

# Mejoramiento del algoritmo ADR en una red de Internet de las Cosas LoRaWAN usando Aprendizaje de Máquina

Mauricio González-Palacio<sup>1</sup>, Lina María Sepúlveda-Cano<sup>2</sup>,  
Jhon Quiza-Montealegre<sup>1</sup>, Juan D'Amato<sup>3</sup>

[magonzalez@udem.edu.co](mailto:magonzalez@udem.edu.co); [lmsepulveda@udem.edu.co](mailto:lmsepulveda@udem.edu.co);  
[jhquiza@udem.edu.co](mailto:jhquiza@udem.edu.co); [juan.damato@gmail.com](mailto:juan.damato@gmail.com)

<sup>1</sup> Ingeniería de Telecomunicaciones, Universidad de Medellín, 050082, Medellín, Colombia.

<sup>2</sup> Ingeniería de Sistemas, Universidad de Medellín, 050082, Medellín, Colombia.

<sup>3</sup> Instituto Pladema, Universidad Nacional del Centro de la Provincia de Buenos Aires, B7001BBO, Tandil, Argentina.

DOI: [10.17013/risti.39.67-83](https://doi.org/10.17013/risti.39.67-83)

**Resumen:** El Internet de las Cosas es un paradigma habilitador de la Industria 4.0, donde sensores y actuadores se conectan a Internet. El protocolo LoRaWAN (Long Range Area Network) es uno de los más empleados, y es usado para transmitir información a largas distancias con mínimo consumo energético. Este protocolo implementa el esquema *Adaptive Data Rate* para mejorar la energía consumida por nodo, que al ser evaluado a través de simulaciones exhaustivas en Omnet++, ha exhibido posibilidades de mejora en el tiempo de convergencia. El presente trabajo muestra una propuesta para el mejoramiento del algoritmo ADR de tal forma que se optimice el consumo energético en redes LoRaWAN. Dentro de la propuesta se comparan diferentes modelos paramétricos y no paramétricos. Los resultados indican que los métodos basados en Máquinas de Vectores de Soporte y en Redes Neuronales Artificiales presentan la mayor exactitud, con un porcentaje por encima del 90% en las estimaciones.

**Palabras-clave:** Internet de las Cosas; Industria 4.0; consumo energético; LoRaWAN; Aprendizaje de Máquina

## *Improvement of the algorithm ADR in an Internet of Things network LoRaWAN by using Machine Learning*

**Abstract:** The Internet of Things (IoT) is an enabling paradigm for Industry 4.0, where sensors and actuators connect to the Internet. The protocol LoRaWAN (Long Range Area Network) is one of the most used in the IoT, and its primary objective is to transmit sensor information over long distances with minimal energy consumption. This protocol implements Adaptive Data Rate scheme to optimize the energy consumed per node, which, when evaluated through exhaustive simulations in Omnet ++, has exhibited opportunities for improvement in convergence time.

The present work shows machine learning models based on parametric and non-parametric methods based on Support Vector Machines (SVM) and Artificial Neural Networks (ANN). The results indicate that the SVM and ANN methods have a success rate greater than 90% in the estimated parameters.

**Keywords:** Internet of Things; Industry 4.0; energy consumption; LoRaWAN; Machine Learning

## 1. Introducción

Cuatro revoluciones industriales han impactado la economía históricamente: la primera, cuyo impacto fue la invención de la máquina de vapor; la segunda, cuyo impacto fue el uso de la energía eléctrica en un sinnúmero de procesos productivos; y la tercera, cuyo impacto fue la digitalización de datos dentro de la industria, a través de sistemas de automatización, control y supervisión (Lasi, Fettke, Kemper, Feld, & Hoffmann, 2014). La cuarta revolución industrial (o Industria 4.0) permite realizar un análisis inteligente de los datos, para transformar la forma en la cual se ejecutan diversos procesos (Schwab, 2017). Dicha revolución se apropia, entre otros, de técnicas de algoritmia que permiten hacer clasificación y/o regresión de fenómenos. El desarrollo de software se configura como una herramienta imprescindible para habilitar la cuarta revolución industrial (Petrasch & Hentschke, 2016), permitiendo, a través de técnicas de inteligencia artificial, realizar análisis sobre grandes volúmenes de información. El Internet de las Cosas (IoT), a su vez, potencia la cuarta revolución industrial como una fuente de datos sin precedentes para su posterior análisis.

Paradigmas como el *Big Data*, la robótica, el aprendizaje automático, y el Internet de las Cosas son potenciadores de la Industria 4.0 (Schwab, 2017). El *Big Data* se define como el tratamiento de grandes volúmenes de información que no pueden ser procesados por esquemas tradicionales de computación en un tiempo tolerable (Chen, Mao, & Liu, 2014). La robótica potencia la maquinaria y su inclusión en la transformación digital de las plantas productivas. La Inteligencia Artificial (IA) predice y clasifica comportamientos con intervención mínima del ser humano (Lee, Davari, Singh, & Pandhare, 2018). El IoT permite, a través de la dotación de objetos de la cotidianidad con sensores y actuadores, conocer el comportamiento de procesos e incidir sobre ellos (Wollschlaeger, Sauter, & Jasperneite, 2017). El análisis de las características de conectividad, en particular en el consumo energético en redes de Internet de las Cosas, son el interés principal en el presente artículo.

En sitios donde no se puede depender de un operador de red, las redes de baja potencia y área amplia (LPWAN) se ajustan para transferir pequeños paquetes de información. Las tecnologías líderes en el mercado actualmente son: LoRaWAN (Sornin, Luis, Eirich, Kramp, & Hersent, 2015), Sigfox (Lavric, Petrariu, & Popa, 2019) y NB-IoT (Ratasuk, Vejlgard, Mangalvedhe, & Ghosh, 2016). El consumo energético es uno de los factores más críticos en el diseño de una red de IoT, ya que está pensada, en una gran cantidad de casos, para operar con baterías y durante tiempos prolongados. El presente artículo muestra una mejora del algoritmo *ADR* (*Adaptive Data Rate*) (San Cheong, Bergs, Hawinkel, & Famaey, 2017) del protocolo LoRaWAN, que se logra por medio del uso de Máquinas de Vectores de Soporte (SVM) y Redes Neuronales Artificiales (ANN). Tal modelación se logra a través de simulaciones exhaustivas en Omnet ++.

El presente trabajo se organiza como sigue: en la sección 2 se condensa el marco teórico requerido para la lectura expedita del trabajo, enfocado al protocolo LoRaWAN; en la sección 3 se muestra el trabajo previo que se ha adelantado para mejorar el consumo energético en redes LoRaWAN; en la sección 4 se muestra el diseño metodológico y experimental; en la sección 5 se muestra la modelación propuesta; en la sección 6 se muestra la propuesta del algoritmo para el mejoramiento del esquema ADR y su evaluación; finalmente, se presentan el trabajo futuro, las conclusiones y las referencias.

