TECNIACUSTICA ELCHE · 2022

53º CONGRESO ESPAÑOL DE ACÚSTICA
XII CONGRESO IBÉRICO DE ACÚSTICA

ADOS: DETECCIÓN DE ANOMALÍAS EN REDES DE SENSORES ACÚSTICOS MEDIANTE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO DISTRIBUIDOS

PACS: 43.50.Rq Ruido medioambiental, medida, análisis, características estadísticas

Navarro Ruiz, Juan Miguel[1], Tomás Gabarrón, Juan Bautista[2], García-Collado, Ángel Joaquín[1], Caicedo García, Jesús[2], Berenguer Vidal, Rafael[1]

[1] Escuela Politécnica Superior, Universidad Católica de Murcia, Murcia, España
[2] Cubic Fort S.L., Murcia, España

Palabras Clave: acústica ambiental, redes de sensores acústicos, aprendizaje máquina, detección de anomalías, cadenas de bloques.

ABSTRACT

In recent years, many acoustic sensor networks have been deployed in cities in all countries allowing the collection of a huge amount of data on sound pressure levels that are helping environmental managers. However, these autonomous and communication-capable devices are exposed to different anomaly events, such as malfunctions and malicious attacks. In this work, an unsupervised machine learning algorithm, namely a graph neural network (GNN), has been applied for early detection of anomalous data from acoustic nodes. This algorithm has been trained and evaluated with data obtained from the open data portal of the acoustic sensor network of the city of Barcelona. Finally, a distributed execution system for anomaly detection (ADOS) has been designed with an application programming interface (API), allowing the use and integration of this service in other platforms or third-party sensor networks.

RESUMEN

En los últimos años, se han desplegado muchas redes de sensores acústicos en ciudades de todos los países permitiendo la recogida de una enorme cantidad de datos sobre niveles de presión sonora que están ayudando a los gestores ambientales. Sin embargo, estos dispositivos autónomos y con capacidad de comunicación están expuestos a diferentes eventos de anomalías, tales como averías y ataques malintencionados. En este trabajo, se ha aplicado un algoritmo de aprendizaje automático no supervisado, en concreto una red neuronal gráfica (GNN), para la detección temprana de datos anómalos procedentes de los nodos acústicos. Este algoritmo ha sido entrenado y evaluado con datos obtenidos del portal de datos abiertos de la red de sensores acústicos de la ciudad de Barcelona. Finalmente, se ha diseñado un sistema de ejecución distribuida para la detección de anomalías (ADOS) con una interfaz de programación de aplicaciones (API), permitiendo la utilización e integración de este servicio en otras plataformas o redes de sensores de terceros.

1. INTRODUCCIÓN

La inteligencia artificial (IA) se asocia a la simulación de procesos de inteligencia humana por parte de las máquinas, especialmente los sistemas informáticos, y en general a cualquier conducta humana que desarrolle una máquina o sistema [1]. En la forma más básica de inteligencia artificial, los sistemas informáticos están programados para emular la conducta humana utilizando grandes conjuntos de datos previos de conductas similares. Mediante el

aprendizaje profundo, el pensamiento estratégico u otras tipologías de inteligencia artificial, la clave de su empleo está en aquellas situaciones que requieren respuestas rápidas ante cualquier tipo de eventos [2]. Con la inteligencia artificial, las máquinas pueden trabajar de forma eficiente y analizar enormes cantidades de datos en tiempo real, y resolver problemas mediante un aprendizaje supervisado, no supervisado o reforzado.

