the evolution of the data are introduced.

The Send-On-Delta concept (SoD) is the simplest strategy. By means of this scheme, the sampling instant occurs when the magnitude of the difference between the signal value and the previous sample exceeds a confidence interval called delta [57]. This simple concept opened the door to a vast field of research under different topic names, such as event based sampling, Lebesgue sampling, magnitude based sampling, etc. [27, 32].

Within the framework of automatic control engineering, event-based sampling techniques are still being developed nowadays. With the mathematical complexity that this discipline requires in terms of stability and convergence of control systems, the sampling criteria have been sophisticated by means of predictive techniques. A simple example of this type of input is [40], in which a linear predictor is used to establish the sampling instant with the estimated sample, instead of the previous sample.

Considering a distributed scenario where signal samples are sent for processing, these techniques have been considered for traffic reduction, since they limit the amount of network packets tolerating a certain loss of information. An example of this approach is proposed in [10], where a technique able to select dynamically the limits (delta) according to the available transmission-rate is introduced.

7.3.3 Discussion and Hypothesis

Reducing traffic in sensor networks is a goal that has been addressed from many perspectives. The wide variety of sensor networks results in a broad range of traffic reduction possibilities. To give a few examples: if the network is multihop, the routing policy can contemplate traffic reduction; if the sensor information has some redundancy, traffic can be reduced by aggregating data with space-time criteria; if the network does not have real-time requirements, buffering techniques can be used to collect data on the route, reducing sent packets, etc. In each of these examples, the logic of data reduction is based on a system feature that offers an opportunity to save traffic. This feature is a limiting factor, a requirement. Any data reduction technique is limited by the

requirements of its scope.

This work focuses on sensor networks with very specific restrictions. The most important are the topology, the hierarchy of nodes, the timing, and the specific purpose of the data collection. In this framework, there is no information redundancy, there is no multi-hop, and the sensors have very little computational capacity. This means that techniques based on aggregation and compression are not the most suitable for this purpose.

Among the techniques reviewed, event-based sampling ones are applicable to the proposed scenario. They require little computational capacity, and traffic is avoided at the source, thus topology and hierarchy are not a problem. However, event-based sampling approaches are clearly oriented to automatic control engineering, being addressed from the mathematical requirements of this discipline, which requires demonstrating stability, convergence, and other concepts exclusive to its paradigm. This approach is difficult to adapt to other application scenarios, such as many areas of computing where there are no control loops and the versatility of abstraction is fundamental.

A common feature of the revised techniques is that the network has a general purpose, since the context of the data is not considered. In the case of error-tolerant techniques, these are measured as the difference in magnitude between the original data and those obtained after the technique reduction process. This concept of error is intrinsic to the measurement of a real-world signal, and is also decontextualized from the purpose of the information.

Inspired by event-based sampling techniques, this paper proposes a different approach to traffic reduction in sensor networks applied to threshold-based event detection. On the one hand, event-based sampling techniques are reformulated as software filters applied after a regular periodic sampling process, rather than as alternative sampling processes. This means that it is considered a periodic sampling process in the sensors, and the applied techniques will decide sample by sample whether it is sent or not. This approach seems to be simpler and closer to actual sensor network deployments. On the other hand, as part of this reformulation, the necessary abstraction is introduced to adapt the techniques to any data structure, not only to real numbers. This feature makes traffic reduction adjustable to any application, providing versatility. Finally, it is proposed to contextualize the use of information in order to optimize traffic avoidance. As mentioned above, the focus of this work are those networks that evaluate threshold-based events. Based on this knowledge, the concept of error can be redefined according to this context, enabling the definition of traffic avoidance techniques that take into account the system result. As a working hypothesis, let us consider that this approach would provide better cost-benefit rates in terms of traffic savings and errors.

The above approaches are integrated into a general model that can be instantiated for any reduction technique according to its limitations. This model defines the reduction technique using a generic and adaptable mathematical procedure. The effects of this reduction are measured using proposed metrics to take the Boolean result into account. The generic reduction model and the mentioned metrics are detailed in the following section.

7.4 Method

The study scope is based on those wireless sensor networks which are used for the computation of Boolean conditions. Under the proposed approach, the

data flow can be represented as a network structure organized in a tree. The root node performs the calculation of the condition by applying an equation whose inputs are provided by the rest of nodes, directly or indirectly, as shown in Figure 7.1.

Some simplifications will be considered for this study:

- Condition evaluation is performed periodically with a t_s duty cycle period.
- Between two consecutive evaluation instants, the nodes will send an update of their variable only if it has changed.
- All inputs share the same data domain, defined by Φ .

Note that the condition evaluation involves a domain change that takes place at the root node, from Φ^n to \mathbb{B} (Boolean domain). This fact contextualizes the use of the input data, and it will allow us to define specific metrics for the analysis of the results.

This section is organized into two subsections. The first one describes a generic mathematical definition for the reduction of input data, with the aim of reducing network traffic. The second one proposes metrics for evaluating the response obtained as a result of this reduction in terms of message savings, precision, and recall.

7.4.1 Data Reduction

One of the motivations of this work is to propose a solution which could be applicable to any system whose purpose and deployment fit in the above-described structure. For this reason, the reduction method is founded on the mathematical description of the domains of the data definition.

Consider the minimum network that fits into the structure described in Figure 7.1. A single root node called $\alpha\beta$ that evaluates a condition whose entries are provided by n nodes denoted by α_i . The entire network can be modeled as a system with n inputs and an output in the following terms:

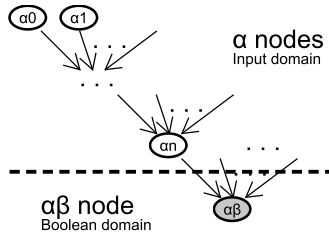


Figura 7.1
Tree structure.

Definition 1. Let Φ be the domain of the values of the system inputs, and let \mathbb{B} be the Boolean domain $\{true, false\}$.

On that basis, the $\alpha\beta$ -node with n α -nodes as inputs can be described by a state function $s : \Phi^n \rightarrow \mathbb{B}$. Assuming that the α -nodes obtained their data samples at the same frequency $f_s = 1/t_s$, the system results can be modelled during a time T by Equation (7.1):

$$s : \Phi^n \rightarrow \mathbb{B}, \forall \varphi \in M_{f,T \times n}(\Phi), \exists b \in M_{f,T \times 1}(\mathbb{B}) \quad | \quad (7.1)$$

$$b_x = s(\varphi_{x,0}, \varphi_{x,1}, \dots, \varphi_{x,n-1}),$$

where $M_{m \times n}(D)$ is a matrix of m rows and n columns of elements defined in D , and $a_{x,y}$ is the element at the row x and column y of a matrix $a \in M_{m \times n}$.

According to the simplification criteria outlined above, α -nodes only send data when information changes. Since the number of unsent packets due to repeated values is unknown, a maximum of $n \cdot f_s \cdot T$ network packets are sent in the T analysis period. Based on this approach, a mechanism to promote data repetition is proposed to introduce a reduction in network traffic.

Definition 2. Focusing on this issue, R is introduced as the family of functions that, through a set of parameters defined by Δ , translate a set of j -values of Φ , Φ^j , into a set of k -values of Φ' , Φ'^k , $j, k = 1 \dots \infty$. As Φ' tends to be a smaller set than Φ , these functions are called reduction functions in the rest of the paper:

$$R: \Phi^j \times \Delta \rightarrow \Phi'^k \quad | \quad \Phi' \subseteq \Phi. \quad (7.2)$$

Thus, giving a function $r \in R$ and the domain of its parameters $\Delta_r \subseteq \Delta$, we obtain Φ_r , the reduced input domain, as defined below:

$$r: \Phi^j \times \Delta_r \rightarrow \Phi_r^k \quad | \quad \Phi_r \subseteq \Phi.$$

Furthermore, every $\delta_r \in \Delta_r \subseteq \Delta$ generates a different vision of the Φ domain, as described by Equation (7.3). Figure 7.2 represents conceptually the reduction of Φ by an $r \in R$ function and its Δ_r parameters:

$$r: \Phi^j \times \delta_r \rightarrow \Phi_{r\delta}^k \quad | \quad \Phi_{r\delta} \subseteq \Phi_r \subseteq \Phi. \quad (7.3)$$

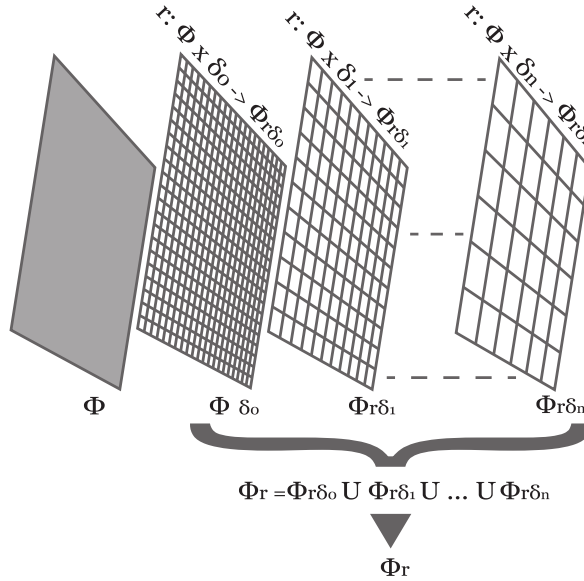


Figura 7.2
 Φ domain reduced to a set of $\Phi_{r\delta}$ domains ($j = k = 1$).

If an injective r function and a set of parameters Δ_r are properly selected, in such a way that the resulting domains have different number of elements, a reduction process is obtained by ordering them by descending cardinality (shown in Figure 7.2). In this process, different elements of any domain correspond to a single element of the next one.

For the sake of simplicity, but without loss of generality, for the rest of the article, it will be assumed that $j = k = 1$ for the input domains, that is, only one input value is considered in the *reduction functions*. On this basis, given a sequence of values defined in $\Phi_{r\delta_i}$ with no consecutively coincident values, it is possible that the same signal translated to $\Phi_{r\delta_{i+1}}$ may present repetitions, especially if the sequence corresponds to a continuous signal and $\Phi_{r\delta_i}$ elements are combined in $\Phi_{r\delta_{i+1}}$ by proximity. At the same time, reducing the cardinality of the signal domain implies a loss of information that will provoke some alterations in data whose evaluation depends on this signal.

In order to show the general applicability of the proposed model and to help for a better understanding of this concept, the following example has been included. Let us consider the definition of Φ domain as triads of 1-byte-coded integers, representing the RGB color domain. In addition, consider a reduction function (r) that receives as a reduction parameter (δ) the number of bits to encode each integer of the triad.

Thus, for each $\delta < 8$, any result of r is within a new domain $\Phi_\delta \subset \Phi$, $\Phi_8 = \Phi$ being the original domain. Defining the set of possible reduction parameters in descending order ($\Delta = \{8, 7, 6, 5, 4, 3, 2, 1\}$) gives a scale of domains by cardinality as described in Figure 7.2.

The top of Figure 7.3 shows a representation of the reduced domains of the example for $\delta = 5$, $\delta = 3$, and $\delta = 1$, which are Φ_5 , Φ_3 , and Φ_1 , respectively. At the bottom of the figure, different versions of a $X \times Y$ matrix of colors are shown. That is an image, and each version is a representation of its pixels (colors) when they are defined in the corresponding domain. In the example described above, each value is a color, and this simplification has allowed the input domain to be represented in a figure. However, let us now consider the domain of $X \times Y$ matrix of values defined in Φ as input domain ($\Phi' : M_{X \times Y}(\Phi)$) and the same reduction function and parameters applied to all pixels of the matrix. In this case, a time-ordered sequence of periodic values defined in Φ' is a video, frame by frame. Coming back to the problem of reducing network traffic, it seems reasonable to assume that, if a remote camera implements this reduction procedure without sending duplicate frames, δ parameter will influence the consecutive equivalent frames statistics and, by extension, the network traffic. The question is how big its effect is, and the answer depends largely on what the camera is recording. The variability between consecutive frames will be much lower if the camera monitors a mall at night than if it records a bike ride from the helmet of the cyclist.

The rise of duplicates by reducing the domain cardinality and subsequent result alterations are phenomena with underlying probabilistic logic. However, the analysis would be too complex, since it would have to take into account different aspects, such as signal characteristics, the sequence of domains generated by r and Δ_r , and all the calculations in which the signal is involved. If the possible reduction of all system signals is taken into account, it is analytically unaffordable. For this reason, this proposal bases the evaluation of the model on empirically obtained metrics detailed in the following subsection.

7.4.2 Metrics

The application of the data reduction method will have two consequences. First, the number of network packets will be reduced due to the promotion of duplicate values in the data streams in the α -nodes. Second, the evaluation of the condition in the $\alpha\beta$ -node will suffer alterations due to the loss of

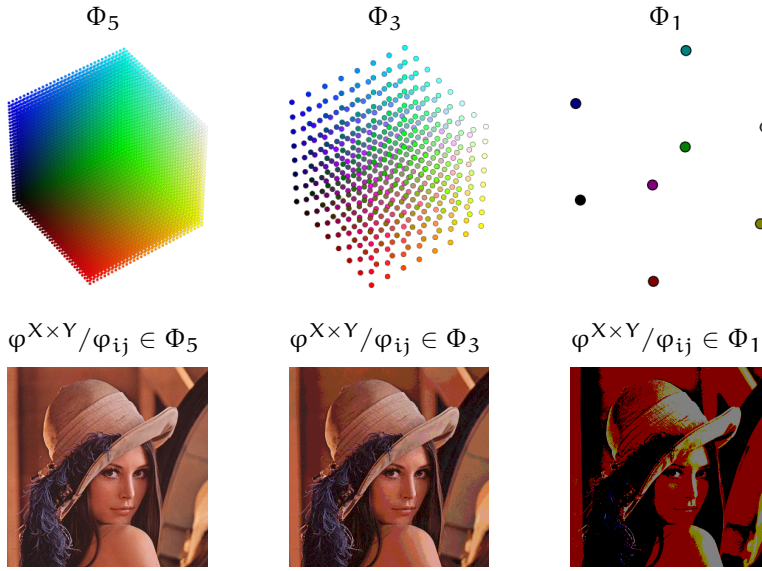


Figura 7.3
Example of RGB color domain reduction. Components of color coded with 5, 3 and 1 bit.

information. At this point, analytic methods to quantify both effects will be proposed.

Traffic Reduction Metrics

To obtain the amount of traffic avoided, a ratio between the number of variations present in the sample sequences at the α -nodes applying the reduction with respect to the number of variations that present the same sequences without reduction is proposed. The proposed metric is called *Traffic Saving Ratio* (TSR) and is defined as follows.

Definition 3. Let $\varphi \in M_{m \times n}(\Phi)$ be a matrix of m rows and n columns of values defined in Φ , $\varphi_{x,y}$ being the element of the row x and the column y , and $m = T \times f_s$. Let us also define $r \in \mathbb{R}$ as the domain reduction function, and $\delta \in \Delta_r$ the selected reduction parameters. In addition, let be $\varphi' \in M_{m \times n}(\Phi_{r\delta}) / \varphi'_{x,y} = r(\varphi_{x,y}, \delta)$. The *Traffic Saving Ratio* (namely, TSR) can be defined as the proportion of repetitions of values that appear using the reduced domain. The calculation procedure is stated in Equation (7.5). Equation (7.4) is required for TSR calculation, and it is a key part of the data reduction model, since it implements the equivalence criterion of two values defined in the Φ domain. As described in the previous section, this criterion is applied to consecutive values to avoid sending packets in case of equivalence. Note that modifying this equation allows the model to be extended, for example, to fuzzy logic:

$$\text{eq}(\varphi_1, \varphi_2) = \begin{cases} 0, & \text{if } \varphi_1 = \varphi_2, \\ 1, & \text{if } \varphi_1 \neq \varphi_2, \end{cases} \quad (7.4)$$

$$\text{TSR} = 1 - \frac{\sum_{i=1}^{m-2} \left[1 + \sum_{j=0}^{n-1} \text{eq}(\varphi'_{i-1,j}, \varphi'_{i,j}) \right]}{\sum_{i=1}^{m-2} \left[1 + \sum_{j=0}^{n-1} \text{eq}(\varphi_{i-1,j}, \varphi_{i,j}) \right]}. \quad (7.5)$$

Result Evaluation Metrics

To measure the effects on the result, it is interesting to briefly discuss the concept of error in this context. According to the condition Equation (7.1), the variables are values defined in Φ , and the result is a Boolean value. The error of Boolean values is not quantifiable in terms of magnitude or distance because there are only two possibilities: true or false. In this sense, in each sampling turn, the accuracy of the result can only be measured in terms of correct or incorrect. According to this logic, it is possible to calculate a simple statistic of the number of erroneous results obtained over a period of time. Definition 4 is based on this concept, which can be considered a starting point for the computation of the system precision.

Definition 4. Based on Definition 3, let us define $s : \Phi^n \rightarrow \mathbb{B}$ as the condition function implemented in $\alpha\beta$. In addition, let $b, b' \in \mathbb{B}^m / b_x = s(\varphi_{x,0}, \varphi_{x,1}, \dots, \varphi_{x,n}), b'_x = s(\varphi'_{x,0}, \varphi'_{x,1}, \dots, \varphi'_{x,n})$ be the system response using the original and the reduced input domain, respectively. Precision (namely, P) can be defined as the proportion of condition results that, when evaluated with input data defined in the reduced domain, match those obtained when using the original domain. Equation (7.6) details the calculation:

$$P = 1 - \frac{\sum_{i=0}^{m-1} (b_i \oplus b'_i)}{m} \quad (7.6)$$

Although this metric is mathematically rigorous, the sample-by-sample evaluated precision is appropriate only if the input data are statistically independent in a sample-by-sample basis. Therefore, each sample is completely independent from the previous one and from the following one. In the scenarios of interest for this research, the input data are samples of continuous signals. Thus, input values correlate with their predecessors, in the same way that the resulting Boolean signal presents transitions between stable levels. In addition, if the reduction method is chosen appropriately, an input value reinterpreted in a reduced domain involves a limited variation from the original one. These facts make it reasonable to consider that most of the errors of the result signal will occur around the transitions, delaying them or bringing them forward in time. To illustrate this, Figure 7.4 represents on the left side a continuous signal ($F(x)$) and the result of an inequation applied to it ($F(x) > \text{Threshold}$). The same signal, but with its samples transferred to a reduced domain, is displayed on the right side. Below, the result of the same inequation for the reduced signal, highlighting the different samples with respect to the result obtained with the original signal is shown. As it can be observed, the differences are grouped around the transitions. This effect has such an important impact that it should be measured.

Based on this idea, the concept of error can be discussed in each application scenario. If the detection of the condition has no real-time requirements, a certain delay in transitions may be tolerable, not being considered an error. For example, returning to the example of the remote camera, if the condition is whether an image recognition algorithm detects a face or not, it may be acceptable for the detection to occur a few frames before or after. To give another example, a network for home automation control that measures the outside temperature to close windows or activate air conditioning can tolerate a delay of some sampling periods to perform the corresponding action.

Here, three metrics that consider this aspect are proposed. These metrics allow for a more descriptive analysis of the consequences of reducing traffic with the proposed method, including a variable, τ , that models the tolerable delay in each scenario. On the one hand, two of the metrics analyze the correspondence between the transitions of the original and the reduced signal. As

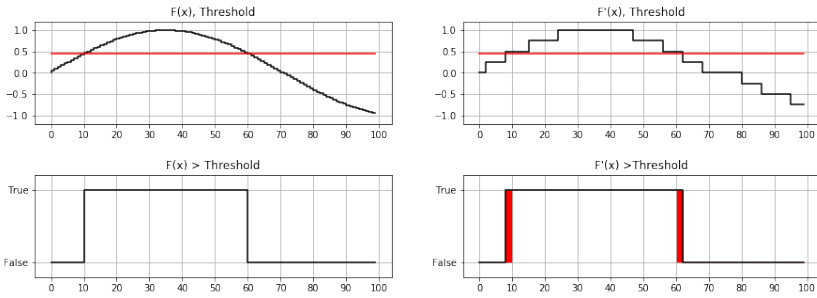


Figura 7.4

Upper left: Original signal $F(x)$, in black; and a constant threshold (Threshold), in red. Bottom left: Boolean response of the condition $(F(x) > \text{Threshold})$ applied to the original signal and threshold. Upper right: Reduced signal $F'(x)$, in black; and a constant threshold (Threshold), in red. Bottom right: Boolean response of the condition $(F'(x) > \text{Threshold})$ applied to the reduced signal and threshold. In red, Boolean variations with the obtained in the bottom left image.

transitions may appear on one of the signals without matching on the other, this can be analyzed in terms of precision (proportion of obtained transitions that are present in the original signal) and recall (proportion of transitions of the original signal that are obtained). On the other hand, a third metric analyzes the correspondence between signals outside the proximity of transitions, sample by sample.

These three metrics are called *Transition Precision*, *Transition Recall*, and *Level Precision*. For the sake of a better understanding of these, Figure 7.5 represents how the proposed metrics are calculated. For the computation of the transition metrics, the signal value changes are parsed one by one. This process is done with the original signal to obtain the *Transition Recall*, and with the reduced signal to obtain the *Transition Precision*. For each transition, it is checked whether there is an equivalent change in the other signal. By means of the parameter τ , a window of $\pm\tau$ sampling periods is established, within which the equivalent transition will be considered as valid. In (A) and (B), this analysis is represented for $\tau = 0$ (the most restrictive case) and for $\tau = 1$, respectively. All other signal samples that are not transitions and are not within any $\pm\tau$ interval are taken into account in *Level Precision* metric. This metric evaluates the occurrence of situations such as those illustrated in (C), where the reduced signal presents erroneous states over time. To exemplify the importance of this metric, consider that two signals without transitions but with opposite values would present 100% transition precision and recall, but 0% level precision.

With regard to the interpretation of the metrics, if a training signal is properly selected, they can be interpreted as probabilities. Thus, the *Transition precision* is the probability that a transition obtained in the reduced signal would have been obtained without reducing it; the *Transition recall* is the probability that the transitions of the original signal will be detected in the reduced one; and *Level Precision* is the probability that the status of the reduced signal is correct after τ samples since the last transition. The level accuracy has a very interesting interpretation when this is maximum (100%). When this happens, it means that there are no errors outside the $\pm\tau$ transition intervals, which means that the possible erroneous states of the reduced signal will not cover more than $2\tau + 1$ samples. To understand this, note the upper right side of the illustration (C). As it can be seen, with $\tau = 1$, a possible transition to an erroneous state that is not detected in the *Level Precision*, could not cover more than three samples, since a fourth erroneous sample would lay outside the $\pm\tau$

intervals, and would be computed as an erroneous sample.

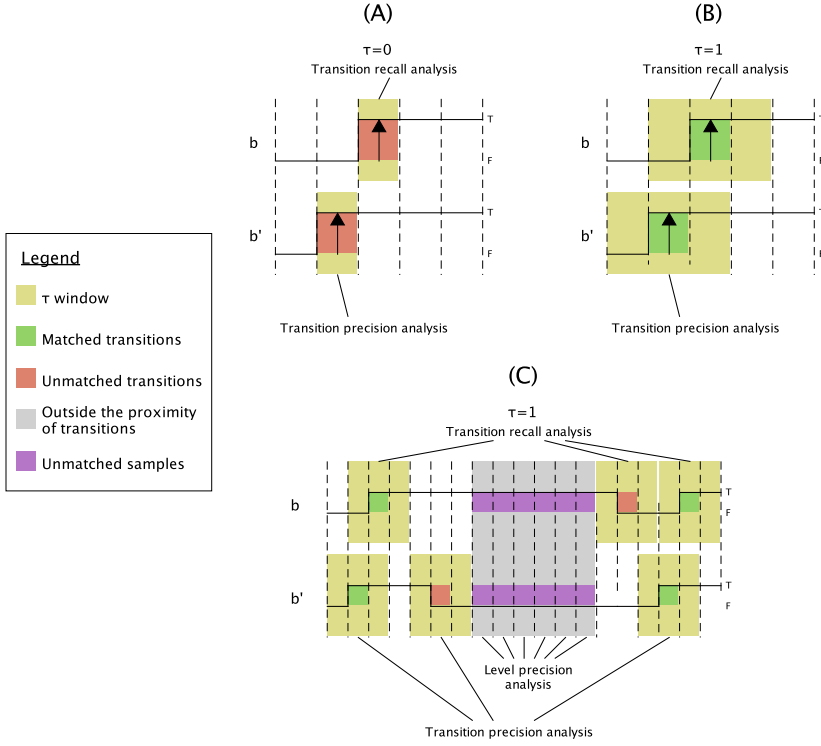


Figura 7.5

Illustration of the metrics analysis process. (A) Transition recall analysis with $\tau = 0$. In red, unmatched transitions; (B) Transition recall analysis with $\tau = 1$. In green, matched transitions; (C) Extended transition recall analysis with $\tau = 1$. In green, matched transitions; in red, unmatched transitions; in purple, unmatched samples.

The three metrics calculations are detailed in Definitions 5 and 6.