## 2. Marco teórico

Con respecto al componente arquitectónico, las redes LoRaWAN operan usando los siguientes componentes (de Carvalho Silva, Rodrigues, Alberti, Solic, & Aquino, 2017): (i) nodos, que incluyen sensores, sistemas embebidos y radios para transportar los datos recolectados; (ii) *gateways*, que se encargan de recibir información de los nodos y reenviarla a un servidor de red; (iii) servidor de red, el cual provee un servicio para recolectar los mensajes de diversos *gateways* y reenviarlos a demanda a algún servidor de aplicación, y de implementar esquemas como el ADR; y (iv) servidor de aplicación, donde se realizan las labores de despliegue y análisis de datos.

En cuanto a la utilización del espectro, el protocolo LoRaWAN opera en la banda ISM (*Industrial – Scientific – Medical*) que no requiere licencia<sup>1</sup>. En América, la banda usada es 902-928MHz, y en Europa, 862 a 870 MHz (Sornin et al., 2015). Cada región define los canales permitidos para subida y bajada. En cuanto al uso apropiado del espectro, entidades como la FCC (*Federal Communications Commission*) y la ETSI (*European Telecommunications Standards Institute*) se encargan de dar directrices para permitir su uso colectivo.

LoRaWAN usa la modulación *Chirp Spread Spectrum* (CSS) (Reynders & Pollin, 2016). Los canales pueden tener anchos de banda (BW) de 125 kHz, 250 kHz y 500 kHz, dependiendo del plan de frecuencia. Si la frecuencia central es  $f_0$ , una señal conocida como *chirp* se mueve en el intervalo  $(f_0 - BW/2, f_0 + BW/2)$ , durante un tiempo de símbolo  $T_s$ . El número de bits por símbolo se conoce como *Spreading Factor* (SF) y puede variar entre 7 y 12. El espectro se divide en  $2^{SF}$  posibles frecuencias de inicio (*chips*), de tal manera, en un mismo símbolo puede incluir  $2^{SF}$  diferentes números. Para agregar bits de redundancia, es posible usar el *Coding Rate* (CR) que puede tomar valores entre 1 y 4. La rata de *chips* ( $R_c$ ) es homóloga al ancho de banda y se define en la ecuación (1). La rata de símbolos se define en función del ancho de banda y el SF, como se indica en la ecuación (2), así como el tiempo de símbolo  $T_s$ . La rata de bits, *data rate*, se define en la ecuación (3).

$$BW \text{ (Hz)} = R_c \text{ (chips/sec)} \quad (1)$$

$$R_s \text{ (símbolos/sec)} = \frac{BW}{2^{SF}} ; T_s \text{ (sec)} = \frac{2^{SF}}{BW} \quad (2)$$

<sup>1</sup> <https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country.html>

$$R_b(\text{bits/sec}) = SF \frac{BW}{2^{SF}} \frac{4}{4 + CR} \quad (3)$$

De las ecuaciones (2) y (3) se puede deducir que para valores mayores de SF, el tiempo de símbolo también será mayor, permitiendo mayores distancias y mayor consumo. Con respecto a los parámetros de diseño, se definen el *duty cycle* (dc) y el *Time on Air* (ToA). Los *dc* son períodos en los cuales un nodo no transmite ninguna información, después de haber hecho una transmisión (Adelantado et al., 2017). El *dc* puede tomar valores de 0.1%, 1% o 10%, e indica la proporción de tiempo usada para realizar la transmisión. El *ToA* indica el tiempo que tarda un paquete de información en ser enviado. Así, el tiempo que debe esperarse para realizar una nueva transmisión desde un nodo se define en función del *dc* y el *ToA* como se muestra en la ecuación (4).

$$T_{nt} = \frac{(100 - dc)ToA}{dc} \quad (4)$$

Donde  $T_{nt}$  indica el tiempo para efectuar una nueva transmisión. A su vez el *ToA* se define en la ecuación (5).

$$ToA = (4.25 + n_s)T_s \left( 8 + \max \left( \text{ceil} \left( \frac{(8PL - 4SF + 28 + 16CRC - 20H)}{4(SF - 2DE)} \right), 0 \right) \right) \quad (5)$$

Donde  $n_s$  es el número de preámbulos (típicamente 8), *PL* es el tamaño en bytes del *payload*, *CRC* es una variable binaria que indica si se incluye comprobación por redundancia cíclica, *H* indica si el tipo de trama es implícito o explícito y *DE* si se activa la función *low data rate optimization*, el cual se activa para valores de *ToA* mayores a 100 milisegundos (obligatorio para valores de *SF* de 11 y 12). Para determinar los valores óptimos de *SF* y la potencia de transmisión *TP* de un nodo es posible habilitar el algoritmo *ADR*, ejecutado en el servidor de red (Sornin et al., 2015). Dicho servidor se encarga de recolectar los últimos veinte valores de relación de señal a ruido (*SNR*) y los *data rate* usados (por ejemplo,  $SNR = 5$  dB,  $DR = SF12BW125$ ), y procede a tomar el valor máximo. El servidor de red aplica la ecuación objetivo descrita en (6).

$$\text{Min margin}_{corrected} = SNR_{measured} - SNR_{limit} - \text{linkbudget}_{default} \quad (6)$$

Donde  $\text{linkbudget}_{default}$  indica el nivel de seguridad, por encima de la sensibilidad del radio,  $SNR_{limit}$  indica la relación de señal a ruido límite que varía con el *SF* (a mayor *SF*, mejor sensibilidad de recepción del radio),  $SNR_{measured}$  es el valor máximo de la *SNR* medida y  $\text{margin}_{corrected}$  es la variable objetivo de optimización, cuyo valor debe ser mayor a cero. Nótese que, si las distancias son grandes, puede requerirse tener valores de *SF* y *TP* altos. No obstante, después de realizar las mediciones, el valor de  $\text{margin}_{corrected}$  puede estar lejos de cero, indicando un desperdicio en el *TP* y un *ToA* elevado. Así pues, el servidor de red ajusta el *SF* y recalcula el margen. Si incluso el  $\text{margin}_{corrected}$  no es cero, se ajusta *TP* para bajar la potencia de transmisión del radio. Las simulaciones implementadas muestran que el tiempo de convergencia de este algoritmo es susceptible de mejora (Li, Raza, & Khan, 2018).