La inteligencia artificial está fuertemente asociada a la tecnología de cadenas de bloques (*blockchain*). Mientras la primera ayuda a valorar, comprender, reconocer y decidir, *blockchain* ayuda a verificar, ejecutar y registrar, dando entidad y seguridad en todos los procesos digitales que pueden tener lugar. Así, la inteligencia artificial ayuda a encontrar oportunidades y mejorar la toma de decisiones, y los *smart contracts* (contratos inteligentes) y la tecnología *blockchain* automatizan la verificación de las partes transaccionales del proceso [3]-[4]. En definitiva, la inteligencia artificial y la tecnología *blockchain* son complementarias y sinérgicas. Sin embargo, siendo las cadenas de bloques redes seguras y fiables para el intercambio de valores, carecen de la funcionalidad para, de forma nativa, obtener datos externos o enviar datos desde/hacia sistemas fuera de la cadena mientras se mantiene la resistencia a la manipulación de extremo a extremo [5]. En este sentido, los oráculos desempeñan un papel extremadamente importante facilitando la información de las diferentes fuentes de datos fuera de la cadena que los contratos inteligentes necesitan para operar. No obstante, todavía existen varios retos para mejorar la fiabilidad de los oráculos de *blockchain*, tales como prevenir o contrarrestar los ataques sibilinos (corrupción de datos, ataques masivos, *freeloading*, etc.), proporcionar nuevos algoritmos y arquitecturas para una validación de datos más eficaz, u optimizar los costes operativos de los oráculos que recogen datos de fuentes externas, conservando o mejorando la fiabilidad.

Debido a lo importante que resulta prevenir y contrarrestar ataques no deseados, en este trabajo nos centraremos en la detección de anomalías para identificar patrones anormales dentro de un conjunto de datos [6]-[7]. Los sistemas definen la detección de anomalías como "un método utilizado para identificar patrones irregulares o inusuales en un entorno complejo", es decir, la detección de anomalías detecta patrones de una manera que un humano no puede. Discriminar si un dato o un valor es normal o anormal es un problema de clasificación, que generalmente se resuelve mediante el aprendizaje supervisado con una combinación grande y balanceada de datos etiquetados en las dos clases. Cualquier comportamiento que se digitalice o mida numéricamente, está sujeto a la detección de anomalías.

Las plataformas de IA deben poder responder a los cambios rápidos de información y datos bajo serias amenazas de seguridad. No hay una forma manual para dar fiabilidad a grandes conjuntos de datos ya que para un equipo de personas no es posible analizar e interpretar miles de estadísticas por segundo, pero una solución de IA sí puede afrontar la detección de anomalías que proporciona una interpretación en tiempo real de la actividad de datos [8]. Las plataformas de detección de anomalías pueden profundizar en datos en tiempo real para identificar anomalías que un usuario humano no detectaría en un panel de control.

El objetivo principal de este trabajo ha sido el entrenamiento y la validación de un algoritmo de aprendizaje automático basado en redes neuronales gráficas para la detección de anomalías en los datos proporcionados por una red de sensores acústicos inalámbricos. En concreto, se han entrenado diferentes modelos utilizando un conjunto de datos de la red de la ciudad de Barcelona. La evaluación y la validación del algoritmo se ha llevado a cabo mediante una serie de experimentos donde se han medido las prestaciones de los modelos propuestos.

2. ARQUITECTURA DE LA SOLUCIÓN ADOS

ADOS (*AirTrace Decentralized Oracle System*) es un servicio de valor añadido fundamental de AirTrace (airtrace.io), una plataforma SaaS (*Software as a Service*) que permite a los integradores de sistemas de IoT (Internet de las Cosas) y a las plataformas SaaS de IoT respaldar las capacidades avanzadas de ciberseguridad mediante la integración de las capacidades de *blockchain* en sus productos y servicios de forma escalable y ágil. ADOS, en

particular, se basa en el desarrollo de un esquema de puntuación de la fiabilidad de los datos antes de su inyección en la *blockchain*.

La razón de ser de ADOS radica en que los datos del mundo físico están intrínsecamente sujetos a riesgos de manipulación, corrupción o cualquier tipo de alteración que pueda afectar a su calidad antes de ser llevados a la *blockchain*. La propuesta presentada en este trabajo se basa en el empleo de un modelo GNN [9] dinámico que construye un conjunto de grafos incrustados para la red de dispositivos IoT que miden una magnitud física específica. Mediante GNN es posible ejecutar procedimientos de detección de anomalías en los datos proporcionados por los sensores, permitiendo detectar si las mediciones periféricas han sido (o no) manipuladas [10].

Así, es posible inyectar en la *blockchain* los datos medidos y un Factor de Calidad de Datos (Data Quality Factor, DQF) que puede utilizarse posteriormente para auditar los datos. Además, gracias a la plataforma iExec, ADOS puede ejecutar tareas de inferencia en una configuración distribuida de trabajadores que, mediante un algoritmo de consenso, devolverán el DQF al blockchainizar los datos, permitiendo que sean recompensados con tokens RLC gracias a su contribución.