Definition 5. Based on Definition 4, let $b_T, b_F, b'_T,$ and b'_F be the resulting signals of applying Equations (7.7) and (7.8) to b and b' , respectively. These signals are transition masks to *true* and *false* and are used for the following metrics:

$$\forall b \in B^m \exists b_T \in B^m \mid b_{Ti} = \begin{cases} b_i & \text{if } i = 0, \\ (b_i \oplus b_{i-1}) \cdot b_i & \text{if } i > 0, \end{cases} \quad (7.7)$$

$$\forall b \in B^m \exists b_F \in B^m \mid b_{Fi} = \begin{cases} |b_i - 1| & \text{if } i = 0, \\ (b_i \oplus b_{i-1}) \cdot b_i & \text{if } i > 0. \end{cases} \quad (7.8)$$

Definition 6. According to Definition 5, $\tau \in \mathbb{N}$ being the time offset tolerance margin in number of samples, let us define the Transition Precision (TP) as the proportion of error-free transitions in relation to the total number of reported transitions; the Transition Recall (TR) as the proportion of reported transitions in relation to the total number of transitions to be reported; and the Level Precision (LP) as the proportion

of error-free samples measured outside $\pm\tau$ interval of any transition. The calculation procedures are stated in Equations (7.9)–(7.11):

$$TP = 1 - \frac{\sum_{i=0}^{m-1} \left(b'_{Fi} \cdot \left(b'_{Fi} \oplus \min \left(\sum_{j=i-\tau}^{i+\tau+1} b_{Fj}, 1 \right) \right) + b'_{Ti} \cdot \left(b'_{Ti} \oplus \min \left(\sum_{j=i-\tau}^{i+\tau+1} b_{Tj}, 1 \right) \right) \right)}{\sum_{i=0}^{m-1} (b'_{Fi} + b'_{Ti})} \quad (7.9)$$

$$TR = 1 - \frac{\sum_{i=0}^{m-1} \left(b_{Fi} \cdot \left(b_{Fi} \oplus \min \left(\sum_{j=i-\tau}^{i+\tau+1} b'_{Fj}, 1 \right) \right) + b_{Ti} \cdot \left(b_{Ti} \oplus \min \left(\sum_{j=i-\tau}^{i+\tau+1} b'_{Tj}, 1 \right) \right) \right)}{\sum_{i=0}^{m-1} (b_{Fi} + b_{Ti})} \quad (7.10)$$

$$LP = 1 - \frac{\sum_{i=0}^{m-1} \left((b_i \oplus b'_i) \cdot \left| \min \left(\sum_{j=i-\tau}^{i+\tau+1} (b_{Tj} + b_{Fj} + b'_{Tj} + b'_{Fj}), 1 \right) - 1 \right| \right)}{\sum_{i=0}^{m-1} \left(\left| \min \left(\sum_{j=i-\tau}^{i+\tau+1} (b_{Tj} + b_{Fj} + b'_{Tj} + b'_{Fj}), 1 \right) - 1 \right| \right)} \quad (7.11)$$

As it can be seen, to characterize a system with these metrics, the non-reduced result is required. This implies the need for training data. The effectiveness of the method will depend on how well these data represent the overall behavior of the system.

Regarding the quality of the training data, it is reasonable to assume that the most favorable scenario is the evaluation of conditions based on periodic pattern parameters. For example, parameters related to biological activities influenced by circadian cycles, or meteorological parameters that depend on natural periods. However, since the circumstances that lead to traffic avoidance are not necessarily permanent, we suggest that the entire time of the unrestricted traffic network may be used to fine-tune the metrics to obtain the most reliable model possible.

7.5 Case Study

This section includes two experimental scenarios in which the model is applied and analyzed. Through these use cases, two objectives are intended. First, to exemplify the application of the model over actual and extensive data. Secondly, to test the potential use of the technique to reduce the traffic of a network in a saturation situation, knowing what proportion of traffic will be avoided and how many errors will be introduced into the system.

For these purposes, the data signals from the public repository of the Italian meteorological network Arpa Piemonte [7] are used. This repository contains nine years of hourly data from several stations, and has been considered adequate due to the abundance of quality data.

Each use case presents an experimental scenario consisting of a conditional expression involving signals from the repository. Each signal represents a network sensor, and each signal sample represents a sent data packet. The details of the reduction according to the model requirements are described for each scenario. In order to analyze the potential on-demand application of the model (the second objective), the signals from the repository are split into two periods. The first period corresponds to the first year of data and is considered as a training period. It is assumed that during this period the data reduction process is modeled according to the results obtained. The remaining eight years are considered the test period. During this period, it is assumed that the network suffers saturation conditions, and the data reduction is applied according to the results of the training period. The comparison of the proposed metrics obtained in each of these periods constitutes an analysis of the repetitiveness of the model and, therefore, of its potential use on demand to reduce traffic knowing the consequences, which is the working hypothesis of these experiments.

First of all, let us briefly summarize the proposal to contextualize the study cases: the starting point is a network of sensors whose data is collected by a central node that evaluates a Boolean condition (called s function in this work). The model defines a procedure for reducing network packets by promoting repeated values, which are not sent. The starting point is an input data domain (Φ), which, using a reduction function (r) and a set of parameters (Δ_r), gives rise to a domain scale Φ_r (one for each $\delta_r \in \Delta_r$ parameter) with different cardinalities. Using a set of ordered values defined in Φ , the reinterpretation of this information in any $\Phi_{r\delta}$ ($\in \Phi_r$) domain has two consequences: the promotion of duplicate values and the loss of information. Metrics have been proposed to model these effects.

According to the above, the model introduces some generalities that must be defined for each scenario: (1) input data domain Φ , (2) condition expression s , (3) sensor data for training and testing, (4) reduction function r , and (5) set of parameters for the reduction function Δ . These aspects are specified for each case study, which are detailed in the following subsections.

7.5.1 Case Study I

For this experimental environment, the model has been adapted for a condition tree involving two temperature sensors ((1) $\Phi = \mathbb{R}$) and the sink node. The sink node periodically collects the information and checks whether one temperature is higher than the other ((2) $s(\varphi_1, \varphi_2) = \varphi_1 > \varphi_2$). The model metrics are calculated for two segments of these signals. The first segment is the first year of data, and corresponds to the time window considered for training. This training period is used to characterize the cost-benefit ratio in terms of traffic savings and errors introduced by the reduction technique in its different configurations. The second segment includes the remaining eight years, and its results are used to analyse the repeatability of the results obtained in the training period. Results of both periods are compared to discuss the hypothesis. The reduction function truncates the values to a multiple of a parameter, which is the temperature resolution ((4) $r(\varphi, \delta_r) = \text{Integer}\left(\frac{\varphi}{\delta_r}\right) \cdot \delta_r$). The set of possible resolutions (Δ) is $\{0.1, 0.25, 0.5, 1.0, 2.5\}$, 0.1 being the original resolution (non-reduced case) (5). This scale is typical for oscilloscopes and measuring instruments. Regarding the sensor data, from the repository mentioned above, the temperature data from the Alessandria and Montaldo Scarampi stations were used. The selection criterion corresponds to the proportion of true and false in the results of the evaluation of the previously stated condition.

Specifically, both signals are composed of 78,840 samples; the signal resulting from the condition, sample by sample, presents 7625 transitions; and the true and false balance is 53%/47%, respectively. For each $\delta \in \Delta_r$, the values of the metrics given in Section 7.4.2 have been calculated for both training and experimental data.

The results are presented in a table for each metric. By columns, $\delta \in \Delta_r$ parameters from $\delta = 0.25$ because $\delta = 0.1$ models the original system state, with no reduction, in which case $\text{TSR} = 0$, $\text{LP} = 1$, $\text{TP} = 1$, and $\text{TR} = 1$. The last two rows show the absolute and relative error for each δ value (ϵ and ϵ_r). In the case of LP, TP and TR, these errors are averaged ($\bar{\epsilon}$ and $\bar{\epsilon}_r$) and represented with standard deviation (σ). In addition, each τ value results in a row, and the last two columns represent the absolute and relative error averaged by each τ , with standard deviation as well. The cells in the body of the tables show two values separated by a slash (/): on the left, the result of the calculation with the training data, and on the right of the slash, the results applied to the experimental data (not including the data used for training). Relative errors are calculated considering the value obtained with the training data as the actual value.

Table 7.1 shows the traffic saving ratio metric (TSR), which has been calculated according to Equation (7.5). Table 7.2 shows the level precision, according to Equation (7.11). Table 7.3 shows the transition precision, according to Equation (7.9). Finally, Table 7.4 shows the transition recall, according to Equation (7.10). The last three metrics have been calculated with an offset tolerance interval (τ) from 0 to 9. The results are discussed below.

	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$	$\delta = 2.5$
	0.069/0.066	0.201/0.186	0.307/0.291	0.393/0.373	0.677/0.666
ϵ	0.003	0.015	0.016	0.020	0.011
ϵ_r	4.35%	7.46%	5.21%	5.09%	1.62%

Tabla 7.1
Case study I. Traffic Saving Ratio (TSR).

	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$	$\delta = 2.5$	$\bar{\epsilon}(\sigma)$	$\bar{\epsilon}_r(\sigma)$
$\tau = 0$	0.999/0.999	0.995/0.994	0.986/0.982	0.971/0.968	0.832/0.835	0.002 (0.001)	0.24% (0.16%)
$\tau = 1$	1.000/1.000	0.999/0.998	0.996/0.993	0.989/0.985	0.871/0.881	0.004 (0.003)	0.39% (0.40%)
$\tau = 2$	1.000/1.000	0.999/0.999	0.998/0.995	0.993/0.990	0.886/0.893	0.003 (0.003)	0.28% (0.29%)
$\tau = 3$	1.000/1.000	1.000/1.000	0.999/0.996	0.995/0.992	0.891/0.895	0.002 (0.002)	0.21% (0.18%)
$\tau = 4$	1.000/1.000	1.000/1.000	1.000/0.997	0.997/0.994	0.884/0.892	0.003 (0.003)	0.30% (0.33%)
$\tau = 5$	1.000/1.000	1.000/1.000	1.000/0.998	0.998/0.994	0.871/0.884	0.004 (0.005)	0.42% (0.56%)
$\tau = 6$	1.000/1.000	1.000/1.000	1.000/0.998	0.999/0.995	0.859/0.874	0.004 (0.006)	0.47% (0.66%)
$\tau = 7$	1.000/1.000	1.000/1.000	1.000/0.999	1.000/0.996	0.849/0.867	0.005 (0.007)	0.52% (0.81%)
$\tau = 8$	1.000/1.000	1.000/1.000	1.000/0.999	1.000/0.997	0.846/0.866	0.005 (0.008)	0.55% (0.91%)
$\tau = 9$	1.000/1.000	1.000/1.000	1.000/1.000	1.000/0.998	0.848/0.868	0.004 (0.008)	0.51% (0.93%)
$\bar{\epsilon}(\sigma)$	0.000 (0.000)	0.000 (0.000)	0.002 (0.001)	0.003 (0.001)	0.012 (0.006)		
$\bar{\epsilon}_r(\sigma)$	0.00% (0.00%)	0.02% (0.04%)	0.22% (0.12%)	0.33% (0.06%)	1.37% (0.72%)		

Tabla 7.2
Case study I. Level Precision (LP).

In general, the metrics calculated for the training data fit well with the calculations made for the experimental data. Absolute errors do not present a proportionality relationship with the parameters δ and τ , except for LP metric (Table 7.2), whose error increases as δ increases too. The largest errors in mag-

	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$	$\delta = 2.5$	$\bar{\epsilon}(\sigma)$	$\bar{\epsilon}_r(\sigma)$
$\tau = 0$	0.910/0.906	0.743/0.748	0.555/0.600	0.474/0.491	0.189/0.200	0.016 (0.015)	3.73% (2.96%)
$\tau = 1$	0.959/0.950	0.863/0.863	0.738/0.752	0.662/0.676	0.396/0.366	0.013 (0.010)	2.51% (2.64%)
$\tau = 2$	0.966/0.963	0.890/0.892	0.774/0.803	0.722/0.737	0.500/0.475	0.015 (0.011)	2.27% (1.88%)
$\tau = 3$	0.973/0.971	0.906/0.912	0.804/0.838	0.753/0.778	0.573/0.552	0.018 (0.012)	2.42% (1.65%)
$\tau = 4$	0.974/0.975	0.913/0.927	0.821/0.864	0.778/0.812	0.624/0.618	0.020 (0.016)	2.44% (2.00%)
$\tau = 5$	0.979/0.979	0.920/0.938	0.848/0.884	0.799/0.837	0.666/0.677	0.021 (0.015)	2.52% (1.75%)
$\tau = 6$	0.980/0.981	0.930/0.947	0.861/0.896	0.820/0.857	0.707/0.725	0.022 (0.013)	2.61% (1.59%)
$\tau = 7$	0.984/0.984	0.935/0.954	0.873/0.908	0.832/0.874	0.735/0.768	0.026 (0.015)	3.12% (1.86%)
$\tau = 8$	0.988/0.985	0.941/0.959	0.891/0.920	0.847/0.887	0.766/0.799	0.025 (0.013)	2.90% (1.62%)
$\tau = 9$	0.990/0.987	0.949/0.965	0.900/0.932	0.853/0.901	0.790/0.828	0.027 (0.016)	3.20% (1.96%)
$\bar{\epsilon}(\sigma)$	0.003 (0.002)	0.011 (0.007)	0.033 (0.008)	0.031 (0.012)	0.023 (0.010)		
$\bar{\epsilon}_r(\sigma)$	0.27% (0.01%)	1.29% (0.09%)	4.19% (0.64%)	4.23% (0.83%)	4.52% (2.61%)		

Tabla 7.3
Case study I. Transition Precision (TP).

	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$	$\delta = 2.5$	$\bar{\epsilon}(\sigma)$	$\bar{\epsilon}_r(\sigma)$
$\tau = 0$	0.917/0.927	0.786/0.798	0.668/0.686	0.590/0.587	0.277/0.296	0.012 (0.006)	2.54% (2.28%)
$\tau = 1$	0.967/0.973	0.913/0.921	0.888/0.860	0.825/0.807	0.580/0.541	0.020 (0.012)	2.71% (2.21%)
$\tau = 2$	0.979/0.983	0.926/0.947	0.917/0.902	0.876/0.864	0.691/0.668	0.015 (0.007)	1.80% (0.97%)
$\tau = 3$	0.988/0.986	0.947/0.960	0.937/0.924	0.904/0.897	0.755/0.738	0.010 (0.005)	1.20% (0.69%)
$\tau = 4$	0.989/0.989	0.954/0.968	0.947/0.940	0.916/0.919	0.790/0.784	0.006 (0.005)	0.66% (0.49%)
$\tau = 5$	0.991/0.990	0.964/0.973	0.954/0.949	0.937/0.931	0.836/0.817	0.008 (0.006)	0.89% (0.74%)
$\tau = 6$	0.991/0.991	0.974/0.977	0.960/0.956	0.950/0.943	0.852/0.844	0.004 (0.003)	0.48% (0.33%)
$\tau = 7$	0.992/0.993	0.974/0.981	0.963/0.964	0.955/0.952	0.869/0.863	0.004 (0.002)	0.39% (0.27%)
$\tau = 8$	0.995/0.994	0.978/0.985	0.968/0.970	0.963/0.960	0.884/0.879	0.004 (0.002)	0.38% (0.23%)
$\tau = 9$	0.996/0.995	0.982/0.988	0.975/0.974	0.967/0.967	0.898/0.892	0.003 (0.003)	0.30% (0.28%)
$\bar{\epsilon}(\sigma)$	0.003 (0.003)	0.010 (0.005)	0.009 (0.008)	0.006 (0.005)	0.015 (0.010)		
$\bar{\epsilon}_r(\sigma)$	0.27% (0.01%)	1.07% (0.07%)	1.04% (0.13%)	0.71% (0.12%)	2.24% (1.07%)		

Tabla 7.4
Case study I. Transition Recall (TR).

nitude are presented by transition precision metric (TP, in Table 7.3), reaching 0.033 in the worst case. In addition, the greatest relative error is experienced by traffic saving ratio metric (TSR, Table 7.1), reaching 7.46% in the worst case scenario. As the magnitude of the current value (0.201 in the worst case) is close to 0, the relative error is high. Absolute errors in TSR are also reduced. It seems reasonable to conclude that the training data faithfully represent the experimental data, taking into account the magnitude of errors and that the training data consists of one year of data against eight years of experimentation.

As it was expected, the larger the δ , the larger the traffic saving (TSR, Table 7.1). Transition precision (TP) and recall (TR) metrics can be interpreted as a probability, since they are obtained from the casuistry of the training data. With this perspective, transition precision is the probability that a transition obtained with reduced input data according to δ would have been detected with non-reduced data with a time offset of no more than $\pm\tau$ samples. In turn, transition recall is the probability that a transition that would have been detected with non-reduced data will be detected with reduced data according to δ with a time offset of no more than $\pm\tau$ samples. It is interesting to note the significant increase in precision and recall experienced when comparing an absolutely rigorous approach, without tolerating delays ($\tau = 0$), to a more flexible approach tolerating delays ($\tau > 0$). The most pronounced increase occurred in the change from $\tau = 0$ to $\tau = 1$, which suggests that a large part of the transitions that are not instantly detected are detected indeed with a temporal delay of one sampling period. This aspect confirms the grouping of errors around transitions, an effect that is illustrated in Figure 7.4.

Since it is a demonstrator, the numerical data does not deserve further attention. However, the potential of this proposal lies in the interpretation that can be made of its results. At this point, let us exemplify what the model provides to reduce network traffic. Suppose that the network in which this condition is implemented is overloaded for some reason, and it is urgent to reduce its traffic load by 15%. Also assume that the application tolerates a time delay of three sampling periods ($\tau = 3$). An operator or algorithm that controls the model will have the following information (involved values are marked in bold in the tables):

- Table 7.1 reveals that $\delta = 0.5$ will reduce traffic by 20.1%.
- According to Table 7.3, each transition notified will have a 90.6% chance of being correct.
- Likewise, according to Table 7.4 and under the same conditions, 94.7% of the transitions that would have been detected without reduction will be detected.
- Table 7.2 reveals that 100% of the samples outside transitions $\pm\tau$ sampling periods will be correct. This means that there are no erroneous values that remain more than seven samples over time ($2\tau + 1$).

These interpretations are based on the values obtained from the training data. Assuming that the reduction is carried out with these parameters during the whole period of experimentation, the following would have been achieved:

- Reduced network traffic by 18.6% (1.5% lower than expected).
- 91.2% of the transitions detected were correct (0.6% higher than expected).
- 96% of the transitions that would have been detected without reduction were detected (1.3% higher than expected).
- As it was expected, there were no erroneous values that remain more than seven samples over time.

7.5.2 Case Study II

This case study is based on a network of four sensors, two temperature sensors, and two wind speed sensors. Each pair of temperature and speed sensors is used to calculate wind chill (according to the classic Steadman formula [35]) and to compare them. It is intended to enrich the previous demonstrator with a more complex application example. The parameters of the model are as follows:

1. Input domain: $\Phi = \mathbb{R}$.
2. Condition state function: $s(t_1, v_1, t_2, v_2) = (1.41 - 1.162 \cdot v_1 + 0.98 \cdot t_1 + 0.0124 \cdot v_1^2 + 0.0185 \cdot v_1 \cdot t_1) > (1.41 - 1.162 \cdot v_2 + 0.98 \cdot t_2 + 0.0124 \cdot v_2^2 + 0.0185 \cdot v_2 \cdot t_2)$. This is the classic Steadman formula.
3. Input data: all data has been extracted from the repository of the Italian weather network Arpa Piemonte [7], used in the previous case study. Data from sensors t_1 and v_1 are extracted from the weather station located in Cameri. The data for t_2 and v_2 correspond to the station located in Boves. The training data correspond to the first year of data, the experimental data are the rest. These four signals are composed of 78840 samples; the signal resulting from the condition, sample by sample, presents 14015 transitions; and the true and false balance is 58%/42%, respectively.
4. Reduction function: $r(\varphi, \delta_r) = \text{Integer}\left(\frac{\varphi}{\delta_r}\right) \cdot \delta_r$.

5. Reduction parameters: $\Delta_r = \{0.1, 0.25, 0.5, 1.0\}$.

From the results obtained, the traffic saving ratio metric (TSR) is represented in Table 7.5, similar to the previous study. Level precision (LP), transition precision (TP), and transition recall (TR) metrics are shown in graphs, presented in this order from left to right in Figure 7.6. The graphs show the τ -dependence of the metrics for each $\delta \in \Delta$ (differentiated by color) with training and experimental data (differentiated by line type).

	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$
	0.062/0.069	0.194/0.211	0.332/0.342	0.430/0.440
ϵ	0.007	0.017	0.010	0.010
ϵ_r	11.29%	8.76%	3.01%	2.33%

Tabla 7.5
Case study II. Traffic Saving Ratio (TSR).

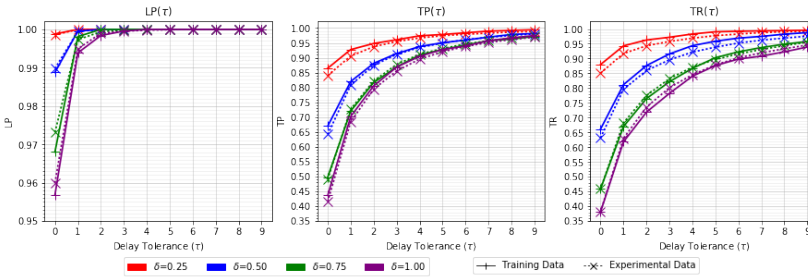


Figura 7.6
Case study II. LP, TP, and TR metrics for training and experimental data.

The results show that the training data fits well with the behavior of the system. The graphics offer a visual perspective that reveals the horizontal asymptotic shape of all curves, in the form $y = 1 - A \cdot e^{-B \cdot x}$. This fact demonstrates from another perspective the grouping of errors around transitions.

7.6 Comparative Test

In order to adjust the reduction models according to the context in real time, the proposed method enables a scale of reduction configurations. Specifically, this work focuses on the context of Boolean expression evaluation. In this framework, the error according to system output is measured by specific metrics defined in Section 7.4.2.

This case study addresses a comparative analysis to understand the potential of contextualization. To this end, two known event-driven sampling techniques have been defined according to the model specifications. Then, for each of these techniques, an adaptive version has also been defined for comparison with the original. Adaptive versions use the reduction parameter scale provided by the model to conveniently optimize data reduction based on context. The following subsections detail the aspects of the experimental scenario.

7.6.1 Increasing Linear Threshold 'K'

In this case study, temperature signals from the repository of the Italian weather network Arpa Piemonte [7] have been loaded to evaluate the expression $T_i[n] > K_i[n]$ where i identifies each station. K_i is a linear expression generated for each station according to Algorithm 1, ranging from the minimum to the maximum value of the signal, to eliminate possible biases in experimentation. Figure 7.7 represents the temperature signal corresponding to the Oropa station and the linear threshold K generated for it.

Algorithm 1 K_i Generation Method

Input: $T_i \in \Phi^N \equiv \mathbb{R}^N$
Output: $K_i \in \Phi^N \equiv \mathbb{R}^N$

- 1: factor $\leftarrow \frac{\max(T_i) - \min(T_i)}{N}$
- 2: $K_i \leftarrow []$
- 3: **for** ($j \leftarrow 0; j < N; j \leftarrow i + 1$) **do**
- 4: $K_i[j] \leftarrow \min(T_i) + i \cdot \text{factor}$
- 5: **end for**
- return** K_i

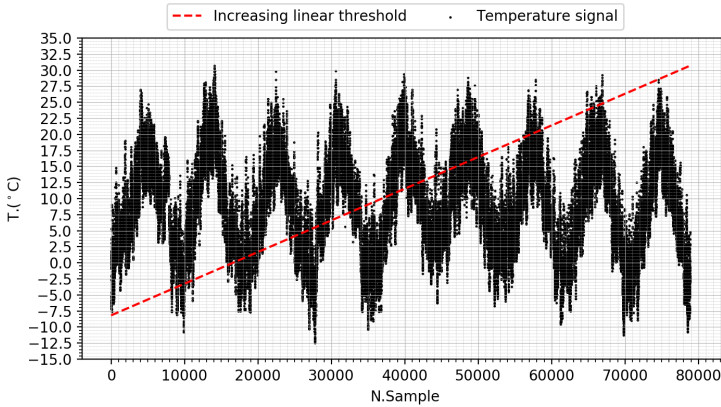


Figura 7.7
Representation of increasing linear threshold 'K', defined for the temperature signal extracted from the Oropa station.

7.6.2 Send-on-Delta and Predictive Sampling

In recent years, non-periodic sampling methods have been developed. These approaches are not based on a regular period of time to report a new sample, but on an event intrinsic to the signal itself. This concept has received different names: event based sampling, Lebesgue sampling, magnitude based sampling, etc.

Send-on-Delta (SoD) [57] is the simplest well known approach. Using the SoD concept, a continuous-time bandlimited signal $x(t)$ is sampled and a new report is sent, when the value of the physical variable being sensed deviates from the value included in the most recent report by an interval of confidence $\pm\Delta$. Many other proposals are extended versions of this one, introducing estimation strategies to enrich this event condition detection. Predictive Sampling (PS) [40] was proposed as a simple example of this case, in which a linear predictor is used to trigger sampling if the value of the measured variable deviates

from the estimated value by $\pm\Delta$.

Most event-based sampling techniques can be easily adapted to the definition of the proposed model if they are understood as filters. That is, it must be assumed that there is an internal process of periodic sampling in the sensors, after which the corresponding technique is applied to decide whether the samples are sent or not. For this scenario, two techniques have been adapted to the model: (a) SoD, and (b) Extended SoD with a linear predictor, which is called Predictive Sampling (PS). The prediction is carried out by linear regression by least squares of the N last samples. This method does not require much computing power, which enables its implementation in small sensors. Algorithms 2 and 3 correspond to the reduction functions for SoD and PS, respectively.