### 3. Trabajo previo

Los trabajos de la literatura relacionados más representativos se muestran en la Tabla 1. Las aproximaciones de manejo energético en la literatura abordan diferentes perspectivas, la mayoría proponiendo valores para  $SF$  y  $TP$ , pero optimizando otras métricas diferentes a la energía, tales como la capacidad, la confiabilidad y el rendimiento en general y no se enfocan a la minimización de la energía. También se prevén métodos usando la recolección de algunos parámetros como la  $SNR$  y la  $RSSI$  durante un tiempo suficiente que permita estabilizarlos en un ambiente real, por lo cual el tiempo de convergencia puede ser incierto. Finalmente, diferentes propuestas plantean problemas de optimización que consideran la energía como una restricción, esto es, se limitan a fijar un valor máximo, no a encontrar su valor mínimo. En virtud de los hallazgos, se observa la oportunidad de proponer un método que mejore el consumo energético a partir de la aplicación de modelos que lo optimicen.

Autores	Título	Contribución	Oportunidades de mejora
<i>(Slabicki, Preamsankar, &amp; Di Francesco, 2018)</i>	<i>Adaptive Configuration of LoRa Networks for Dense IoT Deployments</i>	Mejora el consumo energético en canales altamente ruidosos, basándose, entre otros, en el conocimiento previo de toda la red ( <i>network-aware approach</i> ).	Modifica el máximo de las muestras $SNR$ recolectadas en el algoritmo ADR por el promedio. Para que esta aproximación opere adecuadamente en un entorno real se requeriría alta concentración de los $SNR$ alrededor del promedio.
<i>(Sandoval, Rodenas-Herraiz, Garcia-Sanchez, &amp; Garcia-Haro, 2020)</i>	<i>Deriving and updating optimal transmission configurations for LoRa Networks</i>	Propone la sintonización de los parámetros de la transmisión sin requerir un presupuesto de enlace por nodo ni modelos de canal de manera previa. También clama mejorar la resistencia a ataques de denegación de servicio.	Utiliza, para proponer el esquema el modelo de consumo energético, un microcontrolador LoPy4, por lo cual puede carecer de generalidad cuando otras plataformas sean usadas. Además, el número de mediciones realizadas (12), a diferentes potencias, luce insuficiente para otorgar validez estadística a los experimentos
<i>(Reynders, Meert, &amp; Pollin, 2017)</i>	<i>Power and spreading factor control in low power wide area networks</i>	Propone la disminución del efecto de colisiones destructivas entre nodos cerca y lejos del <i>gateway</i> , por medio de la variación del $SF$ y el $TP$ , mejorando la tasa de error de paquetes.	El resultado de la disminución (o aumento) de consumo energético es una consecuencia de la minimización de las colisiones. El algoritmo, si bien entrega valores para $SF$ y $TP$ , no está pensado para minimizar el consumo energético.
<i>(Abdelfadeel, Cionca, &amp; Pesch, 2018)</i>	<i>Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN</i>	Propone un método que permite resolver el problema de interferencia entre nodos cercanos y nodos lejanos, a través del control del $SF$ y el $TP$ , por medio del algoritmo FADR	Basa el cálculo de la potencia de transmisión y el $SF$ en el promedio de muestras recolectadas de los $RSSI$ , lo cual implicaría un tiempo de convergencia incierto (no se mide). No se aporta un método para calcular los $TP$ iniciales, lo cual puede redundar en problemas iniciales de conexión entre los nodos y el $GW$ .

Autores	Título	Contribución	Oportunidades de mejora
<i>(Gao, Du, Zhao, Min, &amp; Singhal, 2019)</i>	<i>Towards energy-fairness in LoRa networks</i>	Modela el comportamiento de energía, confiabilidad en la transmisión e interferencia como un problema de optimización. Propone un algoritmo para realizar la asignación óptima de recursos en un ambiente de múltiples gateways.	Asume que la asignación de los SF y TP se puede optimizar usando más de una <i>gateway</i> , lo cual es cierto desde el punto de vista teórico. Sin embargo, en escenarios reales, los <i>gateway</i> , que tienen alimentación por red eléctrica, no pueden ser ubicados de manera arbitraria, por las condiciones topográficas de una región en particular.
<i>(Bouguera, Diouris, Chaillout, Jaouadi, &amp; Andrieux, 2018)</i>	<i>Energy consumption model for sensor nodes based on LoRa and LoRaWAN</i>	Propone un modelo de cuantificación de energía para nodos en LoRa, que permite evaluar el comportamiento de potencia en diferentes configuraciones, cuantificándolas en varios escenarios.	No proporciona un algoritmo para mejorar la eficiencia en consumo energético de los nodos finales en una red LoRa. Esta actividad se propone como trabajo futuro.
<i>(Sandoval, Garcia-Sanchez, &amp; Garcia-Haro, 2019)</i>	<i>Performance optimization of LoRa nodes for the future smart city/industry</i>	Presenta un modelo matemático, basado en cadenas de Markov, que permite estimar numéricamente el rendimiento de los nodos en una red LoRaWAN, por medio de <i>policy-derivation theory</i> .	El problema de optimización formula como función objetivo la maximización del rendimiento de la red, y la energía consumida se incluye como una restricción. De tal manera, la energía no se minimiza, sino que se limita a un valor máximo dentro de una región factible determinada.

Tabla 1 – Resumen de trabajos previos

#### 4. Metodología y diseño experimental

Primero, se propone la simulación de una red LoRaWAN con 1000 nodos, 1 *gateway* y 1 servidor de red. Se escogen 1000 nodos para garantizar significancia en los datos de la simulación. A su vez, se usa 1 *gateway* y un servidor de red dado que ésta es una configuración típica en este tipo de redes. Los valores de *TP* (2-14 dBm) y *SF* (7-12) se inicializan de manera aleatoria, así como la posición de los nodos, en una geometría de 1 km<sup>2</sup> de área. El *gateway* se ubica en el centro de la región. Se considera que cada nodo transmitirá un mensaje de manera aleatoria (distribución exponencial) en un intervalo de 0 a 100 segundos. El tiempo de simulación se fija a un día, ya que se encuentra un balance adecuado entre el tamaño de la base de datos generada y la convergencia del algoritmo *ADR*. El ancho de banda se fija a 125 kHz, el más común para canales de subida en los diferentes planes de frecuencia por región. Con respecto a la herramienta de simulación, se usa Omnet ++ V5.2.1 y el complemento *FLoRa* (Slabicki et al., 2018). Se habilita el algoritmo tradicional *ADR* para escogencia de los parámetros *SF* y *TP*.

Como segundo paso, para la modelación de la base de datos, se tienen en cuenta los argumentos de entrada distancia (D),  $TP_{inicial}$ ,  $SF_{inicial}$ . Se toma la razón entre la

energía consumida por un nodo y la cantidad de paquetes enviados como salida del modelo (EPP), así como el  $TP_{final}$ , el  $SF_{final}$  y el tiempo  $T_c$  que le toma al algoritmo *ADR* converger a dichos valores. Se construye la base de datos a través de las corridas de simulación respectivas.

En el tercer paso, con los valores obtenidos de energía/paquete y TP para los diferentes casos de D,  $TP_{inicial}$  y  $SF_{inicial}$ , se evalúa la modelación a través de regresión lineal múltiple, modelos lineales generalizados (GLM) y SVM.