En resumen, el objetivo principal de ADOS es capturar las complejas relaciones entre sensores y detectar y explicar las anomalías que se desvían de estas relaciones mediante el uso de tecnologías basadas en la IA que pueden explotar las propiedades subyacentes y no visibles de los sensores que suelen correlacionarse cuando aparecen estas anomalías. En la arquitectura completa de AirTrace (Figura 1) se tienen diferentes clientes. AirTrace genera datos de monitorización IoT que alimentan a la plataforma a intervalos regulares. Los datos pueden ser blockchainizados (inyectados en la *blockchain*) según la granularidad de intervalo específica establecida por los clientes (por ejemplo, acumulando todas las lecturas de los sensores durante 24 horas para incluirlas todas en una única transacción de *blockchain*). En este caso, antes de inyectar en la cadena de bloques, AirTrace envía el conjunto acumulado de lecturas de IoT a ADOS para realizar la detección previa de posibles anomalías.

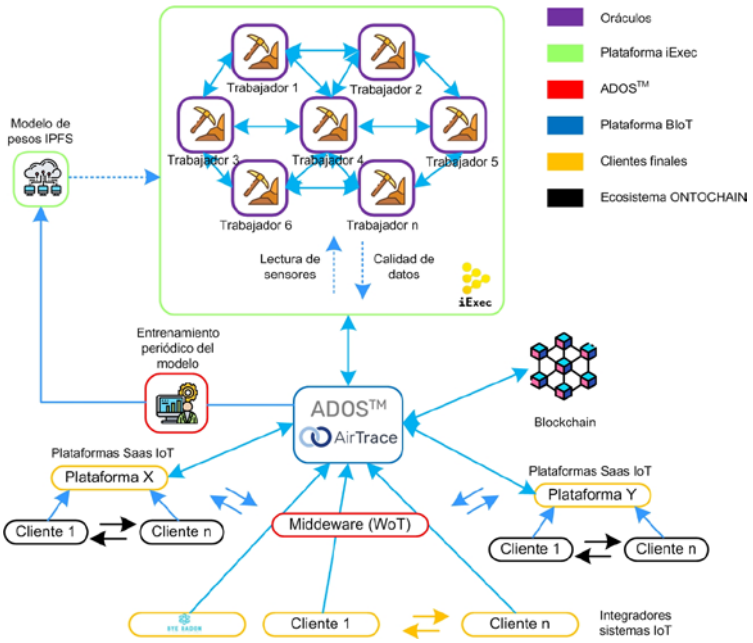


Figura 1 – Arquitectura principal de AirTrace.

3. ALGORITMO DE DETECCIÓN DE ANOMALÍAS

Los datos de entrenamiento consisten en series temporales multivariantes de datos de N sensores durante el periodo temporal T_{train}. Los datos de los sensores constituyen un vector que queda determinado por $s_{train} = \left[s_{train}^{(1)}, \dots, s_{train}^{(T_{train})}\right]$ y se utilizan para entrenar nuestro modelo. En cada instante t, los valores del sensor $s_{train}^{(t)} \in \mathbb{R}^N$ constituyen un vector de N dimensiones que representa los valores de medición de los N sensores. Siguiendo la formulación habitual de detección de anomalías no supervisada, se supone que los datos de entrenamiento consisten únicamente en datos normales. El objetivo es detectar anomalías en los datos de prueba que provienen de los mismos N sensores, pero en un conjunto separado de tiempos de prueba. Los datos de prueba se denominan $s_{test} = \left[s_{test}^{(1)}, \dots, s_{test}^{(T_{test})}\right]$. La salida de nuestro algoritmo es un conjunto de etiquetas binarias T_{test} que indican si cada instante de tiempo de la prueba es una anomalía o no, es decir, $a(t) \in \{0, 1\}$, donde $a(t) = 1$ indica que el instante t es anómalo, debido a un fallo en el sensor, o bien a un ataque externo.