Algorithm 2 SoD Reduction Function

Input: $\varphi, \varphi_{n-1}, \Delta$
Output: φ_n
 1: **if** $|\varphi - \varphi_{n-1}| \geq \Delta$ **then**
 2: $\varphi_n \leftarrow \varphi$
 3: **else**
 4: $\varphi_n \leftarrow \varphi_{n-1}$
 5: **end if**
return φ_n

Algorithm 3 PS Reduction Function

Input: $\varphi, \varphi_{n-1}, \varphi_{n-2}, \dots, \varphi_{n-N}, \Delta$
Output: φ_n
 1: $f \leftarrow \text{LeastSquares}(\varphi_{n-1}, \varphi_{n-2}, \dots, \varphi_{n-N})$
 2: **if** $|\varphi - f[n]| \geq \Delta$ **then**
 3: $\varphi_n \leftarrow \varphi$
 4: **else**
 5: $\varphi_n \leftarrow \varphi_{n-1}$
 6: **end if**
return φ_n

7.6.3 Adaptive Versions of SoD and PS

According to Equations (7.2) and (7.3), giving a reduction function $r \in \mathbb{R}$, Δ_r is a set of reduction parameters, where each $\delta_r \in \Delta_r$ generates Φ_{δ_r} data domain, with less cardinality than the original input domain, Φ . The definition of the Δ_r set enables the implementation of a reduction scale change heuristic that opportunistically adapts to the context. Within this framework, a scale of reduction configurations and a simple method for its variation at runtime has been proposed in this subsection.

Previously, Algorithms 2 and 3 define the reduction function for the classic SoD and PS methods, respectively. These methods are evaluated in the sensors, which are in charge of filtering and sending the new samples of its signal. The reduction parameter change heuristic is a function located in the sink node, which is the node that uses the data for Boolean calculations and therefore provides context. This implies that each change in the reduction parameters implies a network packet sent from the sink node to the nodes. This traffic will be taken into account in the experiment, by adding the number of sink-to-sensor packets to the numerator of the Equation (7.5).

The adaptive versions of SoD (A-SoD) and PS (A-PS) implement the same reduction functions defined in Equations (2) and (3), respectively. Δ_r is an orde-

red list of confidence intervals (different Δ values). The selection of parameter D is made according to Algorithm 4, which applies a distance criterion between the value of the signal and the threshold. In the algorithm, D is an ordered list of distances in magnitude corresponding to each $\delta_r \in \Delta_r$.

Algorithm 4 Reduction Parameter Change Heuristic

Input: $K_{n-1}, \varphi_{n-1}, \delta_{r_{n-1}} \in \Delta_r, \Delta_r \in \mathbb{R}^m, D \in \mathbb{R}^m$
Output: δ_{r_n}

- 1: **for** ($i \leftarrow 0; i < m; i \leftarrow i + 1$) **do**
- 2: **if** $|\varphi_{n-1} - K_{n-1}| < D[i]$ **then**
- 3: **break**
- 4: **end if**
- 5: **end for**
- 6: **if** $\delta_{r_{n-1}} \neq \Delta_r[i]$ **then**
- 7: $\delta_{r_n} \leftarrow \Delta_r[i]$
- 8: **send** δ_{r_n}
- 9: **else**
- 10: $\delta_{r_n} \leftarrow \delta_{r_{n-1}}$
- 11: **end if**

return δ_{r_n}

7.6.4 Experiment Configuration and Results

The case study description leaves some parameters unspecified. In each of the techniques described (SoD, PS, A-SoD, and A-PS), the following remains to be determined:

- SoD_{Δ} : Δ : Confidence interval in magnitude.
- PS_{Δ} : Δ : Confidence interval in magnitude.
 - N : Number of previous samples for linear regression by least squares.
- A-SoD_{Δ_r} : Ordered list of confidence intervals δ_r in magnitude.
 - D : Ordered list of distances between signal and threshold, each distance corresponding to a $\delta_r \in \Delta_r$ parameter.
- A-PS_{Δ_r} : Ordered list of confidence intervals δ_r in magnitude.
 - D : Ordered list of distances between signal and threshold, each distance corresponding to a $\delta_r \in \Delta_r$ parameter.
 - N : Number of previous samples for linear regression by least squares.

Parameters Δ , N , Δ_r , and D are established below, and applied indistinctly to the techniques that require them:

- $\Delta = 0.5$,
- $N = 5$,
- $\Delta_r = 0.1, 0.5, 1.0, 2.5$,
- $D = 2, 5, 8, 11$.

This configuration has been determined by trial and error with one year of data prior to the experiment. Parameter optimization requires in-depth analysis beyond the scope of this work. This case study aims to test the adaptability enabled by the proposed model. It is important to remark that the methods compared here are only examples of the use of the model, which is the main contribution of this work.

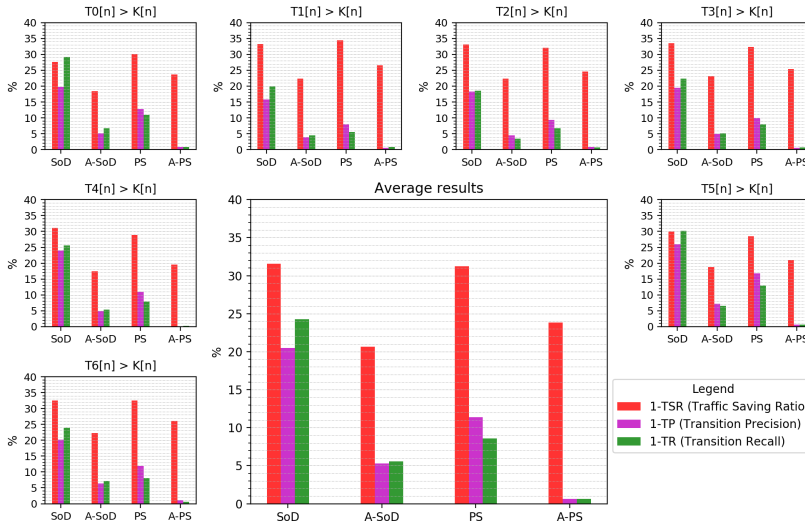


Figura 7.8

Case study III. With event-based sampling techniques SoD, PS, A-SoD, and A-PS previously detailed, the condition $T_i > K_i$ has been evaluated for each temperature signal of each station of the Arpa Piemonte repository [7]. The resulting Boolean signals are analyzed through metrics described in Section 7.4.2 for comparison purposes. The numeric identifier of each station in the repository is as follows: Oropa 0, Cameri 1, Alessandria 2, Vercelli 3, Pallanza 4, Montaldo Scarampi 5, and Bobes 6.

The results of the experiment can be seen in Figure 7.8, where there is a graph for each temperature signal of the repository, and one more with the average values of the metrics applied. Each graph shows the complementary values (for better visualization reasons) of traffic saving ratio TSR, TP, and TR (Equations (7.5), (7.9) and (7.10), respectively). These measurements represent the proportion of sent traffic ($1 - \text{TSR}$), and the proportion of erroneous transitions in terms of false positives ($1 - \text{TP}$) and false negatives ($1 - \text{TR}$). The level accuracy metric (LP, Equation (7.11)) has not been represented because it has a value of 1 in all executions. This means that there are not extended periods of erroneous Boolean samples in either case. As can be observed, the graph of average results faithfully represents all specific cases. For this reason, the following observations are made on the average data of the experiment.

The average values of the experiment (central bar graph) reveal the results below. According to the proportion of traffic sent ($1 - \text{TSR}$), 31.5% of network packets are sent by using SoD, while this amount is reduced to 20.5% by using A-SoD, its adaptive version. The complementary values of this quantity ($1 - (1 - \text{TSR})$) represent the TSR metric according to Equation (7.5), which expresses the proportion of traffic avoided. In this sense, the SoD technique achieves a Traffic Savings Ratio (TSR) of 68.5%, while A-SoD achieves 79.5%. That means an improvement of 16% in this metric. In the case of comparing PS and A-PS in the same way, the TSR values obtained are 69% and 76%, respectively, which means an increase of 10% with the adaptive version.

On the other hand, the values $1 - \text{TP}$ and $1 - \text{TR}$ in the graph represent a proportion of erroneous transitions. The first one ($1 - \text{TP}$) refers to false positives, i.e., transitions that occur but should not. The second ($1 - \text{TR}$) refers to false negatives, i.e., transitions that are expected to occur but do not. The complementary values ($1 - (1 - \text{TP})$ and $1 - (1 - \text{TR})$) are the Transition Precision (TP, Equation (7.9)) and Transition Recall (TR, Equation (7.10)) metrics, respec-

tively, and represent the proportion of successful transitions.

In terms of Transition Precision (TP), the comparison of SoD and A-SoD reveals an increase from 79.5% to 95%, respectively, an improvement of 19.5%. In the case of comparing PS and A-PS, TP increases from 89% to 99% respectively, an improvement of 11%. Finally, the Transition Recall (TR) obtained with SoD and A-SoD increases from 76% to 94.5% respectively, an improvement of 24%. In addition, the TR metric obtained for PS and A-PS increases from 91.5% to 99% respectively, an improvement of 8%.

It is important to emphasize the improvement achieved with the adaptability provided by the model. Especially noteworthy is the improvement in the proportions of erroneous transitions. In this aspect, using SoD, 20% of the detected transitions are erroneous, and 24% of the true transitions are not detected. With the adaptive version A-SoD, these proportions are reduced to 5%. Comparing PS and A-PS, these proportions go from 11% of erroneous detections and 8.5% of no detections, to less than 1% of both with A-PS, achieving an almost error-free system sending only 24% of the traffic.

7.7 Conclusions and Future Work

This paper proposes a data reduction model in sensor networks for a specific application: the evaluation of Boolean conditions. Within this framework, the proposal is based on two fundamental concepts, abstraction and contextualization.

Abstraction is achieved through a generic data filter definition to be implemented in the sensors, which avoids the limitation of working with real numbers. For this purpose, the proposed methodology works with data domains and reduction and equivalence functions to be defined.

Contextualization is addressed by specific metrics, which compare the original Boolean response of a system to the characteristics of the study, with the Boolean response of the system when input signals are reduced. From this comparison, relevant information about the quality of the response is extracted. From this comparison, three percentage measurements are extracted. These values describe different aspects related to the quality of the response, obtaining a more descriptive Boolean error concept.

Taking all this into account, the following hypothesis is put forward: by contextualising the use of data for Boolean calculations, more aggressive reduction techniques can be considered, achieving a more favourable cost-benefit ratio in terms of traffic savings and errors.

In order to test the proposal, two case studies and a comparative test have been carried out. For use cases, a simple quantization filter has been applied for the evaluation of a two-input condition. The input values have been taken, sample by sample, from signals of an extensive repository. Different quantizations were used to characterize the result of the system with a training data window, according to the proposed metrics. Finally, the same analysis was carried out for the rest of the data, comparing the results obtained. The result of both case studies reveals that the training window is valid for characterizing the system. This conclusion is demonstrated by the small variation in the measurements obtained with the rest of the data. These variations are in the range [0.3%, 3.1%] for TP (Table 7.3), [0.3%, 1.5%] for TR (Table 7.4), [0.3%, 1.5%] for TSR (Table 7.1), and [0%, 1.2%] for LP (Table 7.2). On the other hand,

the proposed metrics provide a more accurate interpretation of the variation in system response. Specifically, the interpretation of τ reveals that erroneous samples of system response with reduced inputs tend to concentrate around transitions from the original response. In addition, the smallest τ that makes $LP \simeq 0$ limits the maximum period of consecutive erroneous samples, which is an important feature of the system behavior. Case study 2 shows that the system is capable of working effectively with complex mathematical expressions, which opens the door to future developments.

On the other hand, the comparative study uses the methodology to develop adaptive techniques that maximize the savings-error ratio. In order to test the feasibility of this concept, two known event-based sampling schemes (Send-on-Delta and Predictive Sampling) and their respective adaptive versions have been implemented using the datadomain reduction for thresholdbased event detection (D2R-TED) model. Here, the signals from the repository have been compared sample by sample with an increasing linear threshold that spans the entire signal range. Adaptive versions apply a simple heuristic based on a threshold distance criterion to change the reduction parameters. Despite the simplicity of the adaptability method, the results (Figure 7.8) show that adaptive techniques implemented using a D2R-TED model improve approximately in a range between 10% and 16% the traffic savings obtained by the non-adaptive techniques. Moreover, the Transition Precision and Recall of the A-SoD and A-PS show larger improvements from the standard methods. For instance, for Transition Precision, A-SoD improves SoD approximately in a 19.5%, and A-PS shows an improvement of 11% with PS. In Transition Recall, the improvements are 24% and 8%, for A-SoD and A-PS, respectively. Furthermore, the A-PS is able to reduce the errors almost to 0 by sending approximately 23% of network packets.

As a general conclusion, the model and the proposed metrics provide an efficient framework for the definition of data filters in sensor networks where the sink node calculates Boolean conditions. The abstraction provided by D2R-TED enables scenarios where information packages are modeled as data structures, providing the versatility inherent to computer science. The reduction and equivalence functions (Equations (7.3) and (7.4), respectively) can be redefined to establish convenient criteria in each case, opening the door to future research in areas such as vision, classification, tracking, fuzzy logic, etc. In particular, smart camera deployments can take advantage of the D2R-TED model, following a similar approach to the example shown in Figure 7.3.

REFERENCES

- [1] Christos Antonopoulos, Sofia Maria Dima, and Stavros Koubias: *Event Identification in Wireless Sensor Networks*. In *Components and Services for IoT Platforms*, pages 187–210. Springer International Publishing, Cham, 2017. (Citado en la página 18.)
- [2] Costin Badica, Lars Braubach, and Adrian Paschke: *Rule-based distributed and agent systems*. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 6826 LNCS:3–28, 2011, ISSN 03029743.
- [3] Richard Baraniuk, Emmanuel Candes, Robert Nowak, and Martin Vetterli: *Compressive Sampling [From the Guest Editors]*. IEEE Signal Processing Magazine, 25(2):12–13, mar 2008, ISSN 1053-5888.
- [4] Paolo Barsocchi, Antonello Calabrò, Erina Ferro, Claudio Gennaro, Eda Marchetti, and Claudio Vairo: *Boosting a Low-Cost Smart Home Environment with Usage and Access Control Rules*. Sensors, 18(6):1886, jun 2018, ISSN 1424-8220.
- [5] Jesús Blanco, Andrés García, Javier Morenas, Jesús Blanco, Andrés García, and Javier de las Morenas: *Design and Implementation of a Wireless Sensor and Actuator Network to Support the Intelligent Control of Efficient Energy Usage*. Sensors, 18(6):1892, jun 2018, ISSN 1424-8220.
- [6] Sunyanan Choochotkaew, Hirozumi Yamaguchi, Teruo Higashino, Megumi Shibuya, and Teruyuki Hasegawa: *EdgeCEP: Fully-Distributed Complex Event Processing on IoT Edges*. In *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 121–129. IEEE, jun 2017, ISBN 978-1-5386-3991-7.
- [7] Department of Forecasting Systems, Regional Agency and Environmental Protection of Piedmont, Italy: *Meteorological Data From Arpa Piemonte Weather Stations, Italy*, 2017. (Citado en la página 128.)
- [8] Narendra Kumar Dhar, Nishchal Kumar Verma, and Laxmidhar Behera: *Adaptive Critic-Based Event-Triggered Control for HVAC System*. IEEE Transactions on Industrial Informatics, 14(1):178–188, jan 2018, ISSN 1551-3203.
- [9] Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner: *A Survey about Prediction-Based Data Reduction in Wireless Sensor Networks*. ACM Comput. Surv., 49(3):1–35, 2016.
- [10] Miguel Diaz-Cacho, Emma Delgado, Antonio Barreiro, and Pablo Falcón: *Basic Send-on-Delta Sampling for Signal Tracking-Error Reduction*. Sensors, 17(2):312, feb 2017, ISSN 1424-8220.
- [11] D.L. Donoho: *Compressed sensing*. IEEE Transactions on Information Theory, 52(4):1289–1306, apr 2006, ISSN 0018-9448.
- [12] Norman Dziengel, Martin Seiffert, Marco Ziegert, Stephan Adler, Stefan Pfeiffer, and Jochen Schiller: *Deployment and evaluation of a fully applicable distributed event detection system in Wireless Sensor Networks*. Ad Hoc Networks, 37:160–182, 2016, ISSN 15708705.

- [13] Felipe Espinosa Zapata, José Luis Lázaro Galilea, and Joaquín Olivares Bueno: *Proyecto ALCOR: Contribuciones a la Optimización del Guiado Remoto de Robots en Espacios Inteligentes*. Revista Iberoamericana de Automática e Informática industrial, 15(4):416, sep 2018, ISSN 1697-7920.
- [14] Rajrup Ghosh and Yogesh Simmhan: *Distributed Scheduling of Event Analytics across Edge and Cloud*. ACM Transactions on Cyber-Physical Systems, 2(4):1–28, jul 2018, ISSN 2378962X.
- [15] Yu Hu, Qiang Lu, Yanzhu Hu, Yu Hu, Qiang Lu, and Yanzhu Hu: *Event-Based Communication and Finite-Time Consensus Control of Mobile Sensor Networks for Environmental Monitoring*. Sensors, 18(8):2547, aug 2018, ISSN 1424-8220.
- [16] Mitchell C Kerman, Wei Jiang, Alan F Blumberg, and Samuel E Buttrey: *Event detection challenges, methods, and applications in natural and artificial systems*. Proc. 14th Int. Command Control Res. Technol. Symp., pages 1–19, 2009. (Citado en la página 19.)
- [17] Jagrati Kulshrestha and Manas Kumar Mishra: *Energy balanced data gathering approaches in wireless sensor networks using mixed-hop communication*. Computing, 100(10):1033–1058, oct 2018, ISSN 0010-485X.
- [18] Román Lara, Diego Benítez, Senior Member, Antonio Caamaño, Marco Zennaro, José Luis Rojo-álvarez, and Senior Member: *On Real-Time Performance Evaluation of Volcano-Monitoring Systems With Wireless Sensor Networks*. IEEE Sensors Journal, 15(6):3514–3523, 2015, ISSN 1530-437X.
- [19] Yann Aël Le Borgne, Sylvain Raybaud, and Gianluca Bontempi: *Distributed principal component analysis for wireless sensor networks*. Sensors, 8(8):4821–4850, 2008, ISSN 1424-8220. (Citado en la página 21.)
- [20] Dongho Lee and Kwangsue Chung: *Adaptive duty-cycle based congestion control for home automation networks*. IEEE Trans. Consum. Electron., 56(1):42–47, 2010, ISSN 00983063.
- [21] Zhetao Li, YuXin Liu, Ming Ma, Anfeng Liu, Xiaozhi Zhang, and Gungming Luo: *MSDG: A novel green data gathering scheme for wireless sensor networks*. Computer Networks, 142:223–239, sep 2018, ISSN 1389-1286.
- [22] Wenchang Luo, Boyuan Gu, and Guohui Lin: *Communication scheduling in data gathering networks of heterogeneous sensors with data compression: Algorithms and empirical experiments*. European Journal of Operational Research, 271(2):462–473, dec 2018, ISSN 0377-2217.
- [23] Wenchang Luo, Yao Xu, Boyuan Gu, Weitian Tong, Randy Goebel, and Guohui Lin: *Algorithms for Communication Scheduling in Data Gathering Network with Data Compression*. Algorithmica, 80(11):3158–3176, nov 2018, ISSN 0178-4617.
- [24] Haroon Malik, Ahsan Samad Malik, and Chanchal K. Roy: *A methodology to optimize query in wireless sensor networks using historical data*. J. Ambient Intell. Humaniz. Comput., 2(3):227–238, 2011, ISSN 18685137.
- [25] Miguel Martínez-Rey, Felipe Espinosa, Alfredo Gardel, and Carlos Santos: *On-Board Event-Based State Estimation for Trajectory Approaching and Tracking of a Vehicle*. Sensors, 15(6):14569–14590, jun 2015, ISSN 1424-8220.
- [26] Marek Miskowicz: *Send-on-delta concept: An event-based data reporting strategy*. Sensors, 6(1):49–63, 2006, ISSN 14248220. (Citado en la página 33.)

- [27] Marek Miskowicz: *Event-based sampling strategies in networked control systems*. In *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*, pages 1–10. IEEE, may 2014, ISBN 978-1-4799-3235-1.
- [28] Mohammad Ali Moridi, Mostafa Sharifzadeh, Youhei Kawamura, and Hyong Doo Jang: *Development of wireless sensor networks for underground communication and monitoring systems (the cases of underground mine environments)*. *Tunnelling and Underground Space Technology*, 73:127–138, mar 2018, ISSN 08867798.
- [29] Luca Mottola: *Programming Wireless Sensor Networks : Fundamental Concepts and State of the Art*. *ACM Comput. Surv.*, 5:1–51, 2011, ISSN 03600300.
- [30] A. Nasridinov, S. Y. Ihm, Y. S. Jeong, and Y. H. Park: *Event detection in wireless sensor networks: Survey and challenges*. *Lect. Notes Electr. Eng. (Mobile, Ubiquitous, Intell. Comput. Music.*, 274:585–591, 2014, ISSN 18761119 18761100. (Citado en la página 19.)
- [31] Jeongyeup Paek and Ramesh Govindan: *RCRT: rate-controlled reliable transport for wireless sensor networks*. In *Proc. 5th Int. Conf. Embed. networked Sens. Syst. (SenSys 2007)*, number Section 3, page 305, 2007, ISBN 9781595937636.
- [32] Chen Peng and Fuqiang Li: *A survey on recent advances in event-triggered communication and control*. *Information Sciences*, 457-458:113–125, aug 2018, ISSN 0020-0255.
- [33] Chen Peng, Engang Tian, Jin Zhang, and Dajun Du: *Decentralized event-triggering communication scheme for large-scale systems under network environments*. *Information Sciences*, 380:132–144, feb 2017, ISSN 0020-0255.
- [34] Krishna & Priya and Sebastian Terence: *A Survey on Event Detection and Transmission Protocols in an Event Driven Wireless Sensor Network*. *Int. J. Comput. Appl.*, 58(4):12–18, 2012.
- [35] Robert G. Quayle and Robert G. Steadman: *The Steadman Wind Chill: An Improvement over Present Scales*. *Weather Forecast.*, 13(4):1187–1193, 1998, ISSN 0882-8156.
- [36] M. A. Razzaque, Chris Bleakley, and Simon Dobson: *Compression in wireless sensor networks*. *ACM Trans. Sens. Networks*, 10(1):1–44, 2013, ISSN 15504859. (Citado en la página 20.)
- [37] Syed Afsar Shah, Babar Nazir, and Imran Ali Khan: *Congestion control algorithms in wireless sensor networks: Trends and opportunities*. In *J. King Saud Univ. - Comput. Inf. Sci.*, volume 29, pages 236–245. 2017. (Citado en la página 21.)
- [38] Tongxin Shu, Min Xia, Jiahong Chen, and Clarence de Silva: *An energy efficient adaptive sampling algorithm in a sensor network for automated water quality monitoring*. *Sensors*, 17(11), 2017, ISSN 1424-8220.
- [39] Joris Sijs, Leon Kester, and Benjamin Noack: *A study on event triggering criteria for estimation*. In *17th Int. Conf. Inf. Fusion*, pages 1–8, 2014, ISBN 9788490123553.
- [40] Young Soo Suh: *Send-On-Delta Sensor Data Transmission With A Linear Predictor*. *Sensors*, 2007, ISSN 1424-8220.

- [41] Hongfeng Sun: *Wireless Sensor Traffic Information Collection System Based on Congestion Control Algorithm*. International Journal of Online Engineering (iJOE), 13(11):144, nov 2017, ISSN 1861-2121.
- [42] Annie Uthra Rajan, Antony Jeyasekar, Kasmir Raja S.V., and Anthony J. Lattanze: *Energy-efficient predictive congestion control for wireless sensor networks*. IET Wireless Sensor Systems, 5(3):115–123, jun 2015, ISSN 2043-6386.
- [43] Fusheng Wang, S Liu, Peiya Liu, and Yijian Bai: *Bridging physical and virtual worlds: complex event processing for RFID data streams*. Adv. Database Technol. . . ., pages 588–607, 2006.
- [44] Yong Wang, Dianhong Wang, Fenxiong Chen, and Wu Fang: *Efficient event detection using self-learning threshold for wireless sensor networks*. Wirel. Networks, 21(6):1783–1799, 2015, ISSN 1022-0038.
- [45] G Wittenburg, N Dziengel, S Adler, Z Kasmi, M Ziegert, and J Schiller: *Cooperative event detection in wireless sensor networks*. IEEE Commun. Mag., 50(December):124–131, 2012, ISSN 0163-6804.
- [46] Georg Wittenburg, Norman Dziengel, Christian Wartenburger, and Jochen Schiller: *A system for distributed event detection in wireless sensor networks*. Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks - IPSN 10, page 94, 2010.
- [47] Wenwei Xue, Qiong Luo, and Hejun Wu: *Pattern-based event detection in sensor networks*. Distrib. Parallel Databases, 30(1):27–62, 2012, ISSN 09268782.
- [48] L. Yann-Ael and B. Gianluca: *Round Robin Cycle for Predictions in Wireless Sensor Networks*. In *Int. Conf. Intell. Sensors, Sens. Networks Inf. Process.*, pages 253–258, 2005, ISBN 0-7803-9399-6.
- [49] Yihang Yin, Fengzheng Liu, Xiang Zhou, and Quanzhong Li: *An Efficient Data Compression Model Based on Spatial Clustering and Principal Component Analysis in Wireless Sensor Networks*. Sensors, 15(8):19443–65, 2015, ISSN 1424-8220. (Citado en la página 21.)
- [50] Ikjune Yoon, Hyeok Kim, and Dong Kun Noh: *Adaptive data aggregation and compression to improve energy utilization in solar-powered wireless sensor networks*. Sensors, 17(6), 2017, ISSN 1424-8220.
- [51] Dingwen Yuan, Salil S. Kanhere, and Matthias Hollick: *Instrumenting Wireless Sensor Networks A survey on the metrics that matter*. Pervasive and Mobile Computing, 37:45–62, jun 2017, ISSN 1574-1192.
- [52] Ce Zhang, Ou Li, Guangyi Liu, Mingxuan Li, Ce Zhang, Ou Li, Guangyi Liu, and Mingxuan Li: *A Practical Data-Gathering Algorithm for Lossy Wireless Sensor Networks Employing Distributed Data Storage and Compressive Sensing*. Sensors, 18(10):3221, sep 2018, ISSN 1424-8220.
- [53] Ping Zhang, Jianxin Wang, and Kehua Guo: *Compressive sensing and random walk based data collection in wireless sensor networks*. Computer Communications, 129:43–53, sep 2018, ISSN 0140-3664.
- [54] M Zouboulakis and G Roussos: *Complex Event Detection in Extremely Resource-Constrained Wireless Sensor Networks*. Mob. Networks Appl., 16(2):194–213, 2011, ISSN 1383-469X. (Citado en la página 20.)