En cuarto lugar, se evalúan modelos de aprendizaje de máquina supervisados (regresión logística, SVM y ANN) con los mismos argumentos de entrada.

En quinto lugar, se observa el rendimiento de los métodos empleados y se escoge el que exhibe mejores resultados. La aproximación de evaluación divide la base de datos, de manera aleatoria, en 80% de muestras para entrenamiento y 20% para validación. Finalmente, se valida la mejora de energía contrastando el tiempo  $T_c$  para hallar los parámetros TP y SF a través del algoritmo *ADR* y por medio de la aplicación de los modelos propuestos.

## 5. Modelación

Para el análisis de datos se usa el software R Studio v1.2. Se muestran los análisis para las tres métricas de salida: energía/paquete (EPP),  $TP_{final}$  y  $SF_{final}$ . El resumen estadístico de las variables continuas y categóricas se muestra en la Tabla 2.

Variable continua	Mínimo	Cuartil 1	Mediana	Media	Cuartil 3	Máximo
EPP (J)	0.08062	0.12261	0.15706	0.15914	0.19028	0.28664
D (m)	10.63	137.64	194.19	186.2	234.95	336.7
$TP_{final}$ (dBm)	2	5	8	8.24	14	14
$TP_{inicial}$ (dBm)	2	5	8	7.94	11	14
Variable categórica	SF = 7	SF = 8	SF = 9	SF = 10	SF = 11	SF = 12
$SF_{final}$	37	45	47	44	26	801
$SF_{inicial}$	171	152	156	175	175	171

Tabla 2 – Resumen estadístico de variables continuas y categóricas

### 5.1. Energía por paquete transmitido (EPP)

#### Regresión lineal múltiple

Se evalúa la regresión lineal múltiple siguiendo el modelo en la ecuación (7).

$$EPP = \hat{\beta}_0 + \hat{\beta}_1 D + \hat{\beta}_2 TP_{inicial} + \sum_{j=7}^{12} \hat{\beta}_{3j} SF_{inicial_j} + \epsilon \quad (7)$$

Donde es  $\hat{\beta}_0$  es el intercepto,  $\hat{\beta}_1$  es el estimador para la distancia D,  $\hat{\beta}_2$  es el estimador para TP<sub>inicial</sub>,  $\hat{\beta}_{3j}$  ( $j=7,\dots,12$ ) son los estimadores para las variables binarias  $SF_{inicialj}$ , y  $\epsilon$  es el error del modelo. El método de estimación de los coeficientes usado es LSE (*Least Square Estimator*) con un valor de significación de 5%. Los estimadores se muestran en la Tabla 3.

Coef.	Estimado	p-valor	Coef.	Estimado	p-valor
$\hat{\beta}_0$	$6.413 \times 10^{-2}$	$< 2 \times 10^{-16}$	$\hat{\beta}_{3,9}$	$2.084 \times 10^{-2}$	$< 2 \times 10^{-16}$
$\hat{\beta}_1$	$1.504 \times 10^{-4}$	$< 2 \times 10^{-16}$	$\hat{\beta}_{3,10}$	$5.611 \times 10^{-2}$	$< 2 \times 10^{-16}$
$\hat{\beta}_2$	$2.359 \times 10^{-3}$	$< 2 \times 10^{-16}$	$\hat{\beta}_{3,11}$	$8.466 \times 10^{-2}$	$< 2 \times 10^{-16}$
$\hat{\beta}_{3,8}$	$1.056 \times 10^{-2}$	0.000177	$\hat{\beta}_{3,12}$	$1.054 \times 10^{-1}$	$< 2 \times 10^{-16}$

Tabla 3 – Predictores estimados en el modelo de regresión lineal múltiple

Los p-valores indican que todos los predictores son significativos. El modelo entrega un R<sup>2</sup> múltiple de 0.7686 y un R<sup>2</sup> ajustado de 0.7667, que indica una correlación lineal fuerte entre las variables predictoras y la variable dependiente EPP. En la Figura 1 se muestra el comportamiento de algunos de los puntos reales vs los predichos por el modelo. Se omite el conjunto completo para garantizar la legibilidad en la lectura. En orden de asegurar la validez estadística del modelo, se implementa el análisis de varianza, encontrando p-valores del orden de  $2 \times 10^{-16}$ , mostrando que, en efecto, todas las variables son significativas. Adicionalmente, deben garantizarse los supuestos de normalidad, independencia y homocedasticidad de los residuales. En el primer caso se aplica la prueba de Kolmogorov Smirnov fijando un nivel de significación de 5%, y entregando un valor p de 5.793%. En el caso de independencia, se aplica la prueba de Durbin Watson, entregando un  $DW=1.9907$ , indicando independencia entre las variables predictoras. Finalmente, para garantizar homocedasticidad, se aplica la prueba de Breusch Pagan, obteniendo un valor  $p < 2 \times 10^{-16}$ , indicando que los residuales son heterocedásticos. En tal virtud, el modelo pierde eficiencia en sus estimadores de mínimos cuadrados, aunque sigue siendo insesgado ( $E(X)=X$ ).

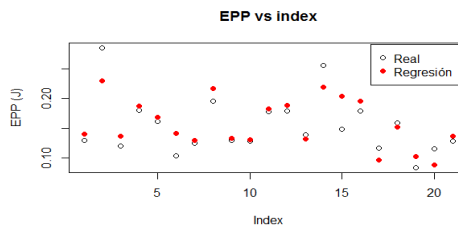


Figura 1 – Comparación entre EPP real y estimado usando el modelo RLM

### Modelos lineales generalizados

Dado que se tiene heterocedasticidad, se recomienda utilizar un modelo lineal generalizado con diferentes funciones de enlace: Gamma y gaussiana inversa.



Las funciones de enlace binomial y Poisson no se consideran por el tipo de proceso que se está modelando. Los coeficientes de los modelos se estimarán utilizando el criterio de máxima verosimilitud (MLE). La calidad de los modelos se evaluará utilizando el índice AIC y la calidad de las predicciones usando el  $R^2$ . Con la función Gamma se logra un AIC de -3918, un  $R^2$  de 0.71 y un RMS de 0.69; y con la gaussiana inversa se logra un AIC de 3828, un  $R^2$  de 0.74 y un RMS de 6.7. Si bien se observa una relación a través del  $R^2$ , usando ambos modelos, también se observa que los errores cuadráticos medios están en órdenes de magnitud mucho mayores que los de los datos. Esta lectura indica que ambos modelos fallan en las regresiones. Si se grafica el EPP del modelo vs EPP teórico (ambos del subconjunto de pruebas), se esperaría una tendencia lineal positiva; no obstante, se obtiene lo graficado en la Figura 2. Por tal motivo, no se publican los valores del modelo, ya que no hay un ajuste satisfactorio.