El DQF tentativo para cada lectura del sensor se obtiene calculando el porcentaje de anomalías con respecto al número total de lecturas durante la prueba T. Se ha aplicado la teoría de las redes neuronales de grafos utilizando las incrustaciones de los sensores, Figura 2, para aprender las relaciones entre los sensores como un grafo, Figura 3, y luego identificaremos y explicaremos las desviaciones de las pautas aprendidas.

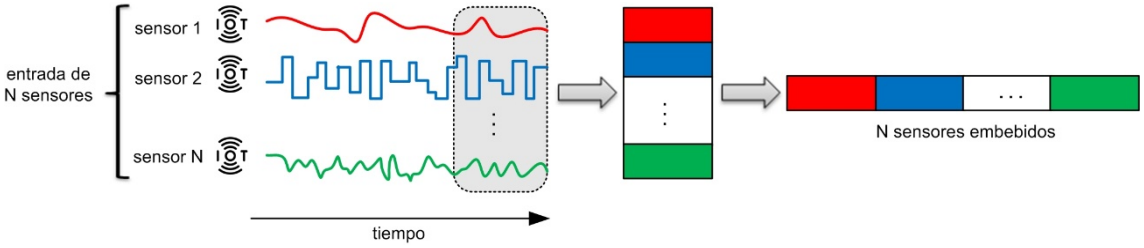


Figura 2 – Datos de los sensores embebidos. Adaptada de [10].

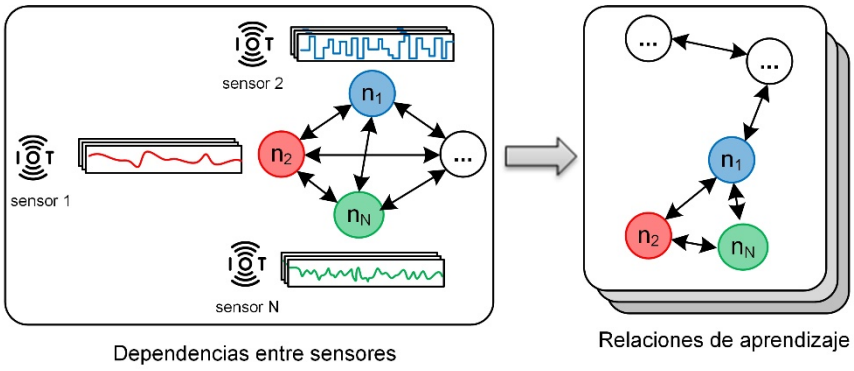


Figura 3 – Estructura de aprendizaje gráfico. Adaptada de [10].

Como podemos ver, uno de los valores añadidos más importantes del modelo corresponde al uso de características que se incluirán en cada sensor para capturar sus características únicas. Así, el esquema de detección de anomalías deberá aprender una estructura de grafos que representan las relaciones de dependencia entre los sensores. En un contexto más amplio, el objetivo es ser capaces de identificar las desviaciones de las relaciones aprendidas, y localizar y explicar estas desviaciones/anomalías, extrayendo al mismo tiempo aquellos datos que tienen la mayor relevancia en el proceso de detección de anomalías.

4. USO EN UNA RED DE SENSORES ACÚSTICOS

Los sensores acústicos se utilizan para controlar el nivel de presión sonora de las ciudades, relacionados con la contaminación acústica que afecta a la calidad de vida de los habitantes.

En este trabajo, se ha probado la arquitectura propuesta con un conjunto de datos recogidos en Barcelona (España). La red de nodos acústicos desplegada en Barcelona por el ayuntamiento durante los últimos años consta de 70 sensores de sonido [11]. El conjunto de datos utilizado en este caso de uso proporciona un análisis a largo plazo, y como se muestra en la Figura 4, los nodos acústicos están distribuidos uniformemente por toda la ciudad, pero el centro de la ciudad concentra el mayor número de nodos. Cada nodo capta la presión sonora de su ubicación en modo continuo, 24 h/7 días a la semana, enviando un nivel de presión sonora equivalente en dBA durante 1 minuto a la base de datos del sistema.

Figura 4 – Localización de los nodos acústicos en la red de la ciudad de Barcelona utilizada para el caso de uso.