ÍNDICE

8.1	Abstract	92
8.2	Introduction and Motivation	92
8.2.1	The Problem of a Proper Defense	93
8.2.2	Contextualization	94
8.3	Background	95
8.3.1	Distributed event detection	96
8.3.2	Sampling schemes	97
8.3.3	Traffic aware strategies in WSNs	97
8.3.4	Discussion about the background	98
8.4	Mathematical Approach and Algorithmic Proposal	98
8.4.1	Notation for pseudocode and equations	98
8.4.2	Mathematical approach to The Problem of a Proper Defense	100
8.4.3	The council solution	102
8.4.4	Algorithmic notation	103
8.4.5	Generalisation: DRACE algorithm	106
8.5	Case Study	107
8.6	Experiments	111
8.6.1	DRACE configuration parameters	111
8.6.2	Randomly generated pseudo-realistic signals	111
8.6.3	Metrics	112
8.6.4	Results and discussion	116
8.7	Conclusions	118
8.8	Future Work	119

Data Communication Optimization for the Evaluation of Multivariate Conditions in Distributed Scenarios

Fernando León-García, Francisco J. Rodríguez-Lozano, Joaquín Olivares and Jose M. Palomares

8.1 Abstract

The current technological landscape is characterized by the massive and efficient interconnection of heterogeneous devices. Sensor networks (SNs) are key elements of this paradigm; they support the local loop, the collection and early manipulation of information. Among the applications of SNs, event detection is a well-explored topic in which strategies such as collaboration, self-organization, and others have been developed in depth. In this topic, the simplest and also most used event concept approach is the threshold-based event, which is usually integrated as part of the local sensor process.

This paper addresses a different perspective by discussing the evaluation of multivariate Boolean conditions with distributed variables. We propose a new algorithm (Data Retaining Algorithm for Condition Evaluation, DRACE) that reduces packet traffic while preserving time accuracy in event calculation on an adaptive approach. To facilitate understanding of DRACE, a case study is presented in the context of a logical simile titled The Problem of a Proper Defense.

The algorithm supports parameters that affects the compromise between accuracy and traffic savings. To analyze its performance, 9000 executions of the algorithm have been performed. 9 configurations tested on a repository of 1000 triads of signals randomly generated. Focusing on the most accurate configuration, 99% of executions are error-free, and the number of packets is reduced by 40% on average, being between 30 and 50% in 68% of cases.

8.2 Introduction and Motivation

Nowadays, the technological landscape ranges from large servers in the cloud to small processors in objects across our environments. Classic network technologies become obsolete to face the challenges posed by the massive distribution of hosts and their high heterogeneity [12]. The research topic *internet of things* (IoT) represents the most global vision of this reality. However, the scientific community has adopted other research topics that refer to more specific approaches, such as pieces of the same puzzle (e.g. [wireless] sensor networks (WSN), *edge computing* or *fog computing*) [32].

Adopting IoT terminology, at the opposite edge of the cloud there are things characterized by high heterogeneity and severely constrained resources[29]. Regardless of the goal of the deployment, from the vast variety of scientific proposals related to this issue, two trends in technological development are revealed: I) the optimization of resources of both the devices and the network; and II) the approximation (as far as possible) of the process to the sources of information, as opposed to the methodological tradition of processing everything in the cloud [3].

As part of the ecosystem of things, the wireless sensor networks (WSN) provide an inexpensive way to capture information from the environment [12]. The two most common uses of WSNs are: 1) to collect data from the environment and route it to one or more sink nodes [7, 13, 15]; and 2) to detect and report events [9, 24, 31]. Due to the scarcity of resources available in the sensors, both applications require optimization, which is the first trend noted above. The detection of events, moreover, involves bringing processing closer to the sensors.

WSN event detection is a broad topic that has been approached from many perspectives. The scientific proposals can be classified according to many considerations. An important consideration is the concept of event itself, which in its broadest sense is a relevant occurrence circumscribed in time and place. However, the detection process requires a technical definition in practice that allows us to classify event detection systems according to criteria such as architecture or method [5, 14, 31, 35, 48].

Threshold detection event is one of the most simple and common methods [48]. Threshold values are suitable for a lot of applications, for example, fire detection, detection of flooding, or generally other applications in which a sensor can detect a critical boundary of the measured value [14]. The current trend in event detection focuses on the use of increasingly sophisticated techniques such as artificial intelligence, data mining, or fuzzy logic [46]. This results in a lack of interest in threshold-based detection techniques, which are widely used in industry, distributed control networks, alarm systems, or in the early stages of more complex computational methods, such as rule-based systems or complex event processing [4].

The general conception of threshold-based event detection refers to local evaluation architectures, in which sensors evaluate the condition with their own signals and notify when it is met [14]. However, the use of thresholds to define events covers more complex scenarios for which it is worthwhile to propose strategies to improve the process. The use of non-static thresholds or the satisfaction of conditions that depend on multiple network distributed variables are examples of this potential complexity.

The motivation behind this paper is to contribute to the development of this topic, introducing an efficient strategy to deploy networks with a double objective: to collect the data generated by the sensors, and to detect and notify in time threshold-based events expressed as conditions with multiple distributed variables. In order to easily illustrate the intrinsic aspects of this proposal and to favour the understanding of the motivation of this work, a warlike simile is introduced in the following sub-section that incorporates all the relevant aspects to be addressed from another logical context.

8.2.1 *The Problem of a Proper Defense*

A warlord controls a fortress that is surrounded by a moat with 3 drawbridges. His spies reveal to him that his enemies are settling in 3 camps, and that they are trying to recruit soldiers to attack all the bridges simultaneously, one for each army formed. The information is so accurate that it reveals which gate will be attacked by each army at the time of the attack, which will start the next morning to get 10,000 soldiers for the cause counting all camps. However, the reputation of the military is so well known that gathering such a number of soldiers is not being easy, because in addition to new recruits there are also deserters in all enemy armies, making it impossible to accurately estimate when the battle will take place.

To prepare the defense, the warlord has 3000 soldiers, a group of engineers, and 150 defensive devices distributed among the gates, including traps, ballistas, oil boilers, etc. In his experience, one defensive machine equals 50 soldiers. Assuming that each soldier brings a force of 1, the warlord has a defensive force of 10500 against an offensive force of 10000, so he thinks that a good organization will give him victory. Thus, the experienced military decides to proceed as follows. The soldiers will be sent to the different gates in groups

according to the force of the expected attack. For this part of the plan, the military requires to know the exact proportion of the enemy armies the same day of the attack, since the soldiers can be mobilized quickly. The engineers, however, will be in charge of redistributing the mechanical traps, deploying a defensive force proportional to the expected offensive one. For this task, on the other hand, the warlord needs to be informed of the state of the enemy armies, because engineers cannot install or uninstall more than 1 defensive devices in each gate per day, so their work cannot be done at the last moment.

To solve the problem of information, a scout is sent to each enemy camp with a command and a carrier pigeon. The command states: "Hide at the edge of the camp and count the number of soldiers daily. If this is different from what I know, send me the pigeon with the data by nightfall. The pigeon will be returned with or without new orders."

As planned, the military was rigorously informed of the composition of the enemy armies, and preparations were initiated at each gate in just measure. However, after the first 10 days one of his explorers was captured. Apparently, bored due to the delay of the battle, the soldiers of the enemy camp noticed the regular flight of the bird and combed through the nearby forest. It was necessary to convene a meeting of councillors to decide on another strategy that would allow the warlord to be properly informed and to protect the explorers as far as possible.

Once the advisors had met and the problem was exposed, even without knowing how to proceed, a conclusion was reached unanimously: to achieve this, the warlord could not be so well informed. As one of them said, "you can't have the cake and eat it".

8.2.2 Contextualization

Translated into an easily understandable logical context, the Problem of a Proper Defense raises the need to consider the "price" of information. By replacing pigeons by network packets, explorers by sensors, and the castle by the sink node, the proposed scenario corresponds to a sensor network with two simultaneous purposes: the detection of the threshold-based event expressed by inequality $v_1(t) + v_2(t) + v_3(t) \geq 10000$ (for moving the soldiers), and the continuous monitoring of involved variables (for preparing the defensive elements).

Sensor networks are often very limited in energy, so traffic avoidance is a priority. However, if the network is designed to detect an important condition, such as an alarm, a conflict arises between the accuracy of the detection and the traffic required. Hence the "price" of information.

A common solution is to evaluate the condition locally and send a notification when the event takes place. However, if some tracking of system status is not available, an unexpected alarm notification may be useless. This situation happens when the response to the event requires some preparation. For example, a network of fire detection sensors with a fire suppression system that strategically mobilizes water reservoirs. In this case, the system requires a certain resource to be applied proportionally to the imminence of the event.

The cost of maintaining precision in anticipation of an event is not affordable in terms of energy or network traffic. For this reason, the motivation of this work is to provide a logical vision and propose a simple algorithmic solution.

With this solution, the sensors participate in an adaptive tracking process, and the accuracy is proportional to the event occurrence estimation. Thus, if the event is far from being fulfilled, the sensor network will remain in a latent mode, in which consumption and traffic are not a problem.

This work involves a new threshold-based event approach, proposing the scenario of multivariate conditions. We present an algorithm that significantly reduces network traffic, which is widely evaluated by simulation and random signals. The proposal involves the concept of *latent sensor network*, in which the sensors remain conveniently silent as long as the condition in question remains far from being met. This concept may be important in wide coverage applications for alarm purposes, such as fire detection.

The rest of the document is organized as follows. Section 8.3 (Background) deals with the theoretical framework of the work, it describes the current panorama of event detection and traffic reduction in WSN, specifically delving into threshold-based events; Section 8.4 (Mathematical Approach and Algorithmic Proposal) details the mathematical model behind the Problem of a Proper Defense described above, also details the proposed algorithm (DRACE) to address the problem. Section 8.5 (Case Study) uses DRACE to solve the proposed problem, describing in detail the algorithm operation. Section 8.6 (Experiments) addresses an experimental methodology for analyzing the DRACE performance, describing the configurations, data, and metrics. Finally the results are discussed. Section 8.7 (Conclusions) notes the contributions of the work and concludes on the basis of the results. Finally, Section 8.8 outlines the direction of future efforts to continue this work.

8.3 Background

In (W)SNs, event detection is perhaps the most widespread application, and saving traffic is one of the most common optimization goals. The publications on both topics are very varied, and previous considerations are required to specify the scope of the contributions.

In its broadest definition, an event is something that happens in a space and time. This simple, general and ambiguous definition raises more questions than answers: *“is an event model available or is it an anomaly with respect to a normal regime?”*; *“is it deterministic?”*; *“does the event occur punctually in a space, or is it a phenomenon that spreads?”*; *“is immediate detection required or is it enough to detect it in post-processing?”*. The answers to these and other questions give rise to a wide range of approaches. Among other things, they determine aspects such as collaboration between nodes, redundancy in detection, convenience of clustering, or the suitability of applying machine learning techniques.

In terms of traffic savings, the variety of contributions is related to the part of the system architecture being optimized. Considering a general architecture, from sensing to processing there are numerous processes to be considered. Duty cycle scheme, sampling, in-node processing, queueing/packing/sending policies, routing, clustering, priorities, remote processing, etc. As in event detection, the literature caseload is vast.

This work aims to contribute to both fields by means of a data queuing strategy that optimizes traffic when the event to be detected can be modeled as a multivariate Boolean condition. The proposed scenario fits into different research topics, such as data gathering schemes, event detection, adaptive sampling, or traffic saving among others. And its application is interesting in dif-

ferent areas, like industrial control, domotics, surveillance, or alarm systems.

Since a comprehensive review of the state of the art is not the aim of this work, the following subsections narrow the scope and describe important contributions that have provided inspiration and background for this paper.

8.3.1 *Distributed event detection*

Although event detection is a concept that transcends the underlying technology, sensor networks have been the main technological framework in the development of the topic.

Event detection in sensor networks is a widely explored and developed research topic. Relevant aspects and techniques to detect events are usually classified according to the criteria described in [48], summarized in [65]. Situational dependence, criticality of application, numerous and diverse data sources, and network topology are key aspects to be considered. Regarding the methods, they are usually classified to: statistical, probabilistic, or artificial intelligence/-machine learning based methods.

According to the conceptual framework exposed above, this work is a traffic-aware model-based method. In this work, a mathematical expression, which provides a Boolean result, has to be evaluated accurately in terms of time triggering. Since the result of the proposed model is Boolean and defined by mathematical expressions, proposals related to threshold-based event detection are considered as background to this work.

Threshold-based approaches are classified within the statistical model-based methods. Essentially, it consists of reporting when a parameter exceeds a given threshold. This model is considered to be a technique with very low computational complexity. Therefore, it is the most widely used method for simple event detection [9, 48].

There are many examples of the use of threshold-based events for the detection of alarm conditions in specific situations, such as wildfires [5, 22], volcanic activity [34], or perimeter invasion [18].

The idea of comparing a value with a threshold is a well-established mechanism. Most related publications normally use threshold-based events, processed at the sensor level. After that, the measured magnitude is compared with a static or dynamic threshold. Finally, the result is sent to a network that exploits other WSNs features, such as redundancy, dynamism, or collaboration. For example, [49] uses threshold-based events along with spatial correlation and redundancy to provide fault-tolerant distributed detection of events. Another example is [12], which proposes a dynamic double threshold technique for detecting abnormal events by aggregation.

Both previous proposals are typical examples in which the threshold is used to detect significant variations on a single variable, not dealing with any composition of multiple variables. This latter fact rises a question about the complexity of threshold-based event models: “*What happens if the threshold is applied to a multivariate function?*” The issues arising from this question have not been addressed from the point of view of sensor constraints and are the main focus of this work.

8.3.2 Sampling schemes

The sampling and notification of distributed variables has been approached from the automatic control. Magnitude-based sampling [56], Lebesgue sampling [8], or Send-on-Delta [57] are different names for the same basic principle: successive samples of a signal are not triggered by time criteria, but by signal variation. There is an extensive literature on the subject, with proposals that integrate predictive methods in the sampling trigger criterion, trying to maximize sample reduction or adapting the process to the intended objectives [58, 89].

These sampling strategies provide an interesting approach to traffic aware methods by reducing the number of data needed for monitoring tasks. The potential of such methods when contextualising the use of data in processes with Boolean results was demonstrated in [21], in which a precision, recall, and traffic savings analysis is presented when evaluating conditional expressions with signals sampled by magnitude criteria. The study also proposes metrics to measure the effect of delays in the detection of transitions in the resulting signal, as a consequence of the magnitude-based sampling scheme. Here, the proposal is based on the conclusions of that study, and proposes an adaptive algorithm that takes advantage of the potential traffic savings by minimizing the introduced error.

8.3.3 Traffic aware strategies in WSNs

One of the challenges of the WSNs is the avoidance of network collapse. WSNs typically deploy many nodes that communicate using low-bandwidth wireless technology. This combination entails a problematic scenario due to potential congestion. For this reason, contributions related to traffic avoidance techniques are of great interest. There are two key concepts that appear frequently in the related proposals: compression and prediction.

Compression techniques are applied to achieve a more efficient use of resources, especially in cases of scarcity. [30] delves into the different approaches to compression in WSNs, classifying them into: a) sampling compression, when the number of sensor sample acquisitions is reduced; b) data compression, with which the number of bits in the data stream is reduced; and c) communications compression, which aims at reducing the number of accesses to the network.

On the other hand, prediction is a powerful tool that is often related to compression, because one can avoid reading/processing/sending what can be predicted with some certainty. [8] approaches this issue from two perspectives: 1) where the prediction procedure is carried out (essentially, sink node, sensor node, or both); and 2) which prediction technique is used (statistic, probabilistic, machine learning, etc.).

According to the previously described concepts, the purpose of this work can be classified as belonging to the communication compression topic. Furthermore, the proposed method is related to forecasting techniques on time signals at a sink node. Some representative examples in this scope are [10, 23, 37]. Nevertheless, all these works deal with reducing network traffic by filtering signal data in the sensor domain (usually, integer or real numbers). None of these proposals take into account the final use of the data for event detection. Therefore, these works reduce the transmissions to the sink with a criterion exclusively focused on minimizing the error in the magnitude of the involved

data. However, the most relevant fact for an event detection system is not the amount of error in the magnitude, but the final Boolean response. Thus, the compression may be more aggressive when the signal is far away from the threshold, and hence, traffic savings would be larger. Moreover, as the proposals are focused on minimizing one data stream, they are not suitable for detecting events based on multiple variables.

8.3.4 Discussion about the background

In the previous subsections, the most relevant scientific background has been introduced. The classification of the techniques in several different scopes has been included. All these methods have provided an outline of the scientific field related to this work.

The definition of “event” according to threshold-based detection is stable in all the revised articles. It only applies the Boolean evaluation on only one variable, either in sensor nodes or in head-cluster ones. There has not been found any reference in which the Boolean evaluation based on thresholding had been applied to functions composed of several different variables. This is a new challenge that this work aims to address.

There are many mechanisms to reduce data sent through the network. However, that reduction of data is achieved without any knowledge about the final use of that data. Therefore, the data sent by the sensors is *decontextualized*. In this work, the final use of the data is known in advance. Therefore, a completely different approach may be applied in this work.

The main background is the D2R-TED model [21]. This model provides the mathematical foundations to allow the data management of multivariate functions with threshold-based event detection. Results shown in that previous work are promising. However, in that work, no algorithm for dynamic adjustment of the threshold was proposed.

8.4 Mathematical Approach and Algorithmic Proposal

This section deals with the algorithmic proposal of this work. Mathematical expressions and algorithms requiring arrays and matrixes are used. For a better understanding, the notation is specified here:

8.4.1 Notation for pseudocode and equations

Array declaration:

- $x \in D^n / n \in \mathbb{N}$ is an array of n elements defined in $D + \{\emptyset\}$, so that $x[i] \in D + \{\emptyset\}$ is the i th element, $i \in \{1, 2, \dots, n\}$.
- $x = \{c\} \times n, c \in D, n \in \mathbb{N}$ is an array of n values equal to c , so $x \in D^n$.
- $x = \{f(e), \forall e \in A\}, A \in D_1^n, f : D_1 \rightarrow D_2, n \in \mathbb{N}$ generates an array of n elements defined in D_2 by mapping A elements with f function. If the generator array contains null values, these are transmitted to the generated array. So $x \in D_2 + \{\emptyset\}^n$.
- $x = \{e \in A / c(e)\}, A \in D^n, f : D \rightarrow \{\text{true}, \text{false}\}$ generates an array with those elements of A for which f is true. So $x \in D^m, m \leq n$.
- All elements of an array whose value has not been explicitly declared are initialized at null value. In other words, $x \in D^n \equiv \{\emptyset\} \times n$, with the

difference that the first declaration specifies the mathematical domain of the elements.

Array operation:

- If $x \in D^n$, $x[-1]$ is the last non-null element in the array, or null if there is none.
- If $x \in D^n$, $\#x$ is the number of non-null value elements contained in x , so $\#x \in \mathbb{N} + \{0\} \leq n$.
- If $x \in D^n$, $x \leftarrow v$ sets the value v at the position of the first null element contained in x . If there are no null values, all values are shifted to the left, discarding the first and leaving the last void, where v is set.
- If $x \in D^n$, $\text{IndexOf}(x, v)$ is the position of the first element of x whose value is v , resulting in \emptyset if none exist. So $\text{IndexOf}(x, v) \in \{1, 2, \dots, n\} + \{\emptyset\}$.
- If $x \in \mathbb{R}^n$, $\text{Max}(x)$ and $\text{Min}(x)$ return the highest and lowest value contained in x , respectively. If the array is empty, both functions will return null.

Matrix declaration:

- $x \in M_{n \times m}(D)/n, m \in \mathbb{N}$ is a matrix of n rows and m columns of elements defined in $D + \{\emptyset\}$, so that:
 - $x[i][j] \in D + \{\emptyset\}$ is the element at the row $i \in \{1, 2, \dots, n\}$ and column $j \in \{1, 2, \dots, m\}$.
 - $x[i][*]$ or simply $x[i]$ is the array of m elements (one per column) corresponding to row $i \in \{1, 2, \dots, n\}$, so $x[i] \in \{D + \{\emptyset\}\}^m$.
 - $x[*][j]$ is the array of n elements (one per row) corresponding to column $j \in \{1, 2, \dots, m\}$, so $x[*][j] \in \{D + \{\emptyset\}\}^n$.
- $x \in M_{n \times *}(D)/n \in \mathbb{N}$ is a matrix of n rows and undefined number of columns of elements defined in $D + \{\emptyset\}$.

Indexing precedence in matrixes:

- If $x \in M_{n \times m}(D^h)/n, m, h \in \mathbb{N}$ is a matrix of n rows and m columns of arrays of h elements defined in D . So that $x[r][c][i] \in D + \{\emptyset\}$ is the i th element of the array in row $r \in \{1, 2, \dots, n\}$ and column $c \in \{1, 2, \dots, m\}$ of the matrix, where $i \in 1, 2, \dots, h$.

Open-ended arrays and matrixes:

In order to specify algorithms that operate undefined length time series, this nomenclature is proposed to declare open sized structures.

- $x \in D^*$ is an unspecified size empty array of elements defined in D . The length of the array is increased by inserting elements at the end with the \leftarrow operator. In this case the array elements never shift to the left, they just grow.
- $x \in M_{n \times *}(D)/n \in \mathbb{N}$ is a matrix of n rows and undefined columns of elements defined in $D + \{\emptyset\}$. initially the number of columns is 0, and the way to increase it is to add elements to the vectors row with \leftarrow operator. The number of columns in the matrix is determined by the number of elements in the longest row vector. And they increase in size using the \leftarrow operator, setting the rest of the elements of the new column to \emptyset . For example:
 1. $x \in M_{2 \times *}(D)$
 2. $\#x[1]$ is 0
 3. $x[1] \leftarrow 1$
 4. $x[*][1]$ is $\{1, \emptyset\}$
- $x \in M_{* \times m}(D)/m \in \mathbb{N}$ is a matrix with undefined rows and m columns of elements defined in $D + \{\emptyset\}$. In this case the rows are created by adding elements to the column vectors.
- Matrixes must have at least one dimension defined.

8.4.2 Mathematical approach to The Problem of a Proper Defense

From the mathematical point of view, the Problem of a Proper Defense can be modelled from a set of parameters and variables, such as those proposed below.

$TDS \in \mathbb{N}$ Total Defense Soldiers.

$TDD \in \mathbb{N}$ Total Defense Devices.

$E \in \mathbb{N}$ number of explorers (one for each enemy camp).

$X \in \mathbb{N}$ Ith day since the implementation of the strategy in which the battle occurs. Therefore:

$$X = \text{Min} (\{i \in \mathbb{N} / \sum AS[*][i] \geq 10^4\}).$$

$\lambda \in \mathbb{R}$ Defensive factor of defense devices.

$\varepsilon \in \mathbb{N}$ Work pace of engineers.

$AS \in M_{E \times *}$ Number of Attack Soldiers per camp and day. So that $AS[i][d]$ is the number of soldiers in camp i at day d .

$AS' \in M_{E \times *}$ Last known value of number of enemies in camp i on day d . So that $AS'[i][d]$ is information that is known day d about the number of soldiers in camp i .

$DD \in M_{E \times *}(N)$ Number of defensive devices per gate and day. So that $DD[i][d]$ is the number of defensive devices in gate i at day d .

The temporal granularity of the story is daily, so the independent variable d represents the day number.

According to military numbers, there are 3000 soldiers and 150 defensive devices, so $TDS = 3000$ and $TDD = 150$. The strength of each defense device in battle is 50 times the defensive strength of a soldier, then $\lambda = 50$. It is also considered that engineers cannot manipulate more than 1 defensive device per day, then $\varepsilon = 1 \frac{\text{units}}{\text{day}}$. These two parameters model the requirement to track monitored time series, as defensive devices are key to winning the battle, for which they must be in place on the day of the fight. In other words, increasing λ or ε means increasing the margin of error in the placement of war machinery.