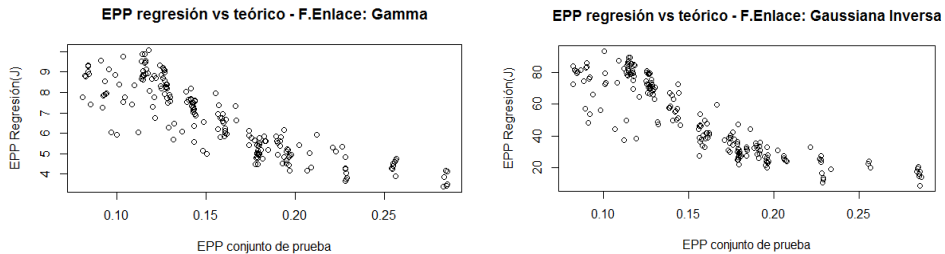


Figura 2 – EPP calculado vs. teórico usando funciones de enlace Gamma y gaussiana inversa

*Máquinas de vectores de soporte (SVM)*

Se procede a aplicar regresión a través de SVM (Noble, 2006). En el caso de clasificación con SVM, se busca el hiperplano que separe dos clases con el máximo margen entre ambas, por lo cual, encontrar el modelo se reduce a un problema de optimización, donde deben hallarse los coeficientes  $w$  que definen dicho hiperplano, como se muestra en la ecuación (8).

$$\text{Min } Z = \frac{1}{2} \|w\|^2 \tag{8}$$

$$\text{Sueto a: } |y_i - w_i x_i| \leq \epsilon,$$

donde  $y_i$  corresponde a las características objetivo,  $w_i$  a los pesos y  $x_i$  a las características de entradas. En el caso de regresión, se agrega a la ecuación (8) un conjunto de variables de holgura, y así, para cada valor que cae por fuera de  $\epsilon$ , se registra su desviación  $\xi$ , de tal manera que se desea que ésta sea mínima. Se introduce el hiper parámetro costo (C) para penalizar dichas desviaciones, el cual es sintonizado a través de validación cruzada. De tal manera, la ecuación (8) se reescribe como se muestra en la ecuación (9).

$$\text{Min } Z = \frac{1}{2} \|w\|^2 + c \sum_{i=0}^n |\xi_i| \tag{9}$$

$$\text{Sujeto a: } |y_i - w_i x_i| \leq \epsilon + |\xi_i|$$

Para el caso de EPP, se toman como características de entrada las mismas usadas en los métodos paramétricos:  $D$ ,  $TP_{\text{inicial}}$  y  $SF_{\text{inicial}}$  y las características objetivo son los valores de EPP. Para realizar las evaluaciones, se usan cuatro tipos diferentes de *kernels*: lineal (RMS=0.025,  $R^2=0.75$ ), polinomial (no converge), sigmoideal (RMS=0.029,  $R^2=0.714$ ) y de base radial (RMS=0.02,  $R^2=0.837$ ). De acuerdo con las métricas presentadas, y a una inspección visual (mostrada en la Figura 3a), el *kernel* de base radial (RBF) es el que se ajusta mejor a los datos. El RBF se considera una medida de similitud modificada (calculada a partir de la distancia euclidiana), y extrapola el espacio de características a infinitas dimensiones. El RBF entre dos muestras se calcula como:  $K(x, x') = \exp(-\gamma \|x - x'\|^2)$ . El hiper parámetro  $\gamma$  se debe sintonizar a través de validación cruzada. Para escoger los hiper parámetros  $C$  y  $\gamma$  se usa el software *R Studio*, librería *e1071*, método *tune*. La Figura 3b muestra la comparación de los errores de clasificación usando diferentes configuraciones de  $C$  y  $\gamma$ , de donde se deduce que, para diferentes conjuntos de entrenamiento, los mejores valores son  $C=10$  y  $\gamma=10$ . Se omite la escritura de los vectores de soporte encontrados, dada su cantidad: 439.

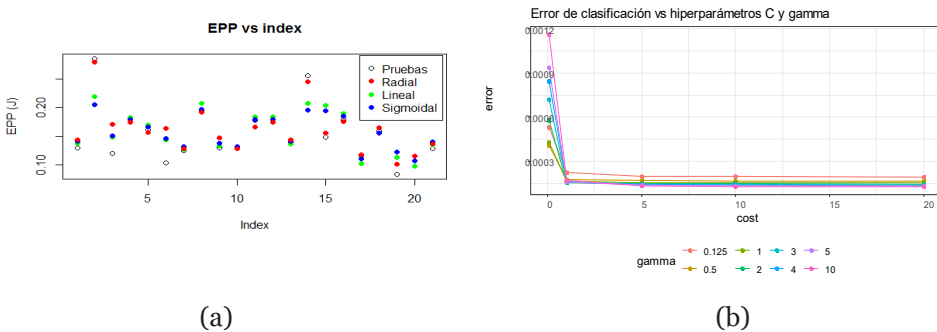


Figura 3 – (a) EPP del conjunto de pruebas vs. regresiones de *kernels* radial, lineal y sigmoideal (b) validación cruzada de parámetros gamma y costo.

### 5.2. Potencia de transmisión final ( $TP_{\text{final}}$ )

Los análisis de RLM, GLM y SVM realizados en este aparte se analizan bajo la misma óptica de la sección 5.1., por lo cual se omitirán los detalles y se procederá a ilustrar los resultados. En todos los casos  $TP_{\text{final}}$  será una función de  $D$ ,  $TP_{\text{inicial}}$  y  $SF_{\text{inicial}}$ .

#### Regresión lineal múltiple

Al implementar el RLM, se encuentra que el predictor  $\hat{\beta}_1$  correspondiente a la distancia no es significativo para el modelo. Tampoco lo es la variable *dummy*  $SF_{\text{final}}$ . El análisis de varianza arroja un p-valor 0.2173 para la distancia, lo cual corrobora la no significancia de dicha variable predictora. El modelo ajustado presenta un  $R^2$  múltiple de 0.9144 y un  $R^2$  ajustado de 0.9136. La distribución de los residuales no es normal (p-valor < 2.2 x

$10^{-16}$  en la prueba de Kolmogorov Smirnov), no hay autocorrelación (prueba de Durbin Watson,  $DW=2.0856$ ,  $p\text{-valor}=0.2246$ ) y no hay homocedasticidad (prueba de Breusch Pagan,  $p\text{-valor} = 1.426 \times 10^{-15}$ ). La violación del supuesto de normalidad en los residuales invalida la aplicación del modelo, por lo cual no se publican los coeficientes.