4.1. Experimentos Realizados

Para determinar las prestaciones de la GNN utilizada como oráculo para predecir los fallos de los sensores [10], se ha diseñado e implementado un protocolo de prueba para el sistema. Para la validación del modelo se ha trabajado con subconjunto de los datos [11], en concreto los primeros 37 sensores, con el fin de reducir la complejidad del modelo trabajando con un conjunto de datos más reducido. Como intervalo temporal, se ha probado con bloques de un mes de datos, en concreto los datos del mes de septiembre de 2019, correspondiendo a un total de 43.200 valores temporales para cada uno de los 37 sensores seleccionados.

En primer lugar, se configura la GNN con una determinada parametrización, siendo entrenada posteriormente con el conjunto de datos proporcionados por los sensores. Notar que el modelo GNN utilizado [10] permite trabajar con datos de diferente naturaleza, pero para este fin requiere que los valores del vector de datos de cada uno de los sensores estén en el rango de valores [0,1]. Como para esta aplicación todos los sensores son del mismo tipo, nodos acústicos, se ha entrenado la GNN con valores normalizados de forma lineal al rango [0,1] y también con los valores de presión sonora en dBA proporcionado por los sensores. Además de este paso, es necesario asignar un valor numérico a los datos ausentes por fallo del sensor o de la red de transmisión, etiquetados normalmente como NaN en la base de datos [11].

A continuación, el conjunto de datos se divide en dos subconjuntos, destinando un 80% para entrenamiento (34.400 valores de presión sonora por sensor) y un 20% para validación (8.600 valores). Este proceso se realiza recursivamente con diferentes subdivisiones de los datos, para minimizar el error de estimación debido a una selección no arbitraria de ambos conjuntos. El subconjunto de entrenamiento es utilizado a continuación para entrenar la GNN, la cual permitirá estimar la calidad de los datos procedentes de dicho conjunto de sensores.

Posteriormente, se aplica un conjunto de ataques de una determinada tipología al subconjunto de datos de validación. Los ataques emulan los errores que podrán tener los sensores y hay tres dimensiones a tener en cuenta. La primera de ellas es el nivel del ataque (valor de presión sonora proporcionada por el sensor), pudiendo ser un valor anómalo, pero de un rango habitual de valores, o un valor excepcionalmente extraño por encima o por debajo del valor habitual. La segunda dimensión es la temporal, esto es la frecuencia de los ataques, así como la duración

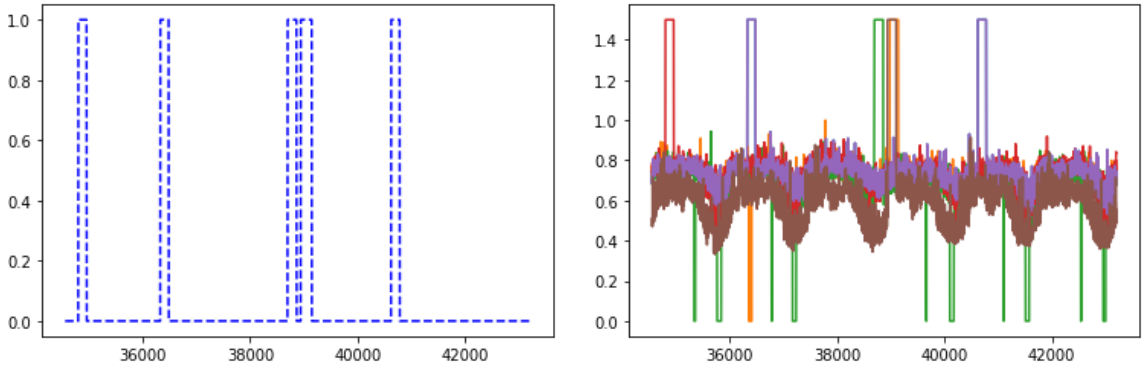
de los mismos. En tercer lugar, tenemos la dimensión espacial, esto es, si los ataques se producen sobre sensores cercanos (emulan por ejemplo un fallo en una determinada zona de la ciudad) o de forma esporádica sobre cualquier sensor (emulan fallos aleatorios sobre sensores sin importar la zona o el lugar donde este sensor se encuentre).