Assuming battle takes place on day X , the defense of the castle requires the victory of the defensive forces at all gates, (8.1) expresses this condition considering the Boolean values *true* = 1 and *false* = 0. Fig. 8.1 is a geographical representation of the military forces according to this model.

$$V(X) = \begin{cases} \text{true, if } \sum_{i=1}^E (DS[i][X] + \lambda \cdot DD[i][X]) \geq AS[i][X] = E \\ \text{false, otherwise} \end{cases} \quad (8.1)$$

Although the above expressions define the state of the conflict on day X , the evolution of the defensive forces depends on the information received by carrier pigeons (AS'), which does not necessarily correspond to the actual data (AS). With regard to the movement of soldiers, the warlord intends to distribute them proportionally to the offensive forces on the day of the attack (day X). For this action to take place it is necessary that $\sum (AS'[*][X]) \geq 10^4$. Equation (8.2) expresses this fact.

$$DS[i][d] = \begin{cases} \frac{TDS \cdot AS'[i][d]}{\sum (AS'[*][d])} \text{ if } \sum (AS'[*][d]) \geq 10^4 \\ 0 \text{ otherwise} \end{cases} \quad (8.2)$$

$\forall i \in 1, 2, \dots, E$



Figura 8.1

Artistic representation of the Problem of a Proper Defense, designed using Inkarnate® web tool (<http://www.inkarnate.com>).

With respect to defensive devices, their relocation is decided on the basis of the daily information received. Algorithm 5 models this process assuming that defensive devices were not operational before implementing this strategy.

Algorithm 5 Update_DD(d)

Require: ε , DD, TDD, AS'

```

1: works =  $\varepsilon$ 
2: done =  $\{0\} \times E$ 
3: if d = 1 then
4:   to_do =  $\left\{ \frac{AS'[i][d]}{\sum (AS'[*][d])} \cdot TDD \right\} \forall i \in \{1, 2, \dots, E\}$ 
5:   free_devices = TDD
6: else
7:   to_do =  $\left\{ \frac{AS'[i][d]}{\sum (AS'[*][d])} \cdot TDD - DD[i][d-1] \right\} \forall i \in \{1, 2, \dots, E\}$ 
8:   free_devices = TDD -  $\sum DD[*][d-1]$ 
9: end if
10: while works > 0 and Max( $\{|x|, \forall x \in \text{to\_do}\}$ ) > 0 do
11:   if free_devices > 0 then
12:     i = IndexOf(to_do, Max( $\{|x|, \forall x \in \text{works}\}$ ))
13:   else
14:     i = IndexOf(to_do, Min(to_do))
15:   end if
16:   x =  $\frac{\text{to\_do}[i]}{|\text{to\_do}[i]|}$ 
17:   done[i] = done[i] + x
18:   to_do[i] = to_do[i] - x
19:   free_devices = free_devices - x
20:   works = works - 1
21: end while
22: for i  $\in \{1, 2, \dots, E\}$  do
23:   if d = 1 then

```

```
24:   DD[i] ← done[i]
25:   else
26:     DD[i] ← DD[i][d - 1] + done[i]
27:   end if
28: end for
```

So much for the model that describes the actions taken in the castle. The problem of the cost of information must be addressed on the side of the explorers. The number of pigeon flights has to be reduced at least as long as the battle is not imminent. The council solution is described below.

8.4.3 *The council solution*

After a long discussion, the councillors agreed to proceed as follows:

On the one hand, explorers will continue to record the count of enemy soldiers on a daily basis, but it is neither appropriate nor necessary to send it every day. Each explorer will retain its data until it sufficiently differs from the last value of the last scroll sent. In this way, the information will be sent when: a) a variation greater than a threshold is recorded, or b) no more values fit on the scroll.

On the other hand, the key question of this strategy is which threshold would be appropriate in each case and how the explorers would know. It was concluded that it should be the military command in the castle who established the thresholds with the latest known information, in addition to preparing the defenses. After all, it is there where the conflict is seen as a whole.

In order to tackle this task an information channel from the castle to each explorer was required. For this purpose, pigeon return flights would be used. In this way, explorers who send information can receive a new threshold as a response if appropriate.

Regarding the calculation of thresholds, the following procedure was outlined:

1. To facilitate the calculation of thresholds, they were limited to a series of predefined values, allowing fine-tuning the surveillance process of enemy camps according to a scale of granularity in the accounts. The proposed scale was 1, 25, 100, 200, and 300. All explorers would have threshold 1 initially.
2. For each explorer, a data record for each scale value would be prepared. Each scroll received would be copied value by value to all the records following this rule: for the record corresponding to threshold X , only those values that differ a minimum of X from the last recorded value would be noted.
3. Every day, after all information has been received and recorded in accordance with this rule, defense devices preparation would be coordinated with the most recent information available. Also, if the battle condition were met, soldiers would be proportionally placed at each gate in preparation for the imminent conflict; if not, all explorers who have sent pigeons would be candidates for changing their threshold value.
4. The calculation of thresholds for candidates is an iterative process in which the battle condition is evaluated by changing known information for expected information, with the intention of finding a safe threshold

setting so that the castle would not be surprised in battle. The calculation would have the following steps:

1. For each record of each explorer, the maximum expected variation in the next value would be forecast using the last 10 values, if available.
2. For each candidate, each maximum expected variation per threshold added to the last known value would suppose a possible configuration. For non-candidate explorers there would only be one configuration, corresponding to their current thresholds.
3. Starting from the combination of configurations with higher thresholds, the battle condition would be re-evaluated. If the condition were met, one of the configurations would be replaced by the one of the next lower threshold. This process would be repeated until:
 - a) a combination is found that does not meet the condition, or b) the thresholds can no longer be lowered.

Two aspects of the strategy were discussed at the end, as it was felt that there were many options on the table. At the one hand, the method of forecasting the maximum variation to be expected. At the other hand, the selection of the candidate to reduce the threshold in the iterative process of searching for the best combination of thresholds.

Finally, a simple method was decided by consensus:

1. The maximum expected variation is K times the standard deviation (σ) of the queue values, and to be on the safe side he proposed $K = 3$.
2. The best candidate to be reduced will be the one with the greatest variation present in the forecast.

Algorithm 6 Explorer

Require: DPM, $\Delta[0]$

Initialisation :

1: $\delta = \Delta[0]$

2: scroll $\in \mathbb{N}^{\text{DPM}}$

3: last $\in \mathbb{N} = \emptyset$

LOOP Process (daily)

4: scroll \leftarrow Count_Soldiers()

5: **if** last = \emptyset **or** #scroll = DPM **or** |scroll[-1] - last| $\geq \delta$ **then**

6: last = scroll[-1]

Reporting interval

7: Send_Pigeon(scroll)

8: **while** #scroll > 0 **do**

9: scroll $\leftarrow \emptyset$

10: **end while**

Out of reporting interval

11: Wait_For_Pigeon(& δ')

12: **if** ($\delta' \neq \emptyset$) **then**

13: $\delta = \delta'$

14: **end if**

15: **end if**

8.4.4 Algorithmic notation

Let us define the variables that complete the model according with the strategy of the council.

TDS	Total Defense Soldiers.
TDD	Total Defense Devices.
E	Number of Explorers.
X	Day of the battle.
λ	Defensive factor of defense devices.
e	Work pace of engineers.
AS	Number of Attack Soldiers per camp and day.
AS'	Last known value of numbers of enemies per camp and day.
DD	Number of defensive devices per gate and day.
DPM	Maximum number of data per message.
Δ	Set of possible threshold.
Q	Number of values considered for forecast (queues lenght).
K	Multiplier factor of the standard deviation.

Tabla 8.1
Summary of notations

DPM $\in \mathbb{N}$ Maximum number of data per message (scroll).

$\Delta = \{\delta_0, \delta_1, \dots, \delta_{n-1}\}, n \in \mathbb{N}$ Set of possible thresholds.

Q Number of values considered to forecast the maximum variation for each threshold record.

K Multiplier factor of the standard deviation when predicting the next maximum value.

For better understanding, all notations are summarized in Table 8.1.

Algorithm 6 (Explorers) consists of a read and send loop on a daily basis. The reading and annotation of the collected value is always done (L: 4). Sending only occurs if required (L: 5), depending on the δ value and the free slots in the scroll (packet). After each sending, there is a wait for a possible message with a new value of parameter δ (L: 11).

Algorithm 7 (Castle) consists of a daily iterative process with three well-differentiated parts. The first part is called the *reporting interval* (L: 7–20). In this part, the packets coming from the sending explorers are processed, adding the containing data in queues, corresponding to the possible values of δ (Δ). These queues are required by the forecasting method. The second part (L: 21–28) uses the most up-to-date information available to organize the defenses, finalizing the algorithm if the battle condition is met (in which case the soldiers would be mobilized to the gates). The third part (L: 29–63) is carried out if the battle condition is not met. This part is an iterative process in which the best δ value is determined for each explorer. In this process, the aim is to maximize the δ -values without incurring the risk of suffering the attack without having foreseen it. For this purpose, the algorithm uses the queues filled in the first part to assess the risk of meeting the battle condition with each δ and explorer.

Algorithm 7 Castle

Require: E, Q, Δ , K

Initialisation :

1: $i_delta = \{1\} \times E$

2: $AS' \in M_{E \times *}(N)$

```

3: queue  $\in M_{E \times \#\Delta}(N^Q)$ 
4: fc_dev  $\in M_{E \times \#\Delta}(R)$ 
5: day = 1
   LOOP Process (daily)
6: candidates  $\in N^E$ 
   Reporting interval
7: while Received scrolle from explorer e do
8:   candidates  $\leftarrow e$ 
9:   for  $x \in \text{scroll}_e$  do
10:    for  $i_\delta \in \{1, 2, \dots, \#\Delta\}$  do
11:       $\delta = \Delta[i_\delta]$ 
12:      if #queue[e][ $i_\delta$ ] = 0 or |queue[e][ $i_\delta$ ][-1] -  $x$ |  $\geq \delta$  then
13:        queue[e][ $i_\delta$ ]  $\leftarrow x$ 
14:        if #queue[e][ $i_\delta$ ] = Q then
15:          fc_dev[e][ $i_\delta$ ] = Std_Deviation(queue[e][ $i_\delta$ ])
16:        end if
17:      end if
18:    end for
19:  end for
20: end while
   Out of reporting interval
21: for  $e \in \{1, 2, \dots, E\}$  do
22:   AS'[e]  $\leftarrow$  queue[e][1][-1]
23: end for
24: Update_DD(day)
25: if  $\sum (AS'[*][\text{day}]) > 10^4$  then
26:   Prepare_To_Battle()
27:   return
28: end if
29:  $\delta_{\text{conf}} \in N^E$ 
30: dev  $\in R^E$ 
31:  $c \in M_{E \times 2}(R)$ 
32: for  $e \in \{1, 2, \dots, E\}$  do
33:   if  $e \in \text{candidates}$  then
34:      $i_\delta = \text{Max}(\{1, \#\text{fc\_dev}[e]\})$ 
35:      $\delta_{\text{conf}}[e] = i_\delta$ 
36:     dev[e] = fc_dev[e][ $i_\delta$ ]
37:   else
38:      $i_\delta = i_{\text{delta}}[e]$ 
39:      $\delta_{\text{conf}}[e] = 0$ 
40:   end if
41:   if fc_dev[e][ $i_\delta$ ]  $\neq \emptyset$  then
42:     c[e][1] = AS'[e][-1] + K · fc_dev[e][ $i_\delta$ ]
43:     c[e][2] = AS'[e][-1] - K · fc_dev[e][ $i_\delta$ ]
44:   else
45:     c[e][1] = AS'[e][-1]
46:     c[e][2] = AS'[e][-1]
47:   end if
48: end for
49: while  $\sum (\delta_{\text{conf}}) > \#\text{candidates}$  and
    $\exists I \in \{1, 2\}^E / \sum (\{c[e][I[e]], \forall e \in \{1, 2, \dots, E\}\}) \geq 10^4$  do
50:    $i = \text{Max}(\{\text{dev}[j], \forall j \in \{h \in \text{candidates} / \delta_{\text{conf}}[h] > 1\}\})$ 
51:    $e = \text{IndexOf}(\text{dev}, i)$ 
52:    $\delta_{\text{conf}}[e] = \delta_{\text{conf}}[e] - 1$ 
53:    $i_\delta = \delta_{\text{conf}}[e]$ 
54:   dev[e] = fc_dev[e][ $i$ .]

```

```

55:   c[e][1] = AS'[e][-1] + K · fc_dev[e][i_δ]
56:   c[e][2] = AS'[e][-1] - K · fc_dev[e][i_δ]
57: end while
58: for e ∈ candidates do
59:   if i_delta[e] ≠ δ_conf[e] then
60:     i_delta[e] = δ_conf[e]
61:     Send_Pigeon_Back(Δ[i_delta[e]])
62:   end if
63: end for
64: day = day + 1

```

8.4.5 Generalisation: DRACE algorithm

Generally speaking, the algorithm consists of collecting data before sending it. How long the data is accumulated depends on the condition satisfaction forecast. In a network context, the castle and the explorer play the roles of sink node and sensor node, respectively. Due to this retaining mechanism, the updating of the data in the sink node is exposed to opportunistic periods of misinformation, resulting in a less accurate time signal when convenient. For all of this, the algorithm is called *DRACE*, acronym of Data Retaining Algorithm for Condition Evaluation.

An interesting aspect of the algorithm is that the data sending criterion is expressed in magnitude by means of δ . Each possible δ value is associated with a queue of Q elements. Due to the filling process, the higher the δ , the greater the time interval represented in the associated queue (as can be observed in Fig. 8.2, which illustrates the queues filling process). However, the duration of this interval depends on the behavior of the signal, and will adapt to it as the queue is updated.

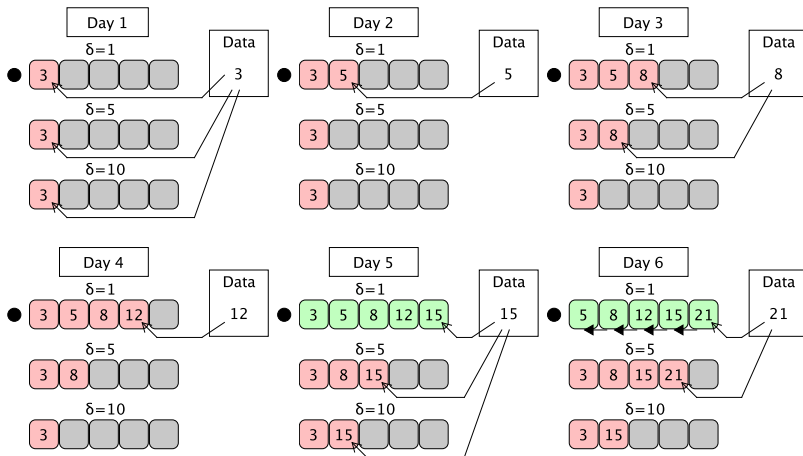


Figura 8.2 Filling process of the queues. The diagram represents an example of the process of filling 3 queues of 5 elements, corresponding to $\delta = 1$, $\delta = 5$ and $\delta = 10$. From left to right and from top to bottom the incorporation of successive data is illustrated. Green elements belong to a complete queue, red elements belong to an incomplete queue, grey elements symbolize absence of value. The black circle indicates the current δ value.

The decision to increase or decrease δ comes from limiting the uncertainty of the next value for the current δ -queue. The prediction method takes into account the standard deviation, which will increase as δ increases. Consequently,

the further away the signal in question is from causing a change in the state of the condition, the greater δ will be which means longer data retaining time in the sender. This magnitude-time relationship is dynamic, and is defined by predictions of the data in queues.

Algorithms 6 and 7 are written in terms of the military simile that represents an application of the proposal. However, the algorithm is applicable to any multivariate condition in a network whose variables are distributed. To do this, some parts of the algorithm that can be adapted are described below.

The sending rule The proposed algorithm uses a well-known and simple sending rule: Send-on-Delta [57] (Alg. 6 L: 5, and Alg. 7 L: 12). This is a magnitude-based sampling technique like others that can be considered for performance improvement.

The forecasting technique The forecast is used to predict the maximum expected increment of time signals, based on the data stored in the queues. In this proposal, the standard deviation multiplied by a factor K is used (Alg. 7, L: 15). This part of the algorithm can be modified to use more sophisticated prediction techniques.

The best candidate selection heuristic In each duty cycle, reporting senders are suitable to be reconfigured with another δ value. The restriction to reconfigure only informants is due to the convenience of allowing sensors to deactivate the reception of network packets. In this way, only the sensors that send information will remain listening for a short period as they are susceptible to response. The way the new configuration is decided involves a process that reduces the δ value of the best candidate in each iteration. In the proposed example all signals participate equally in the satisfaction of the condition, so the candidate with the highest expected increase is wanted (Alg. 7, L: 50-51). This part of the algorithm should be adapted to the condition to be evaluated.

The condition The conditional expression is embedded in the castle algorithm (Alg. 7, L: 25, 49). In addition, the execution of the algorithm ends in the satisfaction of the condition, as an adaptation of the proposed military simile. To apply the algorithm in another context where the sink node reports the state of the condition, this code should check if the condition changes state, not if it is true.

Finally, regarding the update of defensive devices, algorithm 5 represents a process of relocation of resources ballasted by parameter ϵ , which establishes a limit of units per day. By replacing defensive devices with units in which the evolution of a reactive procedure can be measured, this algorithm introduces the up-to-date degree of information required.

8.5 Case Study

This section illustrates how the algorithm works. For this, a parametric configuration of the model has been chosen as an example, the influence of parameter variation is addressed in Section 8.6. The model described in the previous section is incomplete without enemy recruitment data. These are the $AS[e] \forall e \in \{1, 2, \dots, E\}$ signals. In this example, data is generated with the function defined in (8.3), which corresponds to the sum of a line without offset and slope M , and a non-shifted sinusoid of amplitude A and period T . Each $AS[e]$ time series has been generated with the combination of M , A , and T shown in Table 8.2. Fig. 8.3 represents the evolution of the three signals, the sum of all, the threshold of the condition from day 0 to the day of the battle, and the value of each $AS[e]$ signal on the day of the attack (X). Both (8.3) and

		Parameters		
		M_e	A_e	T_e
AS[e]	e=1	4	200	400
	e=2	5.5	350	200
	e=3	7	500	100

Tabla 8.2
Parameter values M , A , and T for each AS[e] signal.

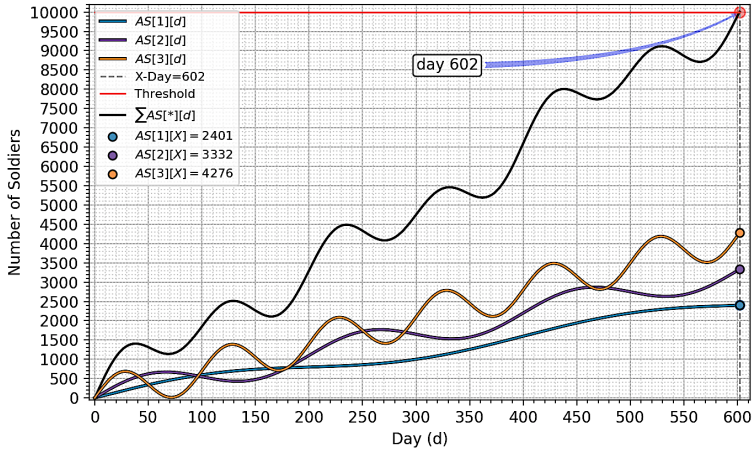


Figura 8.3
Evolution of the number of soldiers in enemy camps ($AS[e][d] = M_e \cdot d + A_e \cdot \sin(\frac{2 \cdot \pi}{T_e} \cdot d)$). The total is represented in black, and the threshold of the attack condition, in red. For this configuration, the battle takes place on day 602, in which the total reaches 10009 soldiers.

the parameter values of Table 8.2 have been chosen as an example, in order to add variability to the case study data.

$$f(x) = M \cdot x + A \cdot \sin\left(\frac{2 \cdot \pi}{T} \cdot x\right) \quad (8.3)$$

The strategy of the council (DRACE algorithm) is applied to this scenario with the parameters summarized below (values according to subsections 8.4.2 and 8.4.3).

TDS 3000.

TDD 150.

λ 50.

ϵ 1.

Δ {1, 25, 50, 100, 200, 300}.

DPM 30.

Q 10.

K 3.

The result is illustrated in Fig. 8.4 and 8.5. Fig. 8.4 shows the information obtained in the castle during the strategy operation. The colour grading of the AS[e] signals reveals the degree of retaining that is set for each of them. The

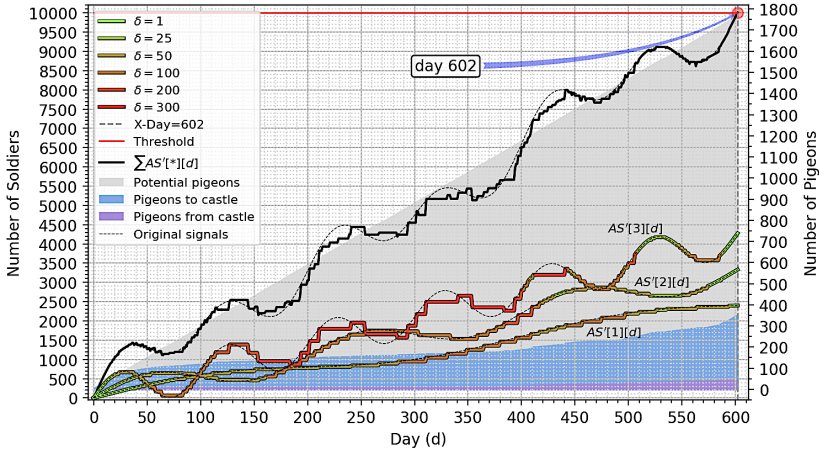


Figura 8.4

Evolution of the $AS'[e]$ signals, which represent the most up-to-date information known at the castle on a daily basis. These signals are coloured on a scale from green to red, representing the δ value of the corresponding explorer at the time of sending. The total is represented in black and the threshold of the condition in red. As a reference, the original signals are shown in dotted lines. In gray, the cumulative number of pigeons that would have been sent if the DRACE algorithm were not applied. The blue area represents the cumulative number of pigeons sent from explorers, and the purple area the cumulative number of pigeons sent from the castle, both add up to the number of pigeons sent using the DRACE algorithm.

higher the δ , the greater the data retaining and the longer the disinformation period in the castle. The presence of longer constant sections in the parts of higher δ values reveals this fact.

Both edges of the lines are characterized by lower δ values, due to different reasons. In the first few days, $\delta = 1$ due to the queuing process. In the days prior to the bout, δ is minimum due to the forecast of condition compliance. The intermediate period is used to avoid traffic, and it can be noticed here how signal $AS'[3]$ is subject to greater retaining. This is because signal $AS'[3]$ has greater variability and, therefore, the queues corresponding to higher δ values are filled, which does not happen in $AS'[1]$ and $AS'[2]$. Fig. 8.6 confirms this fact, by displaying the number of messages of each signal $AS'[e]$ in relation to the corresponding δ value.

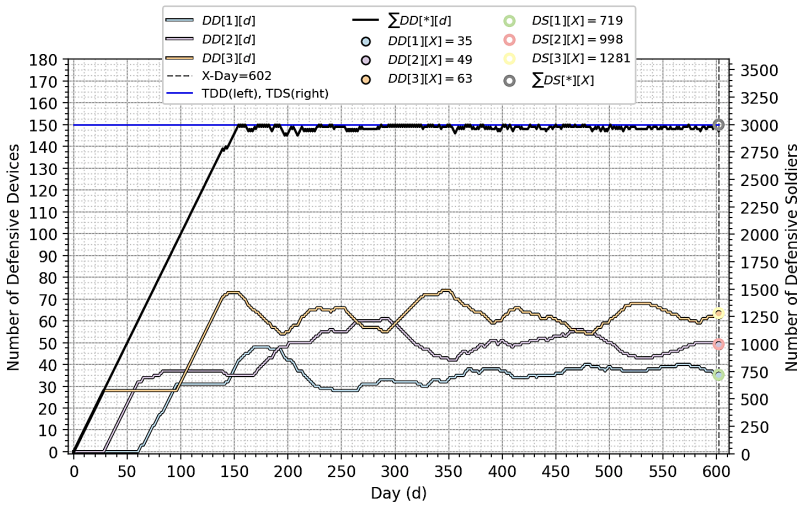


Figure 8.5 Evolution of the defense preparations carried out in the castle. $DD[e]$ signals correspond to the number of defensive devices present in each gate, and evolve according to Algorithm 5. The marks represent the final number of soldiers ($DS[e][X]$) and devices $DD[e][X]$ defending each door. It is important to note that soldiers are not distributed at the gates until the day of battle. The blue line represents the total number of soldiers (TDS) and devices (TDD) available for defence (on both axes).

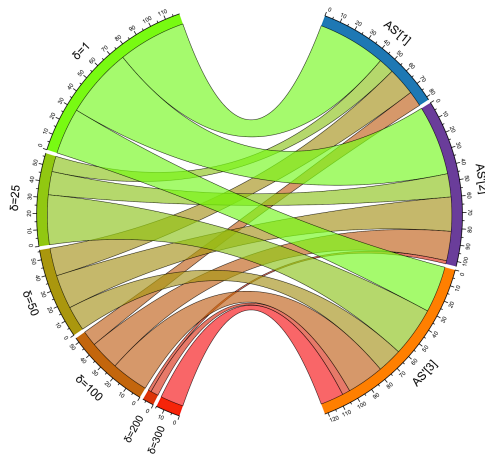


Figure 8.6 Number of pigeons received from each enemy camp ($AS'[e]$ signals) differentiated according to δ value of the corresponding explorer at the time of sending. The diagram has been generated with the R library *Circlize* [11].