*Modelos lineales generalizados*

Al aplicar la función de enlace gamma se obtiene un AIC de 3755, un  $R^2$  de 0.91 y un RMS de 9.24. Al aplicar la gaussiana inversa se obtiene un AIC de 4303, un  $R^2$  de 0.91 y un RMS de 9.48. El comportamiento exhibido es similar al observado en la Figura 2. Dado que la tendencia debería ser lineal y no se obtiene este comportamiento, se concluye que GLM no ajusta adecuadamente los datos, por lo cual no se publican los resultados de los modelos.

*Máquinas de vectores de soporte (SVM)*

Se realizan las evaluaciones con cuatro tipos diferentes de *kernels*: lineal (RMS=1.2686,  $R^2=0.91307$ ), polinomial (no converge), sigmoial (RMS=1.27548,  $R^2=0.91159$ ) y RBF (RMS=0.9196,  $R^2=0.955$ ). Dado que RBF es el que logra los mejores resultados con el conjunto de pruebas (Figura 4a), se procede a sintonizar los hiper parámetros C y  $\gamma$ , cuyos resultados se muestran en la Figura 4b. Los mejores valores son  $C=20$  y  $\gamma=0.125$ . Se omite la publicación de los vectores de soporte, debido a su número: 244.

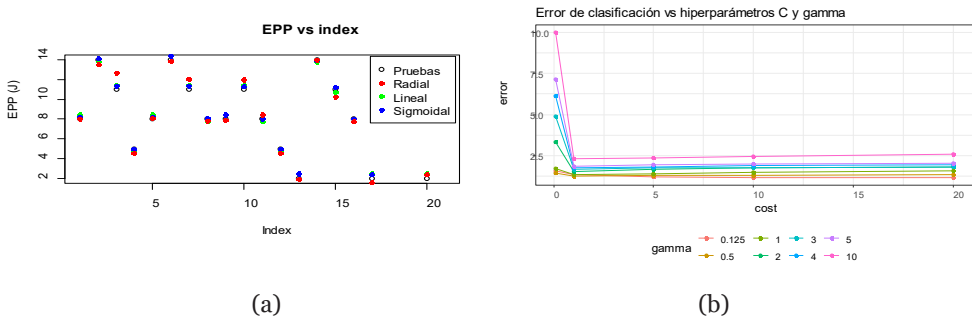


Figura 4 – (a) EPP del conjunto de pruebas vs. regresiones de *kernels* radial, lineal y sigmoial, (b) Sintonización de hiper parámetros del SVM de base radial

**5.3. Número de bits por símbolo (*Spreading Factor*) final ( $SF_{final}$ )**

Dado que el parámetro  $SF_{final}$  es de tipo categórico, el análisis paramétrico se realizará a través de regresión logística múltiple y el no paramétrico con la estrategia de clasificación de SVM y usando ANN. En todos los casos  $TP_{final}$  será una función de D,  $TP_{inicial}$  y  $SF_{inicial}$ .

*Regresión logística multinomial*

Sea  $Y=\{1,2, 3,\dots,n\}$  una variable politómica cuyas probabilidades de ocurrencia son  $p_1, p_2,\dots,p_n$  correspondientemente, las cuales se desean estimar a partir de un conjunto de características de entrada  $x_1,\dots, x_n$ . Se define cada  $p_j$  como se muestra en la ecuación (10).

$$p_j = E(Y_j) = \frac{e^{\sum_{i=0}^m \beta_{ij} x_i}}{1 + \sum_{l=1}^n e^{\sum_{i=0}^m \beta_{il} x_i}} \tag{10}$$

Donde los  $\beta_{ij}$  son los regresores que desean ser estimados, a través del criterio de máxima verosimilitud. Para el caso particular de la estimación del  $SF_{final}$ , la probabilidad de que una nueva observación pertenezca a  $p_j$  ( $j = 7, 8, \dots, 12$ ) se puede reescribir a partir de (9), como se muestra en la ecuación (11).

$$p_j = E(Y_j) = \frac{e^{\beta_{j0} + \beta_{j1} D + \beta_{j2} TP_{inicial} + \sum_{k=8}^{12} \beta_{j3,k} SF_{inicial_k}}}{1 + \sum_{l=1}^n e^{\beta_{l0} + \beta_{l1} D + \beta_{l2} TP_{inicial} + \sum_{k=8}^{12} \beta_{l3,k} SF_{inicial_k}}} \tag{11}$$

De tal manera, se tendrán 8 regresores por cada clase, y debido a que se tienen valores de  $SF_{final} = 7, 8, \dots, 12$ , se tendrán 5 probabilidades ( $SF_{final} = 8, \dots, 12$ ), debido a que  $SF = 7$  se considera como una variable de referencia. Así, el número de parámetros a ajustar será de 40, y se muestran en la Tabla 4. Se usa el software *R Studio*, librería *nnet*, método *multinom*.

SF <sub>final</sub>	Intercepto	D	TP <sub>inicial</sub>	SF <sub>inicial8</sub>	SF <sub>inicial9</sub>	SF <sub>inicial10</sub>	SF <sub>inicial11</sub>	SF <sub>inicial12</sub>
8	-7.918	0.270	-1.727	8.573	18.424	10.334	12.803	20.624
9	-9.036	0.339	-2.429	8.382	22.165	13.190	14.900	26.045
10	-11.290	0.372	-2.736	8.954	22.117	15.987	8.779	19.409
11	-12.799	0.380	-2.729	6.750	21.418	15.141	9.141	19.869
12	-11.873	0.402	-3.195	9.902	24.128	19.913	23.458	35.370

Tabla 4 – Parámetros estimados para el modelo de regresión logística multinomial

La evaluación se implementa generando una matriz de confusión (librería *caret*, método *confusionMatrix*), la cual contrasta los valores estimados vs. los valores esperados y realiza el conteo de aciertos y fracasos, y se muestra en la Tabla 5a (conjunto de entrenamiento) y Tabla 5b (conjunto de pruebas). Puede observarse que, si bien hay casos en los cuales hay éxito en las predicciones, i.e.  $SF_{final}=7,8$  y 12, hay otros casos en los cuales no se hacen predicciones correctas, i.e.  $SF_{final}=10,11$ . La exactitud del modelo es 88.3%, y el intervalo de confianza (IC) [85.81%, 90.4%].

### Máquinas de vectores de soporte (SVM)

Se realizan las evaluaciones con cuatro tipos diferentes de *kernels* (que se evalúan por medio de una matriz de confusión): lineal (exactitud: 86% IC 5%: (80.41%, 90.49%)), polinomial (no converge), sigmoideal (exactitud: 86.5% IC 5%: (83.93%, 88.79%)), y de base radial (exactitud: 85% IC 5%: (79.28%, 89.65%)). Dado que el *kernel* sigmoideal es el que entrega mejores resultados, se procede a sintonizar los hiper parámetros C y  $\gamma$ , cuyos valores son 5.1 y 0.125 respectivamente. No obstante, se presenta el mismo problema ilustrado en la regresión logística: no hay predicciones correctas para  $SF_{final}$  de 10 y 11, presentándose cero coincidencias, como se muestra en la Tabla 5. Se concluye que la SVM óptima hallada no es un clasificador válido para el problema en cuestión.