En este experimento hemos estudiado los fallos aleatorios sobre sensores, midiendo el efecto del nivel del ataque y de la duración de los mismos, en la calidad del modelo GNN para la predicción de estos ataques, obteniendo las métricas de evaluación, en concreto *precision*, *recall* y *F1-score*. *Precision* se define con la expresión $Prec = TP/(TP + FP)$ mientras que el *recall* se calcula mediante $Rec = TP/(TP + FN)$, donde TP son los positivos verdaderos (valores erróneos que han sido correctamente detectados como fallos), FP se refiere a los falsos positivos (valores correctos detectados erróneamente como fallos), FN son los falsos negativos (valores erróneos no detectados). Por último, el *F1-score* se calcula a partir de *precision* y *recall*, con $F1 = 2 \cdot Prec \cdot Rec/(Prec + Rec)$. Se debe observar que, para cualquiera de las métricas definidas, una mayor calidad en la predicción se refleja en métricas cercanas a la unidad. Por diferentes motivos, aunque se proponen valores de referencia para los parámetros del sistema, es aconsejable realizar un proceso de aprendizaje periódico por parte de la GNN para acomodar los datos cambiantes del escenario de uso a lo largo del tiempo.

5. RESULTADOS Y DISCUSIÓN

5.1. Experimento A

El primer conjunto de pruebas se ha realizado sobre valores de presión sonora en dBA normalizados a [0,1] en cada sensor. Esto es, sobre el vector de datos de los valores de cada sensor en dBA se fija al mínimo a 0 y el máximo a 1, escalándose el resto de valores de forma lineal. Para este primer grupo, los valores NaN de los sensores se han fijado a valor nulo. Se aplican ataques ráfagas de 150 unidades temporales, fijando el valor proporcionado por el sensor durante el ataque a 1,5. En cada ataque, se elige de forma aleatoria un sensor del conjunto al que le afectará dicho ataque, manteniendo el resto de sensores con sus valores correctos de presión sonora. La Figura 5(a) muestra los ataques fijados y la figura 5(b) muestra los valores de cada sensor tras dichos ataques. En dicha figura podemos observar que durante algunos instantes los sensores muestran el valor nulo. Estos valores nulos corresponden a los instantes en que el sensor aparece no disponible y por tanto, sus datos son NaN, reemplazándose dichos valores por nulos.



(a) (b)

Figura 5 – (a) Ataques aplicados sobre el sistema. (b) Conjunto de valores de cada uno de los sensores en el conjunto de validación, cada sensor se representa en un color diferente.

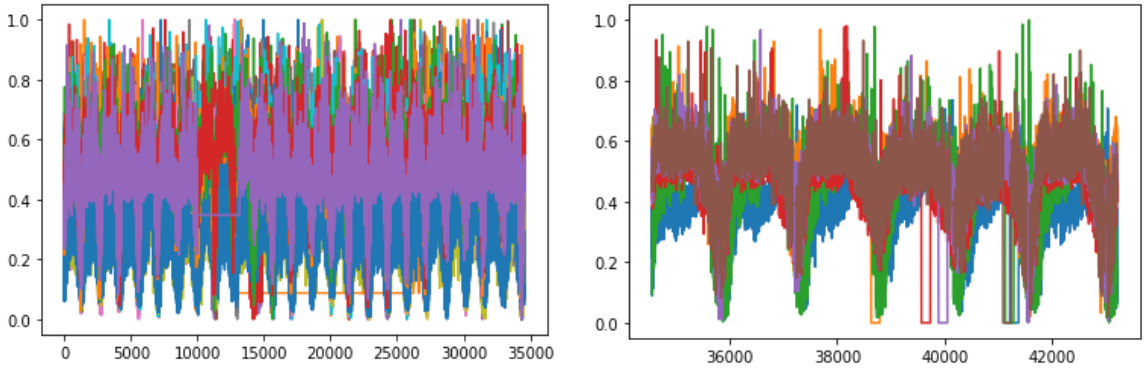
Al aplicar la GNN a dichos datos se obtienen los resultados mostrados en la Tabla 1. A pesar de tener una tasa de precisión elevada (no aparece ningún falso positivo), el *recall* muestra un valor bajo, lo cual indica que existen falsos positivos, que no han sido detectados. Uno de los motivos de esta baja tasa de recall, es debido a que la red GNN se entrena con los valores

NaN como nulos, provocando que la red interprete estos valores nulos como valores normales, en lugar de considerarlos como valores que extraños y fuera de rango.