8.6 Experiments

This section presents a battery of experiments to analyze the performance of the DRACE algorithm. Starting from an initial configuration, each experiment addresses a variation of the parameters to highlight their influence on the performance. Each experiment consists of a pool of executions of the algorithm using the same set of randomly generated pseudo-realistic signals. Two metrics are extracted from each execution, revealing traffic savings and average accuracy and recall in condition evaluation. Finally, each experiment is shown in a scatter plot that represents both metrics, where each point corresponds to an execution. The following subsections deal with the initial configuration and its variations, the signal generation, and metrics. A final subsection presents and discusses the results.

8.6.1 DRACE configuration parameters

The DRACE application scenario requires a multivariate condition and a series of configuration parameters described in Section 8.4. Section 8.5 deals with a case study as a demonstrator, in which it is required to accurately detect when a condition is satisfied, only once. However, a realistic DRACE application scenario consists of the continuous evaluation of the status of a condition. For this, Algorithm 7 never returns, and the condition satisfaction test (Alg. 7, L:25, 49) goes on to check whether the condition state has changed, as discussed in Subsection 8.4.5. This is the initial configuration:

Condition Expression $C[t] = \sum_{i=1}^3 (S[i][t]) \geq 0.3t$ the state of the condition at instant t . It depends on the value of 3 sensor signals, given by matrix $S \in M_{3 \times 1000}(\mathbb{R})$, which is chosen from a group of pseudo-realistic random signals of length 1000 limited between 0 and 100. This procedure is addressed in the following subsection. In this case, the threshold is the increasing diagonal that covers from minimum to maximum on both axes.

Forecasting and candidate selection Both are maintained as in the case study (Section 8.5). Concerning the multiplier factor of the standard deviation, the experiment is performed with $K = 1$, $K = 2$, and $K = 3$.

Δ -Scale $\Delta = \{0.01, 1, 2.5, 5, 10\}$. The values of the scale have been selected by intuition, pretending a valid configuration that remains constant in all experiments.

Data per message (DPM) This parameter defines the maximum size of the retaining buffer in the sensors. $DPM = 30$ in all experiments.

Size of forecasting queues (Q) The experiment is performed with $Q = 5$, $Q = 10$, and $Q = 20$.

With the rest of the parameters remaining constant, the values of Q and K give rise to 9 parameter combinations. Each combination is an experiment, whose results are discussed later.

8.6.2 Randomly generated pseudo-realistic signals

SysGpr is a tool for generating pseudo-realistic random signals [20]. The underlying algorithm uses statistical distributions to generate successive increments of the resulting signal. For each sample, the parameters of these statistical distributions are randomly altered using the same mechanism with another configuration. This concept gives rise to a recursive algorithm with a set of nesting levels. As a result it generates continuous but not necessarily monotonous

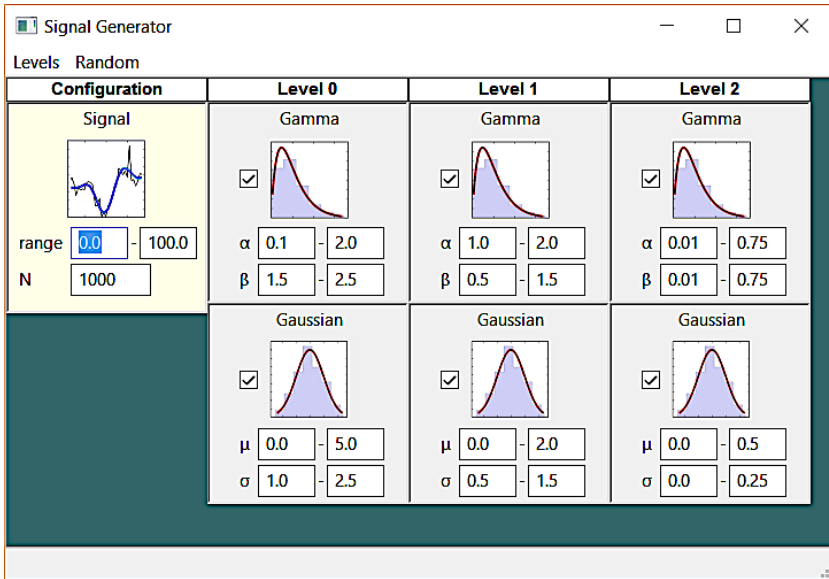


Figura 8.7

SysGpr configuration for the generation of a repository of 10000 pseudo-realistic signals of 1000 samples in the range 0-100. Each level delimits the configuration parameters of the statistical distributions available to generate increases in the previous level.

signals. With proper normalization, these signals are indistinguishable from measurements of many actual phenomena, as the paper concludes by 3 validation methods.

In this paper, each combination of DRACE parameters shown in Subsection 8.6.1 is an experiment. Each experiment involves 1000 executions, and each execution requires a different $S \in M_{3 \times 10^3}(\mathbb{R})$ matrix. Considering that the same dataset is used in all experiments, let us define this dataset as $D \in (M_{3 \times 10^3}(\mathbb{R}))^{10^3}$, and it is obtained as follows:

1. Let be $R \in M_{10^4 \times 10^3}(\mathbb{R})$ a repository of 10^4 signals generated with SysGpr. Each signal has 10^3 samples between 0 and 100 each. Fig. 8.7 illustrates the configuration used. These parameters have been selected by trial and error, to obtain a set of signals whose increments are reasonably limited in relation to the range 0-100, without detriment to diversity. Fig. 8.8 represents some of the signals obtained as a result of this procedure.
2. Iteratively, while $\#D < 10^3$:
 1. 3 signals are randomly selected from R , resulting in an S matrix.
 2. $C = \{\sum S[*][x] \geq 0.3x, \forall x \in \{1, 2, \dots, 10^3\}\}$ is calculated.
 3. Considering $true = 1$ and $false = 0$, if $450 \leq \sum C \leq 550$ then $D \leftarrow S$, and the signals forming S are removed from R . This condition ensures a balanced result in terms of true and false samples.

8.6.3 Metrics

As a result of each execution there are two Boolean signals. The first, $C = \{\sum S[*][x] \geq 0.3x, \forall x \in \{1, 2, \dots, 10^3\}\}$, represents the actual state of the condition, evaluated with the original signals. The second, $C' = \{\sum S'[*][x] \geq 0.3x, \forall x \in \{1, 2, \dots, 10^3\}\}$,

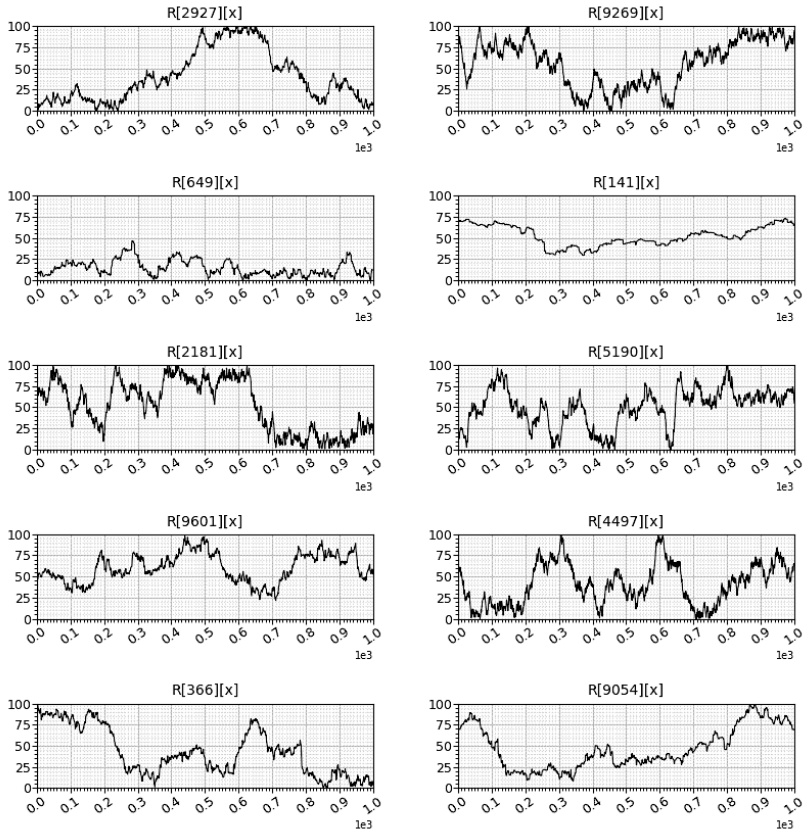


Figure 8.8
Set of signals randomly selected from the R repository.

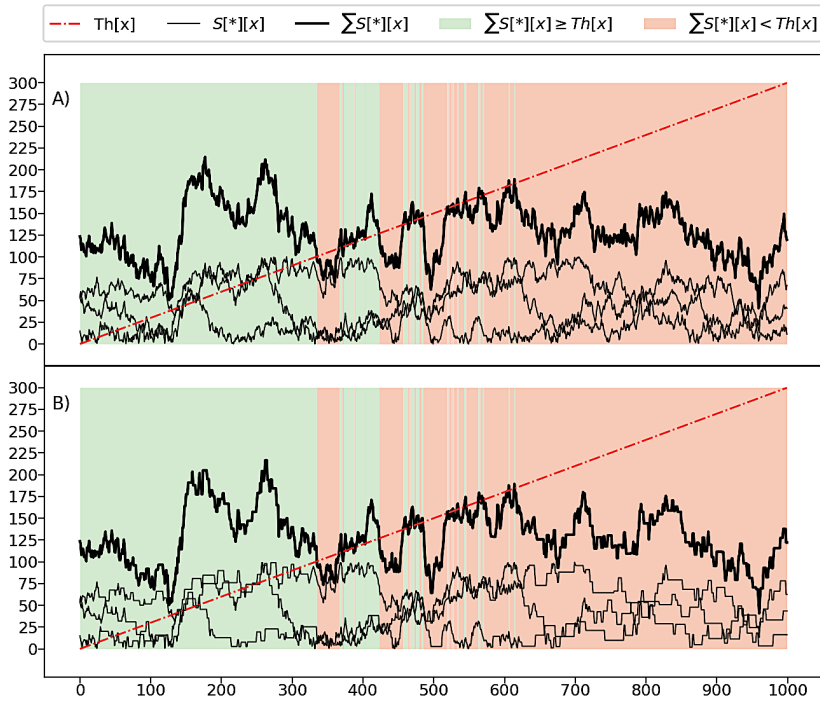


Figura 8.9

Example of execution with $K = 2$ and $Q = 10$. The signals of S are represented by a thin stroke, and the sum, by a thick stroke. The threshold is represented by a red dotted line. B) and A) show the evaluation of the condition with and without DRACE, respectively.

is the state of the condition evaluated with the information obtained by applying the DRACE algorithm, exposing the sensors to data retaining. Fig. 8.9 shows an example of execution, the upper part (A) shows the evaluation of the condition with the original signals, and the lower part (B) with the signals under retaining (DRACE). C and C' are represented with background colors, green for true and red for false. As expected, the result of the condition on both charts is identical. However, B) represents signals with constant segments. These segments correspond to periods without data updates, as a consequence of the traffic reduction strategy of the DRACE algorithm.

As a consequence of retaining data with DRACE, the number of packets required by S' data is lower than that required by S data. However, DRACE is not an error-free technique. An out-of-forecast value can cause an undetected condition change, which introduces discrepant samples in C and C' . This effect is avoided as long as the forecast is guaranteed, but this is detrimental to traffic savings, because wider margins are required in calculations, which incur in more cautious δ configurations. It is reasonable to assume that the two effects are opposite and that a compromise solution is required. This experimentation is intended to clarify this point empirically.

Paper [21] delves into the effect of data resolution on the result of conditional expressions. The methodology proposes metrics to discrepancies between the obtained and actual Boolean signals in terms of precision and recall. For this purpose, instead of evaluating discrepancies on a sample-by-sample basis, transitions from true to false and vice versa are considered. In this calculation the variable τ is introduced, which models a tolerable delay margin between

the actual transition and the detected transition. In this experiment these metrics are applied without delay tolerance ($\tau=0$), which simplifies the calculation.

Equations (8.4), (8.5), (8.6), and (8.7) describe the calculation of transition masks. These masks are binary data arrays with 1s only in those samples where there is a transition from false to true for C_t and C'_t , and from true to false for C_f and C'_f . Note that symbols \oplus and \odot correspond to the logic operations XOR and XNOR, respectively.

$$C_t = \left\{ \left\{ \begin{array}{ll} C[i] & \text{if } i = 1 \\ (C[i] \oplus C[i-1]) \cdot C[i] & \text{if } i > 1 \end{array} \right. \right\}, \forall i \in \{1, 2, \dots, 10^3\} \quad (8.4)$$

$$C_f = \left\{ \left\{ \begin{array}{ll} |C[i] - 1| & \text{if } i = 1 \\ (C[i] \oplus C[i-1]) \cdot C[i] & \text{if } i > 1 \end{array} \right. \right\}, \forall i \in \{1, 2, \dots, 10^3\} \quad (8.5)$$

$$C'_t = \left\{ \left\{ \begin{array}{ll} C'[i] & \text{if } i = 1 \\ (C'[i] \oplus C'[i-1]) \cdot C'[i] & \text{if } i > 1 \end{array} \right. \right\}, \forall i \in \{1, 2, \dots, 10^3\} \quad (8.6)$$

$$C'_f = \left\{ \left\{ \begin{array}{ll} |C'[i] - 1| & \text{if } i = 1 \\ (C'[i] \oplus C'[i-1]) \cdot C'[i] & \text{if } i > 1 \end{array} \right. \right\}, \forall i \in \{1, 2, \dots, 10^3\} \quad (8.7)$$

Transition precision is defined as the proportion of detected transitions that are real, and transition recall as the proportion of actual transitions that are detected. Equations (8.8) and (8.9) describe their calculation from the transition masks, respectively.

$$TP = \frac{\sum_{i=1}^{10^3} (C'_t[i] \odot C_t[i]) + \sum_{i=1}^{10^3} (C'_f[i] \odot C_f[i])}{\sum_{i=1}^{10^3} (C'_t[i] + C'_f[i])} \quad (8.8)$$

$$TR = \frac{\sum_{i=1}^{10^3} (C'_t[i] \odot C_t[i]) + \sum_{i=1}^{10^3} (C'_f[i] \odot C_f[i])}{\sum_{i=1}^{10^3} (C_t[i] + C_f[i])} \quad (8.9)$$

As a unified measure of the accuracy of the result, the metric used is the harmonic mean of precision and recall, usually denoted by F1-score, the calculation of which is defined in (8.10).

$$F1score = 2 \cdot \frac{TP \cdot TR}{TP + TR} \quad (8.10)$$

With regard to traffic savings, the Traffic Saving Ratio (TSR) is defined as the proportion of transmitted packets in relation to those that would have been sent without DRACE. For experimentation, counters are introduced in Algorithm 7 to obtain the number of data packets (from the sensors, line 7) and configuration packets (from the sink node, line 61) transmitted. NDP and NCP are the variables reflecting these scores, respectively. The number of packets

required to transmit the original signals (S) is calculated assuming that all samples that differ from the previous one are sent. Equation (8.11) defines the TSR calculation, assuming the logic expression returns 1 if true and 0 if false.

$$\text{TSR} = 1 - \frac{\text{NDP} + \text{NCP}}{\sum_{i=1}^3 \left(1 + \sum_{j=2}^{10^3} (S[i][j] \neq S[i][j-1]) \right)} \quad (8.11)$$

F1-Score and TSR are the metrics that model the incidence of the two opposite effects described on a 0-1 scale for each execution. F1score = 0 corresponds to an execution in which all detected transitions are erroneous, and F1-score = 1 the opposite. TSR = 0 corresponds to an execution in which traffic is not avoided, and TSR = 1 in which all the traffic is avoided (unreal case). TSR can acquire negative values, which are interpreted as sending more traffic than necessary without DRACE.

8.6.4 Results and discussion

As described above, DRACE has two effects, traffic savings and inaccuracy in condition detection. To analyse the performance of the algorithm, Subsection 8.6.3 describes two metrics to measure the incidence of both in each execution. Concerning the design of the experiment, there are two important aspects: the DRACE parameter set and the test data. Subsection 8.6.1 presents the set of experimental parameters. $\Delta = \{0.01, 1, 2.5, 5, 10\}$ and $\text{DPM} = 30$ remain constant, while K and Q are vary, resulting in 9 experimental parameterizations ($K \in \{1, 2, 3\} \times Q \in \{5, 10, 20\}$). This decision is taken because it is reasonable to assume that their configuration has a greater impact on the effects explained, as they are related to the precision and margins of the forecast. Regarding the test data, large amounts of pseudo-realistic signals have been generated to represent as many different scenarios as possible, as described in Subsection 8.6.2.

According to this methodology, 9 experiments are performed, one per each parametric configuration. Each experiment consists of 1000 executions in which DRACE is applied to the triads of random signals of $D \in (M_{3 \times 10^3}(\mathbb{R}))^{10^3}$, obtaining TSR and F1-score in each case.

The results are shown as a scatter plot matrix in Fig. 8.10. Each graph contains 10^3 marks, one for each triad of signals tested. The size of the marks corresponds to the number of transitions obtained. The experiments present an average of 15 transitions, with 1 being the minimum and 63 the maximum. Each mark is semi-transparent to allow intuition of data frequency by overlap. Also, population mean (μ_{xy}) and confidence ellipses ($\mu_{xy} \pm \alpha \cdot \sigma_{xy}, \forall \alpha \in \{1, 2, 3\}$) are represented in each scatter plot. These ellipses represent the iso-contour of the Gaussian distribution, containing 68%, 95%, and 99% of the samples, with $\alpha = 1$, $\alpha = 2$, and $\alpha = 3$ respectively [33].

Suppose $G[i][j]$ is the graph of row i and column j . The diagonal formed by $G[3][3]$, $G[2][2]$, and $G[1][1]$ represents the transition from the most reliable configuration for forecasting to the least reliable one. As expected, the more reliable the prediction, the more accurate the condition evaluation and the lower the traffic saving ratio.

It can be observed how in extreme cases ($G[1][1]$ and $G[3][3]$), the dispersion is reduced in the favourable metric and increased in the unfavourable

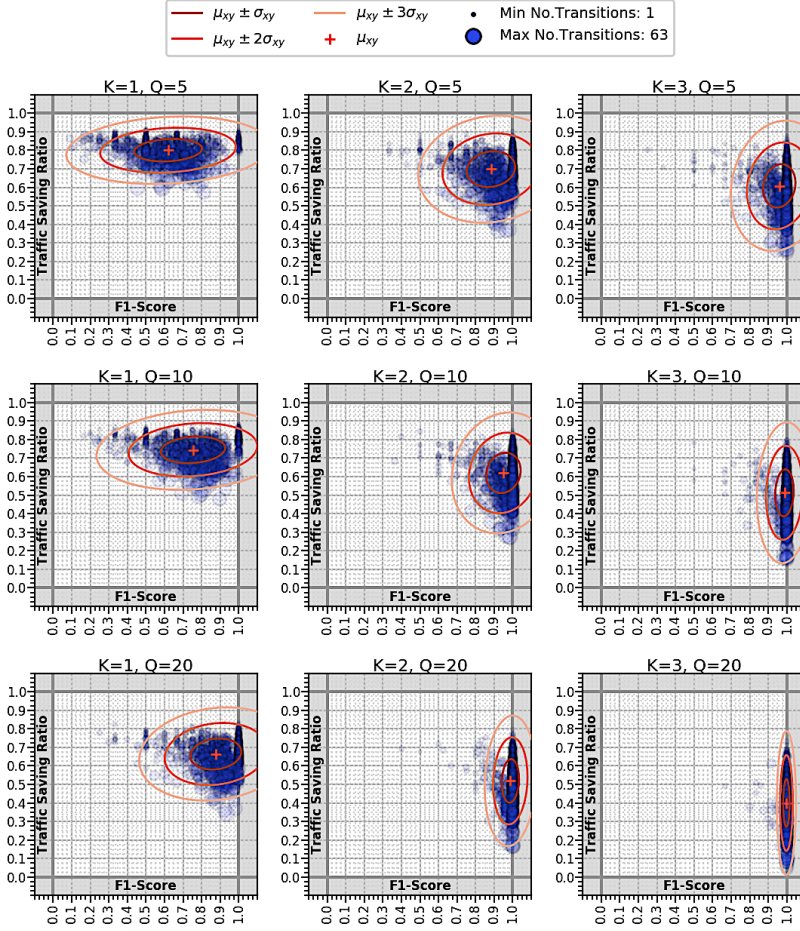


Figure 8.10

Matrix of scatter diagrams with the results of the experimentation. The results of the experiments for increasing Q values are shown from top to bottom. From left to right for increasing K values. In each diagram, 10^3 executions of DRACE are represented by semi-transparent blue marks, proportionally sized to the number of transitions detected in that execution. In red tones, the mean population (μ_{xy}) and iso-contours of the Gaussian distribution for 1σ , 2σ , and 3σ are represented, covering the 68, 95 and 99 percent of the points, respectively.

one. This question is related to the signals behaviour and how they interact with the threshold. DRACE takes advantage of opportunity periods when the condition is far from a change of state. The more periods of opportunity with more distance to the change of state of the condition, the more cautious can be the configuration of the forecast without affecting the reduction of traffic. In diagrams $G[1][1]$ and $G[3][3]$, this casuistry is dispersed with respect to the metric that the experiment is not conditioning.

K is a multiplication factor of the maximum deviation forecasted, while Q sets the sample size on which the forecasting is performed. Therefore, both variables influence the probability of success in prediction. K by introducing a post-forecast margin, and Q taking into account a wider range of data and, consequently, considering a longer signal history. The effects of both parameters are similar, as evidenced by the similarity of the $G[3][2]$ and $G[2][3]$, $G[3][1]$ and $G[1][3]$, and $G[2][1]$ and $G[1][2]$ diagrams.

There is an inversely proportional relationship between the number of transitions and traffic saving ratio, this can be appreciated particularly clearly in the diagrams in column 3. This is due to the fact that as transitions increase, so do the changes in condition statuses, which statistically reduces the opportunity periods for retaining data.

Finally, considering the wide variety of cases represented, some dispersion in both axes was expected in all cases. However, the most accurate configurations present a strong sample convergence at $F1\text{-score} = 1$, avoiding, in 68 % of the cases, between 30 and 50 % of the traffic. Although this experiment does not demonstrate the relevance of using DRACE in all scenarios, it seems reasonable to conclude that it is convenient to consider its application, since for most of the test scenarios using random signals, it can avoid a large considerable portion of the traffic without committing errors.

8.7 Conclusions

This paper deals with the evaluation in time of multivariate conditions when variables are distributed. When the multivariate condition is the expression of a threshold-based event, early detection requires continuous monitoring of all involved variables. Algorithm DRACE is proposed from the perspective of a centralized duty cycle solution. This algorithm regulates a data retaining strategy in the data sources, attempting to avoid network packets without affecting the result of the event detection.

DRACE is based on data forecasting to take advantage of periods in which a change in condition status has a remote probability. During these periods and depending on this calculation, it reconfigures the data sources in order to retain the information proportionally. Data retaining is a consequence of the well-known Send-on-Delta mechanism. It is therefore a magnitude-based technique, not a time-based one. This aspect provides robustness, since possible out-of-forecast data are sent even in a regime of strong data retaining, as long as its magnitude is a trigger condition for sending.

In an effort to illustrate the motivation and promote understanding of its operation, DRACE is presented on a warlike simile that deals with all intrinsic aspects of the problem in another context. This contextualized statement is at the disposal of the scientific community, and we hope that it will be motivating for other authors to propose alternative solutions.

To analyze the performance of DRACE, 9000 executions of the algorithm have been performed. 9 configurations tested on a repository of 3 million data organized in 1000 triads of signals of 1000 samples from 0 to 100 each. The data has been generated with SysGpr, a validated technique that synthesizes continuous pseudo-realistic signals suitable for testing purposes. The traffic savings and the accuracy of the event detection are measured in each execution. The metric used to measure accuracy is based on transitions in the resulting Boolean signal, allowing an analysis based on the harmonic mean of precision and recall.

The experiments reveal the need for a compromise solution between accuracy and traffic savings in the parameter configuration. Nevertheless, the obtained results are promising. Focusing on the most accurate configuration, 99% of executions are error-free, and the number of packets is reduced by 40% on average, being between 30 and 50% in 68% of cases. If some degree of inaccuracy is tolerable, DRACE can reduce traffic by more than 50% with an event detection accuracy greater than 90% in most cases.

8.8 Future Work

This work is part of a research line related to dynamic management of network resources for the efficient deployment of distributed intelligence models. [21] is a previous work which addresses the reduction of the information domain to optimize network traffic, avoiding congestion. DRACE algorithm is an important contribution in this topic, providing an adaptive mechanism based on data retaining. However, there are many aspects not yet addressed for which it will provide an adequate grounding. Some of them are discussed here.