Pred. \ Ref (Entren.)	7	8	9	10	11	12
7	21	4	0	0	0	0
8	2	26	3	0	0	4
9	1	4	20	1	3	11
10	0	1	3	2	4	23
11	0	0	2	0	5	15
12	0	4	7	1	1	632

(a)

Pred. \ Ref (Entren.)	7	8	9	10	11	12
7	9	2	0	0	0	1
8	0	7	2	0	0	1
9	0	0	1	1	1	4
10	0	0	5	1	0	5
11	0	0	0	0	0	4
12	0	0	0	0	0	156

(b)

Tabla 5 – Matriz de confusión para valores del conjunto de entrenamiento/pruebas vs. valores reales

### Redes neuronales artificiales (ANN)

Un método alternativo para lidiar con el problema exhibido por la SVM es usar ANN. Dicho método exhibe algunas desventajas con respecto a las SVM, relacionado con el *overfitting* (sobre entrenamiento), el cual puede redundar en regresiones o clasificaciones erróneas. A su vez, el riesgo se calcula a partir de la minimización de una función de pérdida (normalmente gradiente descendente) de los datos de entrenamiento (conocido como riesgo empírico), lo cual puede redundar en generalización errónea, a diferencia de las SVM. Por tal motivo, es muy importante el entrenamiento exhaustivo hasta encontrar el modelo que reduzca el *overfitting* y generalice adecuadamente, lo cual requiere muchos recursos de hardware en conjuntos de datos de gran tamaño. Se presenta una ANN para solucionar el problema de predicción errónea de los SF finales. Se busca el mejor modelo a partir del paquete CARET de R, método *train*, configurando una grilla que varía los parámetros intrínsecos de la red (número de capas, número de neuronas por capa, *decay*, entre otros) y calcula un nuevo modelo por cada configuración de parámetros. La métrica de desempeño es el acierto (*accuracy*). Para asegurar el mejor modelo, se configuran 100 épocas. Se encuentra una ANN perceptrón con 7 neuronas de entrada (D, SF<sub>inicial</sub> *i*-ésimo (*i*= 8,...,12) y TP<sub>inicial</sub>), 8 neuronas en una única capa oculta, y 6 neuronas de salida (SF<sub>final</sub> *i*-ésimo (*i*=7,...,12)). La evaluación de la clasificación se hace usando una matriz de confusión, como se muestra en la Tabla 6. Puede observarse que los errores de clasificación mejoran con respecto a los dos métodos anteriores, exhibiendo una exactitud de 95% en el conjunto de entrenamiento y en el de prueba de 96%.

## 6. Método propuesto y evaluación

Una vez inferidos los algoritmos con mejor desempeño para inferir los parámetros TP<sub>final</sub> y SF<sub>final</sub>, se propone el Algoritmo 1. Dicho algoritmo se piensa para ser ejecutado en el servidor de red, de tal manera, los nodos enviarán la información necesaria al mismo: TP<sub>inicial</sub>, SF<sub>inicial</sub> y la distancia D desde el nodo hasta el *gateway* más cercano. Por seguridad, se recomienda inicializar los parámetros TP y SF a sus valores máximos,

y así se garantizará la entrega del primer paquete de información al servidor de red. Una vez el servidor recibe los parámetros, procede a calcular los valores mejorados, a través de SVM (en el caso de TP) y ANN (en el caso de SF), como se discutió en las secciones previas.

Pred. \ Ref (Entren.)	7	8	9	10	11	12
7	20	4	0	0	0	1
8	2	29	2	0	0	2
9	0	3	36	0	0	1
10	0	0	1	27	0	5
11	0	0	1	2	15	4
12	2	2	1	0	0	640

(a)

Pred. \ Ref (Entren.)	7	8	9	10	11	12
7	11	0	0	0	0	1
8	2	6	1	0	0	1
9	0	0	7	0	0	0
10	0	0	1	9	0	1
11	0	0	0	0	4	0
12	0	0	0	0	0	156

(b)

Tabla 6 – Matriz de confusión para valores del conjunto de entrenamiento/prueba

<b>Algoritmo 1:</b> Determinación de TP y SF usando SVM y ANN (en el servidor de red)	
1:	<b>entrada:</b> distancia D, TP <sub>inicial</sub> , SF <sub>inicial</sub>
2:	<b>parámetros:</b> SV, C, $\gamma$ , w, b
3:	<b>para cada nodo hacer</b> <span style="float: right;">➤ Inferir TP<sub>final</sub></span>
4:	recibir distancia D del nodo
5:	C $\leftarrow$ 10
6:	$\gamma$ $\leftarrow$ 0.125
7:	fijar SVR a SVM de base radial
8:	TP <sub>final</sub> $\leftarrow$ SVR(D, TP <sub>inicial</sub> , SF <sub>inicial</sub> , C, $\gamma$ , SV)
9:	<b>fin para</b>
10:	<b>para cada nodo hacer</b> <span style="float: right;">➤ Inferir SF<sub>final</sub></span>
11:	inicializar ANN Perceptrón multicapa: Capa entradas (7 neuronas), Una capa oculta (8 neuronas), Capa salida (6 neuronas), Pesos w, Bias b.
12:	SF <sub>final</sub> $\leftarrow$ max(SF <sub>final</sub> i-ésimo) $\leftarrow$ ANN(D, TP <sub>inicial</sub> , SF <sub>inicial</sub> , w, b)
13:	<b>fin para</b>
14:	<b>salida:</b> TP <sub>final</sub> , SF <sub>final</sub>

Para comparar la efectividad de los métodos propuestos para la estimación de los parámetros TP y SF, se analiza el tiempo de convergencia del algoritmo tradicional ADR vs. el tiempo que tarda el método propuesto en este trabajo. Al realizar la evaluación con respecto al parámetro TP, se encontró a través de las simulaciones que, de los 1000 nodos, 120 requirieron el ajuste del TP a través de ADR. El mínimo número de veces que el nodo debió solicitar ajustar el TP fue 20, y el máximo fue 485. Teniendo en cuenta el TP inicial, es posible hallar la potencia consumida en la transmisión en *watts*, si se tiene en cuenta que  $W = 10^{TP/10}/1000$ . Al realizar los cálculos, se encontraron potencias mínimas iniciales por paquete de  $1.5 \times 10^{-3}$  w y máximas iniciales de  $2.5 \times 10^{-2}$  w, las cuales fueron inicializadas de manera aleatoria.