- The proposed procedure causes that all the data is sent (Alg. 6, L: 4). Possible errors in the condition evaluation are due to the fact that the information may not be available on time. This feature allows the sink to produce statistics such as those described in the experimentation. Based on these statistics, it is possible to develop machine learning models in order to seek the optimal DRACE configuration at runtime for a preset level of accuracy or traffic saving ratio.
- DRACE does not reduce the bit size of data. The DPM (Data Per Message) parameter configures the number of data per network packet, but these are sent unaltered. For applications in which the payload must be used to the maximum, it is possible to consider the reduction of bits per data [21], in addition to their retaining. This feature could be especially useful in multimedia sensor network applications where it is required to detect events in a massive data stream, such as video surveillance.
- Parameter Δ corresponds to a scale of magnitudes (δ) that affect the retaining method by means of the Send-on-Delta concept [57]. This scale should be established according to the behaviour and margins of the signals to be monitored. The in-depth analysis of the optimal configuration of this parameter is beyond the scope of this paper, but it would be interesting to address it in the future. In the same way, the restriction of a single Δ for all inputs should be reconsidered, especially in systems that require heterogeneous information.
- As discussed in Subsection 8.4.5, the forecasting technique and the next candidate selection heuristic for seeking the best δ -configuration are subject to reconsideration for improving algorithm performance.

REFERENCES

- [1] Christos Antonopoulos, Sofia Maria Dima, and Stavros Koubias: *Event Identification in Wireless Sensor Networks*. In *Components and Services for IoT Platforms: Paving the Way for IoT Standards*, chapter 10, pages 187–210. Springer International Publishing Switzerland, 2017, ISBN 9783319423043. (Citado en la página 18.)
- [2] K.J. Astrom and B.M. Bernhardsson: *Comparison of Riemann and Lebesgue sampling for first order stochastic systems*. In *Proceedings of the 41st IEEE Conference on Decision and Control*. IEEE, 2002. (Citado en la página 21.)
- [3] Hany Atlam, Robert Walters, and Gary Wills: *Fog Computing and the Internet of Things: A Review*. *Big Data and Cognitive Computing*, 2(2):10, 2018, ISSN 2504-2289.
- [4] Costin Badica, Lars Braubach, and Adrian Paschke: *Rule-based distributed and agent systems*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6826 LNCS:3–28, 2011, ISSN 03029743.
- [5] Majid Bahrepour, Nirvana Meratnia, Paul Havinga, and Pervasive Systems Group: *Automatic fire detection : a survey from wireless sensor network perspective*. Enschede: Centre for Telematics and Information Technology (CTIT)., 2008, ISBN 1381-3625.
- [6] R.a Bi, H.a Gao, and Ya b Li: *Probabilistic threshold based monitoring using sensor networks*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8491:246–255, 2014. (Citado en la página 20.)
- [7] Giuseppe Campobello, Antonino Segreto, and Salvatore Serrano: *Data Gathering Techniques for Wireless Sensor Networks: A Comparison*. *International Journal of Distributed Sensor Networks*, 2016, 2016, ISSN 15501477.
- [8] Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner: *A Survey about Prediction-Based Data Reduction in Wireless Sensor Networks*. *ACM Computing Surveys*, 49(3):1–35, 2016.
- [9] Norman Dziengel: *Distributively Observed Events in Wireless Sensor Networks*. PhD thesis, Freie Universität Berlin, Germany, 2015.
- [10] Daniel Goldsmith and James Brusey: *The Spanish inquisition protocol - Model based transmission reduction for wireless sensor networks*. *Proceedings of IEEE Sensors*, pages 2043–2048, 2010.
- [11] Zuguang Gu, Lei Gu, Roland Eils, Matthias Schlesner, and Benedikt Brors: *Circlize implements and enhances circular visualization in R*. *Bioinformatics*, 30(19):2811–2812, 2014, ISSN 14602059.
- [12] Sarra Hammoudi, Zibouda Aliouat, and Saad Harous: *Challenges and research directions for Internet of Things*. *Telecommunication Systems*, 67(2):367–385, 2018, ISSN 15729451.
- [13] Malik Hasnain, Mazhar Hussain Malik, and Mehmet Emin Aydin: *An adaptive opportunistic routing scheme for reliable data delivery in WSNs*. pages 1–7, 2018.

- [14] Fei Hu and Qi Hao (editors): *Intelligent Sensor Networks*. CRC Press, Boca Raton, 1st edition, 2013, ISBN 9781138199743.
- [15] Naveed Ilyas, Mariam Akbar, Rehmat Ullah, Muhammad Khalid, Arsalan Arif, Abdul Hafeez, Umar Qasim, Zahoor Ali Khan, and Nadeem Javaid: *SEDG: Scalable and efficient data gathering routing protocol for underwater WSNs*. *Procedia Computer Science*, 52(1):584–591, 2015, ISSN 18770509.
- [16] Krasimira Kapitanova, Sang H. Son, and Kyoung Don Kang: *Using fuzzy logic for robust event detection in wireless sensor networks*. *Ad Hoc Networks*, 10(4):709–722, 2012, ISSN 15708705. (Citado en la página 20.)
- [17] Mitchell C Kerman, Wei Jiang, Alan F Blumberg, and Samuel E Buttrey: *Event detection challenges, methods, and applications in natural and artificial systems*. *Proceedings of the 14th International Command and Control Research and Technology Symposium*, (March 2009):1–19, 2009. (Citado en la página 19.)
- [18] Youngsoo Kim, Daeyoung Kim, Poh Kit Chong, Jonggu Kang, Eunjo Kim, and Suckbin Seo: *Design of a Fence Surveillance System based on Wireless Sensor Networks*. *Proceedings of the 2nd International ICST Conference on Autonomic Computing and Communication Systems*, 2008, ISSN 1474-547X (ELECTRONIC) 0140-6736 (LINKING).
- [19] Ja Won Ko and Yoon Hwa Choi: *A grid-based distributed event detection scheme for wireless sensor networks*. *Sensors*, 11(11):10048–10062, 2011, ISSN 14248220. (Citado en la página 20.)
- [20] F León, Fco. J. Rodríguez-Lozano, José M Palomares, and J Olivares: *SysGpr: Sistema de Generación de Señales Sintéticas Pseudo-realistas*. *Revista Iberoamericana de Automática e Informática Industrial*, 16(3):369–379, 2019.
- [21] Fernando Leon-Garcia, Jose Manuel Palomares, and Joaquin Olivares: *D2R-TED: Data-Domain reduction model for threshold-based event detection in sensor networks*. *Sensors*, 18(11):3806, 2018, ISSN 14248220.
- [22] Yingshu Li, Chunyu Ai, Chinh Vu, Yi Pan, and Raheem Beyah: *Delay-bounded and energy-efficient composite event monitoring in heterogeneous wireless sensor networks*. *IEEE Transactions on Parallel and Distributed Systems*, 2010, ISSN 10459219.
- [23] David James McCorrie, Elena Gaura, Keith Burnham, Nigel Poole, and Roger Hazelden: *Predictive data reduction in wireless sensor networks using selective filtering for engine monitoring*. In Driss Benhaddou and Ala Al-Fuqaha (editors): *Wireless Sensor and Mobile Ad-Hoc Networks Vehicular and Space Applications*. Springer-Verlag, New York, 2015.
- [24] Umar Ibrahim Minhas, Ijaz Haider Naqvi, Saad Qaisar, Kamran Ali, Saleem Shahid, and Muhammad Awais Aslam: *A WSN for Monitoring and Event Reporting in Underground Mine Environments*. *IEEE Systems Journal*, 12(1):485–496, 2018, ISSN 19379234.
- [25] M. Miskowicz: *The event-triggered sampling optimization criterion for distributed networked monitoring and control systems*. In *IEEE International Conference on Industrial Technology*, 2003. (Citado en la página 21.)
- [26] Marek Miskowicz: *Send-on-delta concept: An event-based data reporting strategy*. *Sensors*, 6(1):49–63, 2006, ISSN 14248220. (Citado en la página 33.)

- [27] Marek Miśkiewicz (editor): *Event-based control and signal processing*. CRC Press, Boca Raton, 2015. (Citado en la página 21.)
- [28] Aziz Nasridinov, Sun Young Ihm, Young Sik Jeong, and Young Ho Park: *Event detection in wireless sensor networks: Survey and challenges*. In James J. (Jong Hyuk) Park, Hojjat Adeli, Namje Park, and Isaac Woungang (editors): *Mobile, Ubiquitous, and Intelligent Computing*, pages 585–590, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. (Citado en la página 19.)
- [29] Taehyeun Park, Nof Abuzainab, and Walid Saad: *Learning How to Communicate in the Internet of Things: Finite Resources and Heterogeneity*. IEEE Access, 4:7063–7073, 2016, ISSN 21693536.
- [30] M. A. Razzaque, Chris Bleakley, and Simon Dobson: *Compression in wireless sensor networks*. ACM Transactions on Sensor Networks, 10(1):1–44, 2013, ISSN 15504859. (Citado en la página 20.)
- [31] Mehanas Shahul, Sanu Thomas, and R Vijayakumar: *A Review on Event Detection Techniques in Wireless Sensor Network*. International Journal of Innovative Research in Science, Engineering and Technology, 6(7), 2017, ISSN 2319-8753.
- [32] Ikram Ud Din, Mohsen Guizani, Suhaidi Hassan, Byung Seo Kim, Muhammad Khurram Khan, Mohammed Atiquzzaman, and Syed Hassan Ahmed: *The Internet of Things: A Review of Enabled Technologies and Future Challenges*. IEEE Access, 7:7606–7640, 2019, ISSN 21693536.
- [33] Bin Wang, Wenzhong Shi, and Zelang Miao: *Confidence analysis of standard deviational ellipse and its extension into higher dimensional Euclidean space*. PLoS ONE, 10(3):1–17, 2015, ISSN 19326203.
- [34] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees: *Deploying a wireless sensor network on an active volcano*. IEEE Internet Computing, 10(2):18–25, 2006, ISSN 10897801.
- [35] G Wittenburg, N Dziengel, S Adler, Z Kasmi, M Ziegert, and J Schiller: *Cooperative event detection in wireless sensor networks*. IEEE Communications Magazine, 50(December):124–131, 2012, ISSN 0163-6804.
- [36] Xian Ming Zhang, Qing Long Han, and Bao Lin Zhang: *An Overview and Deep Investigation on Sampled-Data-Based Event-Triggered Control and Filtering for Networked Systems*. IEEE Transactions on Industrial Informatics, 13(1):4–16, 2017, ISSN 15513203. (Citado en la página 21.)
- [37] Sebastian Zöllner, Christian Vollmer, Markus Wachtel, Ralf Steinmetz, and Andreas Reinhardt: *Data filtering for wireless sensor networks using forecasting and value of information*. Proceedings - Conference on Local Computer Networks, LCN, (October):441–449, 2013.

Bloque III

DISCUSIÓN Y CONCLUSIONES

ÍNDICE

9.0.1	Fundamentos matemáticos de la reducción de tráfico: el modelo D2R-TED	128
9.0.2	Aplicación adaptativa de la reducción de tráfico: Algoritmo DRACE	129
9.0.3	Generación de señales pseudo aleatorias: SysGpr	130

El panorama tecnológico actual está enfocado a la interconexión de multitud de dispositivos distribuidos en el entorno. Como respuesta a este creciente desafío, la optimización de los recursos de red es un objetivo prioritario a todos los niveles de la infraestructura de comunicaciones. Esta tesis doctoral aborda esta problemática en el entorno local y para una aplicación específica: optimizar los recursos de red para la detección de eventos basados en umbral con redes de sensores. Con este objetivo se ha planteado una solución hipotética y un procedimiento metodológico para demostrarla, resultado del cual se han publicado los artículos abordados en el bloque anterior. En este capítulo se aborda una discusión del trabajo realizado desde la perspectiva general del planteamiento de la tesis.

9.0.1 Fundamentos matemáticos de la reducción de tráfico: el modelo D2R-TED

El modelo D2R-TED y en consecuencia el algoritmo DRACE parten de un principio de simplicidad aplicable a un ámbito más general que el abordado en esta tesis: en la lógica de un proceso de recolección de información para procesar un resultado deseado, los datos que no repercutan en este resultado son prescindibles. Llevar a la práctica esta obviedad conlleva la necesidad de resolver una cuestión no tan obvia: cómo detectar la información no relevante. En el caso que nos ocupa, el resultado a obtener está sujeto a un proceso de evaluación continua, evoluciona en el tiempo al igual que las fuentes de datos de las que depende. Además, este resultado se define en el dominio Booleano, lo que introduce una cuestión interesante que es abordada implícitamente en los fundamentos matemáticos propuestos: la repercusión de los datos de entrada en el resultado pueden medirse en clave temporal. Expresado en otras palabras: no se trata de *cuánto* o de *cómo*, sino de *cuándo*. Esta cuestión es ilustrada en la Figura 7.4, en la que se observa cómo una reducción de información en la entrada de la evaluación de un umbral retrasa o adelanta la conmutación del estado booleano del resultado. Si bien este efecto puede no acontecer, teniendo en su lugar una conmutación errónea o ausencia de ella, es común que ocurra y puede ser medido con las métricas propuestas en las ecuaciones (7.9), (7.10) y (7.11) de la Sección 7.4.2. Para ello se define el parámetro τ en las métricas del modelo D2R-TED, introduciendo así la tolerancia al desfase del sistema (explicada detalladamente en la Figura 7.5). La tolerancia al desfase introduce una interesante interpretación de los resultados: cuando se selecciona un valor de τ tal que los instantes de transición $\pm\tau$ no se solapen (lo cual es muy común), y que haga que la precisión de nivel (LP) saturé a su máximo (proporción de errores cero comparando muestra a muestra las partes de la señal que no están dentro del intervalo del instante de transición $\pm\tau$), se puede afirmar que la señal reducida no presenta intervalos erróneos de más de $2\tau + 1$ muestras. Por ejemplo, los resultados mostrados en la Figura 7.6 revelan que, para $\delta = 0.75$, el intervalo máximo de muestras consecutivas erróneas está acotado a 5, porque $\tau = 2$ hace $LP = 1$.

La Figura 7.6 muestra los resultados obtenidos sobre el conjunto de datos reales de la red meteorológica de Arpa Piemonte (Italia) [7]. Estas gráficas representan las métricas de precisión y exhaustividad de las transiciones y de las señales muestra a muestra variando τ y el parámetro δ . Como era esperable, reinterpretar la salida del sistema con una mayor tolerancia al desfase muestra mejoras en la calidad de la detección, apreciándose en las métricas una evolución en relación a τ que se ajusta a la curva característica de la función de distribución exponencial acumulada $1 - e^{-\alpha(x+\beta)}$. Este comportamiento se observa utilizando tanto señales reales como las señales generadas con SysGPR, y sugiere que el retraso de las transiciones en la señal reducida medido en muestras, sigue una distribución exponencial negativa, siendo las curvas τ su distribución acumulada de probabilidad.

Por otra parte, si bien las tendencias en el comportamiento de las métricas observables en la experimentación con D2R-TED son aceptables como prueba de concepto y, por tanto, válidas para el propósito del artículo; la función de reducción definida para la experimentación ($r(\varphi, \delta_r) = \text{Integer}\left(\frac{\varphi}{\delta_r}\right) \cdot \delta_r$) utiliza un truncado de los datos, lo que provoca que la distribución del error medido en magnitud de los datos reducidos no sea homogénea en toda la escala, sino que presente incrementos en torno a los múltiplos del valor δ utilizado. En la experimentación comparativa cuyos resultados se muestran en la Figura 7.8 se puede observar como pequeños cambios introducidos en la función de

reducción (comparar Algoritmos 2 y 3 con Algoritmo 4) pueden mejorar sustancialmente el resultado de la técnica.

D2R–TED se fundamenta en la reducción de la cardinalidad del dominio de definición de los flujos de datos que sustentan una condición. Como era esperable, se ha observado que la técnica ofrece resultados tanto mejores, cuanto menos variabilidad presentan los incrementos de muestras consecutivas de las señales, siendo inapropiada su utilización en aquellos casos en los que la condición se establece sobre valores que evolucionan en el tiempo como señales discontinuas o aleatorias. Por otra parte, la fundamentación de la lógica de la reducción de datos en los dominios de definición de la información hace que D2R–TED sea aplicable a cualquier escenario donde una condición Booleana esté sujeta a un proceso de aproximación continua al cumplimiento de la condición, sea cual sea el formato de la información considerada. Por ejemplo, en una aplicación de visión artificial que detecte rostros conocidos en imágenes, si el grado de detección es una variable continua, puede modelarse con D2R–TED definiendo apropiadamente la función de reducción (Ecuación 7.3) y equivalencia (Ecuación 7.4) para representar información de una imagen. De igual modo, cualquier abstracción soportada en una estructura de datos puede integrarse en el modelo siguiendo el mismo procedimiento.

9.0.2 *Aplicación adaptativa de la reducción de tráfico: Algoritmo DRACE*

DRACE se desarrolla con el propósito de aplicar las bases conceptuales de D2R–TED de manera adaptativa, utilizando las métricas que propone para minimizar el tráfico y maximizar la calidad de la evaluación de las condiciones. Algunos aspectos relevantes del algoritmo se discuten a continuación.

DRACE centraliza la evaluación de la condición en un proceso que ajusta la parametrización de las fuentes de datos. En este proceso se contempla una escala constante de magnitudes, los posibles valores del parámetro δ en cada generador de información, que en última instancia determina el criterio de envío de los datos. Además, el método predictivo que se encarga de detectar los periodos de oportunidad para reducir información consta de una ventana deslizante de datos de tamaño fijo Q . Estas estructuras de datos se computan para elegir la parametrización idónea en cada caso, y representan las condiciones de la señal en cada fuente de información. La lógica subyacente del algoritmo utiliza los periodos de escasa probabilidad de conmutación en el resultado para reducir el refresco de información de una señal, con un criterio de envío basado en magnitud (δ). Por una parte, es lógico esperar que la señal más variable presente mayores rangos de incertidumbre y sea, por tanto, menos susceptible de reducir sus envíos. Sin embargo hay otro factor importante que influye de forma determinante en este proceso: para que una señal sea configurada con un valor δ concreto, la ventana deslizante con la que se analiza la desviación típica que presenta con el valor δ inmediatamente inferior debe estar completa. Esto hace que las señales con menor variabilidad requieran más tiempo en rellenar esta estructura de datos. Este efecto contrario se puede apreciar en la Figura 8.6 correspondiente al demostrador (caso de estudio), en la que se observa cómo el tráfico correspondiente a configuraciones más restrictivas con el tráfico, corresponden a la señal AS'[3], que es la señal que más desviación típica presenta. Presumiblemente, este efecto se vería compensado en un experimento de mayor duración, en el que todas las ventanas deslizantes de todas las señales se completan en el tiempo en que transcurre la ejecución.

Los resultados mostrados en la Figura 8.8 representan las mediciones obtenidas sobre un extenso dataset de señales aleatorias. En este experimento se analiza la influencia de los parámetros del algoritmo en el resultado. El análisis visual de la casuística revela una tendencia muy clara hacia una mayor precisión al incrementar el tamaño de las ventanas deslizantes (mejorando así el resultado del análisis predictivo) o el factor multiplicativo de la desviación típica (incrementando así el margen de incremento con respecto a la última muestra de las señales que se utiliza en la evaluación de riesgo de conmutación). Igualmente, se observa en consecuencia una reducción en el ahorro de tráfico. Ambos efectos se contraponen, debiendo aplicar una solución de compromiso que dependerá de las características intrínsecas del sistema de detección.

9.0.3 Generación de señales pseudo aleatorias: SysGpr

En las pruebas realizadas con el generador, se ha observado cómo el primer nivel de parámetros para la generación aleatoria genera la forma principal de la señal. Sin embargo se aprecia en estas señales la simplicidad intrínseca de una distribución estadística pura, lo que hace que este tipo de señales sean fácilmente identificables como señales generadas. Es necesario añadir un segundo nivel para hacer indistinguible la señal generada de una posible señal real, como se aprecia en los resultados publicados (Figuras 6.4, 6.5, 6.6 y 6.7).

Si bien la interfaz gráfica hace que la herramienta sea fácil de utilizar y permita generar grandes conjuntos de datos en unos minutos, la configuración de los sucesivos niveles de generación requiere cierto conocimiento del procedimiento subyacente. Si la configuración de un nivel genera valores parecidos a su predecesor las señales tienden a experimentar variaciones muy bruscas. Para obtener señales continuas y con tendencias apreciables, los sucesivos niveles de generación deben configurarse para obtener valores en un rango sustancialmente inferior a los del nivel anterior, como puede apreciarse en la Figura 6.2

CONCLUSIONES

Como se desarrolla en el Capítulo 2, las tendencias del desarrollo de la computación distribuida están orientadas a la optimización de los recursos y al diseño sensible al contexto. Esta tesis está centrada en la optimización del tráfico de redes de sensores que evalúan el estado de condiciones multivariantes y cuyas variables se encuentran distribuidas en la red. El enfoque de las propuestas es la síntesis de técnicas adaptativas, que hacen uso del contexto lógico de la información y de las variaciones en el comportamiento de las señales.

El trabajo desarrollado ha sido objeto de tres publicaciones. La primera, con un objetivo metodológico, presenta una herramienta para generar grandes cantidades de datos, adecuados para simular y validar las propuestas de investigación desarrolladas en las otras dos publicaciones. La segunda aborda los fundamentos matemáticos de la reducción de tráfico en el escenario de aplicación objeto de estudio, aportando las métricas necesarias para medir el volumen de tráfico evitado y la calidad de la respuesta del sistema. Planteada sobre esta base matemática, la tercera publicación presenta un algoritmo capaz de maximizar la reducción de tráfico y minimizar la distorsión en el resultado de la evaluación de la condición simultáneamente.

Del trabajo de esta tesis doctoral se extraen las siguientes conclusiones:

- El software SysGpr, desarrollado como un objetivo colateral y estrictamente metodológico, ha resultado ser una herramienta útil para experimentación y fácil de utilizar, al permitir generar grandes conjuntos de datos aleatorios pero con la variabilidad propia de señales reales.
- Se ha propuesto el modelo D2R-TED que, en redes de sensores cuyo objetivo es evaluar condiciones multivariantes, relaciona la magnitud de la diferencia entre muestras consecutivas de un proceso de muestreo aperiódico de las señales con el error introducido en la señal Booleana resultante en términos de precisión y exhaustividad. Lo cual demuestra la primera hipótesis de esta tesis doctoral.
- El modelo D2R-TED incluye métricas que permiten un análisis de los resultados tanto en términos de tráfico evitado como de errores introducidos en la detección. Estas métricas aportan un enfoque novedoso al permitir analizar este comportamiento en clave temporal, centrado en los instantes de transición y permitiendo definir la tolerancia al desfase. Este enfoque enriquece la interpretación del resultado de aplicar el modelo en comparación con el clásico análisis muestra a muestra.
- La formulación de D2R-TED basada en dominios matemáticos no definidos lo convierte en una técnica de propósito general dentro de sus limitaciones, pudiendo ser aplicado en cualquier escenario en el que el cumplimiento de una condición Booleana multivariante dependa de funciones continuas.
- Se ha propuesto el algoritmo DRACE, que consigue adaptar dinámicamente el criterio basado en magnitud del proceso de muestreo de señales involucradas en la evaluación de condiciones multivariantes para conseguir reducir el tráfico de la red evitando introducir errores en el resultado, lo cual demuestra la segunda hipótesis de esta tesis doctoral.

- El amplio conjunto de señales aleatorias pseudo–realistas empleado en la experimentación del algoritmo DRACE, así como los prometedores resultados obtenidos, permiten concluir que el algoritmo propuesto tiene el potencial de optimizar considerablemente el uso de la red en escenarios de evaluación de condiciones multivariantes.

Concluyendo, el resultado de las experimentaciones llevadas a cabo ilustran el potencial de la metodología propuesta, cuyo algoritmo es capaz de reducir el tráfico de la red un 40% en promedio sin cometer errores en un 99% de las pruebas. Estos resultados demuestran las hipótesis de esta tesis que, vertebando el contenido de los artículos con una serie de objetivos metodológicos, plantean la posibilidad de modelar la reducción de datos, medir la afectación de esta reducción en la precisión de la salida, y la utilización de este conocimiento en un procedimiento que consiga los resultados obtenidos.

Finalmente, la evaluación de condiciones multivariantes en redes de sensores pueden implementarse mediante la centralización de los flujos de información sin recurrir a procesos de muestreo y envío continuo de datos de las señales involucradas. La reducción de información en estos flujos de datos implica un deterioro de la calidad de la detección en términos temporales, que pueden ser medidos en base a la precisión y exhaustividad de los cambios de estado. Esta relación puede ser modelada para establecer métodos de predicción que permitan administrar la reducción de información limitando los errores en el resultado del sistema. El modelo D2R–TED y el algoritmo DRACE confirman este hecho mediante los resultados obtenidos experimentalmente, demostrando así las hipótesis de esta tesis doctoral.

En este capítulo se aportan visiones de trabajo futuro y potenciales mejoras del trabajo descrito en esta tesis doctoral.

En cuanto al generador de señales aleatorias pseudo-realistas, SysGPr, una limitación importante es la incapacidad de generar datos afectados por cierto patrón periódico o gradiente global. Este aspecto puede ser abordado mediante la introducción de un sesgo probabilístico en el proceso iterativo de generación de muestras consecutivas. Por otra parte, un aspecto interesante a considerar de la generación de señales es plantear la posibilidad de reproducir patrones parecidos a modelos suministrados. Esta aplicación requiere replantear los algoritmos de generación para incorporar un análisis previo de la información de los modelos, cabiendo plantear el uso de redes neuronales adversarias u otros modelos de aprendizaje automático. Esta potencial aplicación abre la puerta a desarrollar sensores virtuales que suministren datos ilimitados para simulación de procesos con señales de parámetros concretos del mundo real.