En tal virtud, es posible calcular la energía consumida en exceso por las retransmisiones, si se conoce el ToA (ecuación 5). Utilizando los parámetros de la simulación ( $BW = 125$  kHz,  $min\ SF = 7$ ,  $max\ SF = 12$ ,  $Coding\ rate = 1$ , longitud del *payload* = 10 bytes), es posible inferir que el mínimo ToA es igual a 0.041 segundos y el máximo a 0.991 s. Así, la energía consumida en exceso (para  $SF = 7$ ) puede oscilar entre 0.002 J y 0.318 J y la energía consumida en exceso (para  $SF = 12$ ) puede oscilar entre 0.062 J y 7.67 J. Ahora bien, dado que el método que se propone en este trabajo sólo requiere enviar de manera satisfactoria desde el nodo hasta el *gateway* los parámetros iniciales TP, SF y distancia, el costo energético para el nodo para ajustar el algoritmo de control de energía será equivalente a una única transmisión/ recepción, la cual, en función de los valores de retransmisión enunciados previamente, pueden estar en proporciones desde 1/485 hasta 1/20, es decir, el método logra reducir, porcentualmente, la energía consumida por el algoritmo ADR tradicional al 5%, en el peor de los casos, lo cual exhibe una mejora sustancial en consumo energético.

En cuanto al tiempo de convergencia, se encontró que el tiempo mínimo para hallar el valor óptimo fue de 1871 s, y el máximo fue de 86.314 s; no obstante, dicho tiempo está en función de las transmisiones que haga cada nodo, las cuales fueron aleatorizadas a través de una distribución exponencial.

## 7. Conclusiones y trabajo futuro

En el presente trabajo se presentó un método basado en técnicas no paramétricas de aprendizaje de máquina que permiten estimar la potencia consumida en un nodo de Internet de las Cosas que transmita información a través del protocolo LoRaWAN. Para recolectar los datos, se usó el simulador Omnet++ usando el *framework FLoRa*, en el cual se desplegaron 1000 nodos. La energía consumida en un nodo, como se mostró, depende de la distancia, la potencia de transmisión y el número de bits por símbolo, valores que son sintonizados a través del algoritmo ADR provisto por el protocolo LoRaWAN. Se mostró que la cantidad de paquetes transmitidos para sintonizar dichos valores oscila entre 20 y 485, exhibiendo una posible mejora para el tiempo de convergencia, y de esta manera, una reducción en el consumo energético. Del conjunto de métodos evaluados, se encontró que las máquinas de vectores de soporte (SVM) son efectivas para la regresión de energía por paquete (EPP) y para ajustar la potencia de transmisión ( $TP_{final}$ ), en tanto las redes neuronales son efectivas para clasificar el valor del número de bits por símbolo ( $SF_{final}$ ). Se propone un algoritmo para modificar el ADR tradicional,

el cual corre en el servidor y asegura la convergencia de los parámetros  $SF_{final}$  y  $TP_{final}$  en una única transmisión/recepción por nodo. Se planea, como trabajo futuro, realizar la implementación física en un ambiente de radio definido por software, para asegurar la validez de los modelos y métodos propuestos en un ambiente de operación real.

## Agradecimientos

A la Universidad de Medellín por financiar el trabajo de investigación “Esquema algorítmico para la transmisión de información que minimice el consumo energético y mejore el uso del espectro radioeléctrico en dispositivos de internet de las cosas”, código de proyecto 1039, y por la beca de estudios doctorales otorgada a uno de los autores.

## Referencias

- Abdelfadeel, K. Q., Cionca, V., & Pesch, D. (2018). *Fair adaptive data rate allocation and power control in lorawan*. Paper presented at the 2018 IEEE 19th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM).
- Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the limits of LoRaWAN. *IEEE Communications magazine*, 55(9), 34-40.
- Bouguera, T., Diouris, J.-F., Chaillout, J.-J., Jaouadi, R., & Andrieux, G. (2018). Energy consumption model for sensor nodes based on LoRa and LoRaWAN. *Sensors*, 18(7), 2104.
- Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, 19(2), 171-209.
- de Carvalho-Silva, J., Rodrigues, J. J., Alberti, A. M., Solic, P., & Aquino, A. L. (2017). LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities. In *Proceedings of the 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*.
- Gao, W., Du, W., Zhao, Z., Min, G., & Singhal, M. (2019). Towards Energy-Fairness in LoRa Networks. In *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*.
- Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & information systems engineering*, 6(4), 239-242.
- Lavric, A., Petrariu, A. I., & Popa, V. (2019). Long range sigfox communication protocol scalability analysis under large-scale, high-density conditions. *IEEE Access*, 7, 35816-35825.
- Lee, J., Davari, H., Singh, J., & Pandhare, V. (2018). Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. *Manufacturing letters*, 18, 20-23.
- Li, S., Raza, U., & Khan, A. (2018). How Agile is the Adaptive Data Rate Mechanism of LoRaWAN? In *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*.



- Noble, W. S. (2006). What is a support vector machine?. *Nature biotechnology*, 24(12), 1565-1567.
- Petrasch, R., & Hentschke, R. (2016). Process modeling for Industry 4.0 applications: Towards an Industry 4.0 process modeling language and method. In *Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*.
- Ratasuk, R., Vejlgard, B., Mangalvedhe, N., & Ghosh, A. (2016). NB-IoT system for M2M communication. In *Proceedings of the the 2016 IEEE wireless communications and networking conference*.
- Reynders, B., Meert, W., & Pollin, S. (2017). Power and spreading factor control in low power wide area networks. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*.
- Reynders, B., & Pollin, S. (2016). *Chirp spread spectrum as a modulation technique for long range communication*. Paper presented at the 2016 Symposium on Communications and Vehicular Technologies (SCVT).
- San Cheong, P., Bergs, J., Hawinkel, C., & Famaey, J. (2017). *Comparison of LoRaWAN classes and their power consumption*. Paper presented at the 2017 IEEE symposium on communications and vehicular technology (SCVT).
- Sandoval, R. M., Garcia-Sanchez, A.-J., & Garcia-Haro, J. (2019). Performance optimization of LoRa nodes for the future smart city/industry. *EURASIP Journal on Wireless Communications and Networking*, 2019(1), 1-13.
- Sandoval, R. M., Rodenas-Herraiz, D., Garcia-Sanchez, A.-J., & Garcia-Haro, J. (2020). Deriving and Updating Optimal Transmission Configurations for Lora Networks. *IEEE Access*, 8, 38586-38595.
- Schwab, K. (2017). *The fourth industrial revolution*. Penguin Random House Grupo Editorial España, 2016.
- Slabicki, M., Premsankar, G., & Di Francesco, M. (2018). *Adaptive configuration of LoRa networks for dense IoT deployments*. Paper presented at the NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium.
- Sornin, N., Luis, M., Eirich, T., Kramp, T., & Hersent, O. (2015). *Lorawan specification*. LoRa alliance. <https://www.lora-alliance.org>
- Wollschlaeger, M., Sauter, T., & Jasperneite, J. (2017). The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE industrial electronics magazine*, 11(1), 17-27.