Con respecto a D2R-TED, la información aportada por las métricas propuestas sugiere la posibilidad de analizar el efecto del modelo en un sistema mediante una familia de curvas $1 - e^{-\alpha(x+\beta)}$. El análisis de estas curvas puede dar lugar a un conocimiento profundo del comportamiento de las transiciones de la señal del sistema, abriendo la puerta a posibles métodos analíticos que permitan trabajar con una variable δ continua, en lugar de un conjunto acotado y predefinido (Δ). Por otra parte, la función de reducción tiene una gran influencia en el resultado, tanto en términos de reducción de tráfico como de alteración del patrón de transiciones, por lo que sería interesante analizar los márgenes de optimización que se pueden alcanzar mediante la definición de estas funciones.

En relación a DRACE, el algoritmo utiliza un método predictivo para reconfigurar dinámicamente el parámetro δ de las fuentes de datos. Si se modelan apropiadamente los efectos de la variación de la definición estática de las estructuras de datos que lo soportan (longitud de las colas, factor de la desviación típica, etc.) se podría plantear cómo determinar la configuración óptima en cada fuente de datos, evitando así posibles problemas en escenarios con sensores heterogéneos. Además, el método predictivo propuesto se basa en la desviación típica de los últimos datos recibidos de cada fuente de información. Si bien la propuesta es válida y ofrece buenos resultados, sería interesante explorar otras propuestas de técnicas predictivas con el objetivo de mejorar el desempeño del algoritmo. Por otra parte, DRACE parte del subconjunto de nodos que han enviado para reducir o aumentar los parámetros δ en un proceso iterativo en el que cada iteración determina el nodo candidato idóneo para este proceso. Esta lógica sigue una heurística de selección de candidato basada en los resultados del método predictivo sobre los últimos datos de cada nodo, pero merece un análisis en profundidad que contemple otros criterios que puedan mejorar el algoritmo.

Finalmente, las técnicas desarrolladas en esta tesis tienen cabida en aplicaciones que requieran detectar con precisión condiciones que dependan de variables distribuidas. La adaptación del volumen de tráfico generado a la

inminencia de un cambio de estado en la condición es una característica interesante que habilita un concepto de red de detección novedoso para el que se propone el nombre de *red latente*. Este tipo de red tiene como objetivo detectar condiciones inusuales que, generalmente, pueden suceder en cualquier punto de un área geográfica densamente monitorizada. La *red latente* tiene la capacidad de silenciar sus nodos durante los largos periodos de tiempo en los que predomina una escasa probabilidad de cumplimiento de la condición, incrementando progresivamente la actividad de los sensores de aquellas áreas en las que la condición se hace más probable. Las redes latentes pueden ser un concepto relevante en aplicaciones como detección de incendios o alarmas de invasión de perímetros extensos.

As discussed in Chapter 2, trends in the development of distributed computing focus on resource optimization and context-sensitive design. This thesis contributes to the optimization of the traffic in sensor networks that evaluate the state of multivariate conditions, and whose variables are distributed in the network. The proposals are obtained through the synthesis of adaptive techniques, which exploit the logical context of information and variations in signal behaviour.

The developed work is presented by means of three publications presented in Chapter ii. The first one, with a methodological objective, presents a tool for generating large amounts of data, suitable for simulating and validating the research proposals developed in the other two publications. The second article deals with the mathematical fundamentals of traffic reduction in the application scenario under study, providing the necessary metrics to measure the avoided volume of traffic and the quality of the system response. Based on these mathematical foundations, the third publication presents an algorithm that maximizes traffic reduction and minimizes condition evaluation errors simultaneously.

The following conclusions can be drawn from the work of this doctoral thesis:

- The SysGpr software, developed as a collateral and strictly methodological objective, has proved to be a useful and easy to use tool for experimentation, as it allows for the generation of large random datasets with the variability of real signals.
- Model D2R-TED has been proposed. In sensor networks aimed at evaluating multivariate conditions, this model relates the magnitude of the difference between consecutive samples of an aperiodic sampling process of the signals, with the error introduced in the resulting Boolean signal in terms of precision and recall. This demonstrates the first hypothesis of this doctoral thesis.
- The D2R-TED model includes metrics that allow an analysis of the results in terms of both traffic avoided and errors introduced in the detection. These metrics provide a novel approach to analyze this behavior in terms of time. This is achieved by basing the metrics on the transition instants of the resulting Boolean signal, and introducing a delay tolerance parameter. This approach enriches the interpretation of the results, compared to the classical sample-by-sample analysis.
- The statement of D2R-TED is based on undefined mathematical domains, which makes it a general-purpose technique within its limitations. Thus, D2R-TED can be applied in any scenario where the fulfillment of a multivariate Boolean condition depends on continuous variables.

- The DRACE algorithm has been proposed, which dynamically adapts the criterion based on the magnitude of the signal sampling process involved in the evaluation of multivariate conditions in order to reduce network traffic without introducing errors in the result, which demonstrates the second hypothesis of this doctoral thesis.
- The large set of pseudo-realistic random signals used in the experimentation of the DRACE algorithm, as well as the promising obtained results, allow to conclude that the proposed algorithm has the potential to optimize significantly network resources in multivariate condition evaluation scenarios.

In conclusion, the experiments illustrate the potential of this methodology, whose algorithm is able to reduce network traffic by an average of 40% without errors in 99% of the tests. These results demonstrate the hypotheses of this thesis which, structuring the papers with a series of methodological objectives, propose the possibility of modelling data reduction, measuring the effect of this reduction on the precision of the output, and the use of this knowledge in a procedure that achieves the results obtained.

Finally, the evaluation of multivariate conditions in sensor networks can be implemented through the centralization of information flows without using continuous sampling and sending processes with the data of the signals involved. The reduction of information in these data flows implies a decrease in the quality of detection in temporal terms, which can be measured based on the precision and recall of the condition state changes. This correlation can be modeled to establish predictive methods to manage information reduction by limiting errors in the system result. The D2R-TED model and the DRACE algorithm confirm this fact through the results obtained experimentally, thus demonstrating the hypotheses of this doctoral thesis.

BIBLIOGRAFÍA

- [1] IEEE 802.15 WPAN Task Group 4. <http://www.ieee802.org/15/pub/TG4.html>. (Citado en las páginas 18 y 20.)
- [2] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggle: *Towards a better understanding of context and context-awareness*. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 304–307, 1999, ISBN 3540665501. (Citado en la página 12.)
- [3] Norman Abramson: *THE ALOHA SYSTEM—Another alternative for computer communications*. In *Proceedings of the November 17-19, 1970, fall joint computer conference on - AFIPS '70 (Fall)*, pages 281–285, 1970. (Citado en la página 6.)
- [4] M. Mehdi Afsar and Mohammad H. Tayarani-N: *Clustering in sensor networks: A literature survey*. *Journal of Network and Computer Applications*, 46:198–226, nov 2014. (Citado en la página 18.)
- [5] Christos Antonopoulos, Sofia Maria Dima, and Stavros Koubias: *Event Identification in Wireless Sensor Networks*. In *Components and Services for IoT Platforms*, pages 187–210. Springer International Publishing, Cham, 2017. (Citado en la página 18.)
- [6] Malte Appeltauer, Robert Hirschfeld, Michael Haupt, Jens Lincke, and Michael Perscheid: *A comparison of context-oriented programming languages*. In *International Workshop on Context-Oriented Programming, COP '09, 2009*, ISBN 9781605585383. (Citado en la página 13.)
- [7] K. Ashton: *That Internet of Things Thing*. *RFID Journal*, 50:97–114, 2009. (Citado en la página 14.)
- [8] K.J. Astrom and B.M. Bernhardsson: *Comparison of Riemann and Lebesgue sampling for first order stochastic systems*. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002. (Citado en la página 21.)
- [9] Luigi Atzori, Antonio Iera, and Giacomo Morabito: *The Internet of Things: A survey*. *Computer Networks*, 54(10):2787–2805, 2010, ISSN 13891286. (Citado en la página 14.)
- [10] Majid Bahrepour, Nirvana Meratnia, Mannes Poel, Zahra Taghikhaki, and Paul J.M. Havinga: *Distributed event detection in wireless sensor networks for disaster management*. In *2010 International Conference on Intelligent Networking and Collaborative Systems*. IEEE, nov 2010. (Citado en la página 20.)
- [11] Tim Berners-Lee, Robert Cailliau, Jean Francois Groff, and Bernd Pollermann: *World-wide web: The information universe*, 1992. ISSN 10662243. (Citado en la página 5.)
- [12] R.a Bi, H.a Gao, and Y.a b Li: *Probabilistic threshold based monitoring using sensor networks*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8491:246–255, 2014. (Citado en la página 20.)
- [13] F Bonomi: *Connected Vehicles, the Internet of Things, and Fog Computing*. In *The eighth ACM international workshop on vehicular Inter-NETworking (VANET 2011)*, 2011. (Citado en las páginas 16 y 17.)

- [14] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli: *Fog computing and its role in the internet of things*. In *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, 2012. (Citado en las páginas 16 y 17.)
- [15] Patrick Brézillon and Avelino J. Gonzalez (editors): *Context in Computing: A Cross-Disciplinary Approach for Modeling the Real World*. Springer, 2014. (Citado en la página 13.)
- [16] Martin G. Brown: *Supporting User Mobility*. In *Mobile Communications*, pages 69–77. 1996. (Citado en la página 12.)
- [17] Peter J. Brown, John D. Bovey, and Xian Chen: *Context-aware applications: From the laboratory to the marketplace*. IEEE Personal Communications, 4(5):58–64, 1997, ISSN 10709916. (Citado en la página 12.)
- [18] J Bruijn: *Using Ontologies: Enabling Knowledge Sharing and Reuse on the Semantic Web*. Technical report, 2003. (Citado en la página 13.)
- [19] Rafael Capilla, Oscar Ortiz, and Mike Hinchey: *Context variability for context-aware systems*. Computer, 47(2):85–87, 2014, ISSN 00189162. (Citado en la página 13.)
- [20] Carlos Cetina, Pau Giner, Joan Fons, and Vicente Pelechano: *Prototyping Dynamic Software Product Lines to evaluate run-time reconfigurations*. In *Science of Computer Programming*, 2013. (Citado en la página 13.)
- [21] D. Chen, M. Nixon, and A. Mok: *Wirelesshart: real-time mesh network for industrial automation*. Springer Publishing Company, Incorporated, 2010. (Citado en las páginas 18 y 20.)
- [22] Penny Chen: *Standard for an architectural framework for the Internet of Things (IoT) - IEEE P2413*. In *Proceedings of the International Instrumentation Symposium*, 2016. (Citado en la página 14.)
- [23] Pascal Costanza and Robert Hirschfeld: *Language constructs for context-oriented programming: An overview of ContextL*. In *DLS 2005 - Symposium Proceedings: Dynamic Languages Symposium 2005*, 2005, ISBN 1595932836. (Citado en la página 13.)
- [24] Almir Davis: *A Survey of Wireless Sensor Network Architectures*. International Journal of Computer Science & Engineering Survey, 3(6):1–22, 2012, ISSN 09763252. (Citado en la página 18.)
- [25] Brecht Desmet, Jorge Vallejos, Pascal Costanza, Wolfgang De Meuter, and Theo D'Hondt: *Context-oriented domain analysis*. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, ISBN 9783540742548. (Citado en la página 13.)
- [26] AK Dey: *Context-aware Computing*. In John Krumm (editor): *Ubiquitous Computing Fundamentals*, chapter 8, pages 321–352. Taylor and Francis Group, LLC, New York, 2010. (Citado en la página 12.)
- [27] Anind K Dey: *Context-aware computing: The CyberDesk project*. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, 1998. (Citado en la página 12.)
- [28] John Dille, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl: *Globally distributed content delivery*. IEEE Internet Computing, 6(5):50–58, 2002, ISSN 10897801. (Citado en la página 17.)

- [29] Sinem Coleri Ergen: *ZigBee/IEEE 802.15. 4 Summary*. UC Berkeley, September, 2004. (Citado en las páginas 18 y 20.)
- [30] Massimo Fornasier and Holger Rauhut: *Compressive sensing*. In *Handbook of Mathematical Methods in Imaging: Volume 1, Second Edition*. 2015, ISBN 9781493907908. (Citado en la página 21.)
- [31] David Franklin and Joshua Flachsbart: *All Gadget and No Representation Makes Jack a Dull Environment Sensing*. In *Proceedings of AAAI 1998 Spring Symposium on Intelligent Environments (SprSym'98)*, 1998. (Citado en la página 12.)
- [32] Manish J. Gajjar: *Context-aware computing*. In *Mobile Sensors and Context-Aware Computing*, pages 17–35. 2017, ISBN 9780128016602. (Citado en la página 13.)
- [33] Nadia Gámez, Javier Cubo, Lidia Fuentes, and Ernesto Pimentel: *Configuring a context-aware middleware for wireless sensor networks*. *Sensors*, 12(7):8544–8570, 2012, ISSN 14248220. (Citado en la página 13.)
- [34] Neil Gershenfeld, Raffi Krikorian, and Danny Cohen: *The Internet of Things*. *Scientific American*, 291(4):76–81, 2004, ISSN 0036-8733. (Citado en la página 14.)
- [35] Fritz R. Gfeller: *Wireless In-House Data Communication via Diffuse Infrared Radiation*. *Proceedings of the IEEE*, 67(11):1474–1486, 1979, ISSN 15582256. (Citado en la página 6.)
- [36] Sebastián González, Kim Mens, Marius Colacioiu, and Walter Cazzola: *Context traits: Dynamic behaviour adaptation through run-time trait recombination*. In *AOSD 2013 - Proceedings of the 2013 ACM on Aspect-Oriented Software Development*, 2013, ISBN 9781450317665. (Citado en la página 13.)
- [37] Adam Greenfield: *Everyware: The dawning age of ubiquitous computing*. New Riders, Boston, 2006. (Citado en la página 4.)
- [38] Jayant Gupchup, Al Burns, Andreas Terzis, and Alex Szalay: *Model-based event detection in wireless sensor networks*. In *Workshop on Data Sharing and Interoperability on the World-Wide Sensor Web (DSI)*, 2007. (Citado en la página 19.)
- [39] Heinrich Hertz: *Electric waves : being researches on the propagation of electric action with finite velocity through space*. Macmillan, London, 1893. (Citado en la página 6.)
- [40] Robert Hirschfeld, Pascal Costanza, and Oscar Nierstrasz: *Context-oriented programming*. *Journal of Object Technology*, 7(3):125, 2008, ISSN 16601769. (Citado en la página 13.)
- [41] Jonathan Hui and Arch Rock Corporation: *6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture*, 2009. ISSN 1749-6632. (Citado en las páginas 14, 18 y 20.)
- [42] Alexander Ihler, Jon Hutchins, and Padhraic Smyth: *Adaptive event detection with time - Varying poisson processes*. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, ISBN 1595933395. (Citado en la página 19.)
- [43] IIC: *The Industrial Internet of Things Volume G1: Reference Architecture*. Technical report, 2019. (Citado en la página 17.)

- [44] ITU: *ITU Internet Reports 2005: The Internet of Things*. Technical report, 2005, ISBN 978-1-4398-9299-2. (Citado en las páginas 16 y 17.)
- [45] Krasimira Kapitanova, Sang H. Son, and Kyoung Don Kang: *Event detection in wireless sensor networks - Can fuzzy values be accurate?* In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2010, ISBN 3642179932. (Citado en la página 20.)
- [46] Krasimira Kapitanova, Sang H. Son, and Kyoung Don Kang: *Using fuzzy logic for robust event detection in wireless sensor networks*. *Ad Hoc Networks*, 10(4):709–722, 2012. (Citado en la página 20.)
- [47] M C Kerman, Wei Jiang, Alan Blumberg, and S E Buttrey: *A comparison of robust metamodels for the uncertainty quantification (UQ) of New York Harbor oceanographic data*. *Journal of Operational Oceanography*, 1:3–13, 2008. (Citado en la página 19.)
- [48] Mitchell C Kerman, Wei Jiang, Alan F Blumberg, and Samuel E Buttrey: *Event detection challenges, methods, and applications in natural and artificial systems*. *Proc. 14th Int. Command Control Res. Technol. Symp.*, pages 1–19, 2009. (Citado en la página 19.)
- [49] Ja Won Ko and Yoon Hwa Choi: *A grid-based distributed event detection scheme for wireless sensor networks*. *Sensors*, 11(11):10048–10062, 2011, ISSN 14248220. (Citado en la página 20.)
- [50] Yann Aël Le Borgne, Sylvain Raybaud, and Gianluca Bontempi: *Distributed principal component analysis for wireless sensor networks*. *Sensors*, 8(8):4821–4850, 2008, ISSN 1424-8220. (Citado en la página 21.)
- [51] Wenfeng Li, Junrong Bao, and Weiming Shen: *Collaborative wireless sensor networks: A survey*. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2011. (Citado en la página 18.)
- [52] J C R Licklider & Taylor, R.W: *The computer as a communication device*. *Science & Technology*, 1968. (Citado en la página 5.)
- [53] Dragos Margineantu, Weng Keen Wong, and Denver Dash: *Machine learning algorithms for event detection*. *Machine Learning*, 79(3):257–259, apr 2010. (Citado en la página 20.)
- [54] Kim Mens, Rafael Capilla, Nicolás Cardozo, and Bruno Dumas: *A taxonomy of context-aware software variability approaches*. *MODULARITY Companion 2016 - Companion Proceedings of the 15th International Conference on Modularity*, (2):119–124, 2016. (Citado en la página 13.)
- [55] Kevin L Mills: *A brief survey of self-organization in wireless sensor networks*. *Wireless Communications and Mobile Computing*, 7(7):823–834, 2007. (Citado en la página 18.)
- [56] M. Miskowicz: *The event-triggered sampling optimization criterion for distributed networked monitoring and control systems*. In *IEEE International Conference on Industrial Technology*, 2003. (Citado en la página 21.)
- [57] Marek Miskowicz: *Send-on-delta concept: An event-based data reporting strategy*. *Sensors*, 6(1):49–63, 2006, ISSN 14248220. (Citado en la página 33.)
- [58] Marek Miśkowicz (editor): *Event-based control and signal processing*. CRC Press, Boca Raton, 2015. (Citado en la página 21.)

- [59] G E Moore: *Cramming More Components Onto Integrated Circuits*, Electronics, April 19, 1965. Electronics, 1965, ISSN 0018-9219. (Citado en la página 5.)
- [60] Marco Mori, Fei Li, Christoph Dorn, Paola Inverardi, and Schahram Dustdar: *Leveraging state-based user preferences in context-aware reconfigurations for self-adaptive systems*. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, ISBN 9783642246890. (Citado en la página 13.)
- [61] Luca Mottola: *Programming Wireless Sensor Networks : Fundamental Concepts and State of the Art*. ACM Computing Surveys, 5:1–51, 2011, ISSN 03600300. (Citado en la página 18.)
- [62] Azadeh Mousavi, Matt Duckham, Ramamohanarao Kotagiri, and Abbas Rajabifard: *Spatio-temporal event detection using probabilistic graphical models (PGMs)*. In *Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, 2013. (Citado en la página 19.)
- [63] Aitor Murguzur, Rafael Capilla, Salvador Trujillo, Óscar Ortiz, and Roberto E. Lopez-Herrejon: *Context variability modeling for runtime configuration of service-based dynamic software product lines*. In *ACM International Conference Proceeding Series*, 2014, ISBN 9781450327398. (Citado en la página 13.)
- [64] Cory Myers, A V Oppenheim, R Davis, and Webster Dove: *Knowledge based speech analysis and enhancement*. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, pages 162–165, 1984. (Citado en la página 18.)
- [65] A. Nasridinov, S. Y. Ihm, Y. S. Jeong, and Y. H. Park: *Event detection in wireless sensor networks: Survey and challenges*. Lect. Notes Electr. Eng. (Mobile, Ubiquitous, Intell. Comput. Music., 274:585–591, 2014, ISSN 18761119 18761100. (Citado en la página 19.)
- [66] Pinar Ozturk and Agnar Aamodt: *Towards a model of context for case-based diagnostic problem solving*. Proceedings of interdisciplinary conference on modeling and using context (Context97), 1997. (Citado en la página 13.)
- [67] Sidra Rashid, Usman Akram, Saad Qaisar, Shoab Ahmed Khan, and Emad Felemban: *Wireless sensor network for distributed event detection based on machine learning*. In *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*. IEEE, sep 2014. (Citado en la página 20.)
- [68] P. P. Ray: *A survey on Internet of Things architectures*. Journal of King Saud University - Computer and Information Sciences, 30(3):291–319, 2018. (Citado en las páginas 15 y 17.)
- [69] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, and Siobhán Cla: *Middleware for internet of things: A survey*. IEEE Internet of Things Journal, 3(1):70–95, 2016, ISSN 23274662. (Citado en la página 15.)
- [70] Adi Mallikarjuna Reddy V., A. V.U. Phani Kumar, D. Janakiram, and G. Ashok Kumar: *Wireless sensor network operating systems: A survey*. International Journal of Sensor Networks, 5(4), 2009, ISSN 17481279. (Citado en la página 18.)

- [71] Tom Rodden, Keith Cheverst, Nigel Davies, and Alan Dix: *Exploiting context in HCI design for mobile systems*. Workshop on Human Computer Interaction with Mobile Devices, 1998. (Citado en la página 12.)
- [72] Bill Schilit and Marvin Theimer: *Disseminating Active Map Information to Mobile Hosts*. IEEE Network, 8(5):22–32, 1994. (Citado en la página 12.)
- [73] Chana R. Schoenberger and Bruce Uphin: *The Internet of Things*. Forbes Magazine, 2002. (Citado en la página 14.)
- [74] Syed Afsar Shah, Babar Nazir, and Imran Ali Khan: *Congestion control algorithms in wireless sensor networks: Trends and opportunities*. In *J. King Saud Univ. - Comput. Inf. Sci.*, volume 29, pages 236–245. 2017. (Citado en la página 21.)
- [75] Silicon Labs: *UG103.11: Thread Fundamentals, Rev.0.7*. Technical report, 2017. <https://www.silabs.com/documents/public/user-guides/ug103-11-appdevfundamentals-thread.pdf>. (Citado en las páginas 18 y 20.)
- [76] Hugh Southall: *Distributed Sensor Networks. Final report to the Defense Advanced Research Projects Agency*. Technical report, Massachusetts Institute Of Technology (MIT) Lincoln Laboratory, Lexington, Massachusetts, 1986. (Citado en las páginas 17 y 18.)
- [77] Horst Stormer: *21 ideas for the 21st century*. Business Week, pages 78–167, 1999. (Citado en las páginas 18 y 20.)
- [78] Lu Tan and Neng Wang: *Future Internet: The Internet of Things*. In *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, volume 5, pages 376–380. IEEE, 2010, ISBN 9781424465408. (Citado en las páginas 16 y 17.)
- [79] Gayathri Vijay, Elyes Bdira, and Mohamed Ibnkahla: *Cognitive approaches in wireless sensor networks: A survey*. In *2010 25th Biennial Symposium on Communications, QBSC 2010*, 2010, ISBN 9781424457090. (Citado en la página 18.)
- [80] Mm Wang, Jn Cao, Jing Li, and Sk Dasi: *Middleware for wireless sensor networks: A survey*. Journal of Computer Science and Technology, 23(2006):305–326, 2008. (Citado en la página 18.)
- [81] Roy Want: *An Introduction to Ubiquitous Computing*. In John Krumm (editor): *Ubiquitous Computing Fundamentals*, chapter 1, pages 1–35. Taylor and Francis Group, LLC, New York, 2010. (Citado en la página 11.)
- [82] Andy Ward, Alan Jones, and Andy Hopper: *A new location technique for the active office*. IEEE Personal Communications, 4(5):42–47, 1997, ISSN 10709916. (Citado en la página 12.)
- [83] M. Weiser: *The computer for the 21st Century*. Scientific American, 265:94–104, 1991. (Citado en las páginas 4 y 11.)
- [84] M. Weiser: *The world is not a desktop*. interactions, 1(1):7–8, 1994, ISSN 10725520. (Citado en la página 11.)
- [85] Miao Wu, Ting Jie Lu, Fei Yang Ling, Jing Sun, and Hui Ying Du: *Research on the architecture of Internet of Things*. In *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, volume 5, pages 484–487, 2010, ISBN 9781424465408. (Citado en las páginas 16 y 17.)

- [86] Yihang Yin, Fengzheng Liu, Xiang Zhou, and Quanzhong Li: *An Efficient Data Compression Model Based on Spatial Clustering and Principal Component Analysis in Wireless Sensor Networks*. *Sensors*, 15(8):19443–65, 2015, ISSN 1424-8220. (Citado en la página 21.)
- [87] A. Yousefi, S. Member, A. A. Dibazar, and T. W. Berger: *Intelligent fence intrusion detection system: detection of intentional fence breaching and recognition of fence climbing*. In *2008 IEEE Conference on Technologies for Homeland Security*, pages 620–625, 2008. (Citado en la página 20.)
- [88] Qin Yu, Lyu Jibin, and Lirui Jiang: *An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks*. *International Journal of Distributed Sensor Networks*, 12(1):9653230, jan 2016. (Citado en la página 19.)
- [89] Xian Ming Zhang, Qing Long Han, and Bao Lin Zhang: *An Overview and Deep Investigation on Sampled-Data-Based Event-Triggered Control and Filtering for Networked Systems*. *IEEE Transactions on Industrial Informatics*, 13(1):4–16, 2017. (Citado en la página 21.)
- [90] M Zoumboulakis and G Roussos: *Complex Event Detection in Extremely Resource-Constrained Wireless Sensor Networks*. *Mobile Networks & Applications*, 16(2):194–213, 2011, ISSN 1383-469X. (Citado en la página 20.)