Tabla 1 – Resultado de la aplicación de la GNN a los datos en Experimento A

F1 score	precision	recall
0.366	1.00	0.224

Con el fin de minimizar este problema, se repiten los experimentos se repiten reemplazando en esta ocasión los valores NaN de los sensores en los datos del conjunto de entrenamiento por valores medios del sensor. Se aplica la GNN, obteniendo no obstante valores similares al caso anterior.

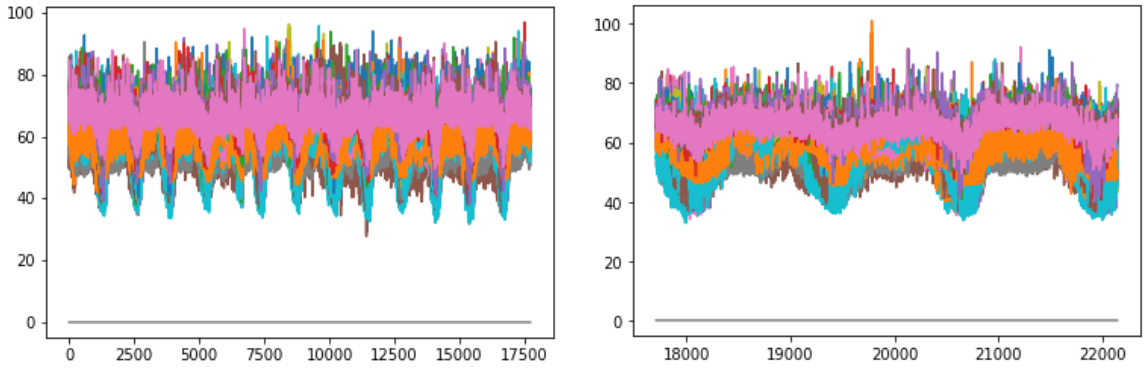


(a) (b)

Figura 6 – (a) Valores de los sensores en el conjunto de entrenamiento. En este caso, los valores NaN de los sensores se sustituyen por valores medios de dicho sensor. (b) Valores de los sensores del conjunto de validación.

5.2. Experimento B

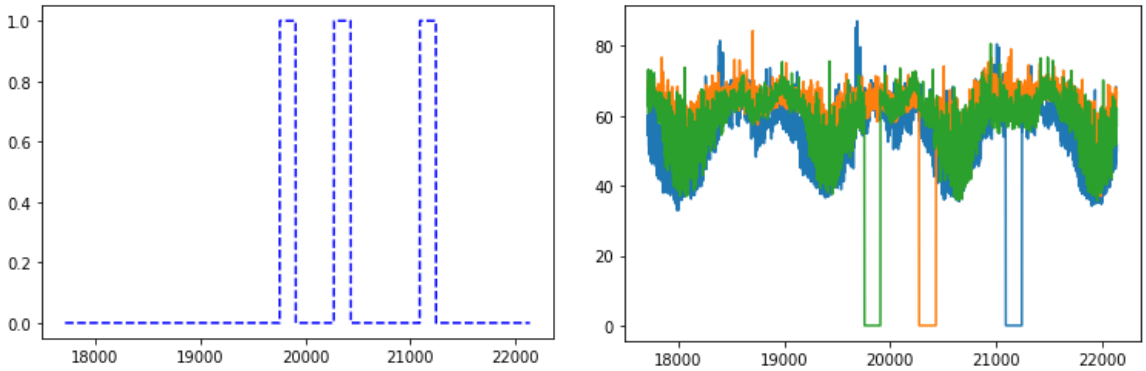
Dado que el entrenamiento de la red es fundamental para obtener una buena predicción, se aplica un nuevo conjunto de pruebas, eliminando los valores de los sensores con valores NaN, con el objetivo de que no puedan influir estos datos corruptos en el entrenamiento de la propia red. Además, los valores de presión sonora no se normalizan, esto tanto durante el entrenamiento como en la validación, utilizando los datos de presión sonora en dBA. La Figura 7 muestra ambos conjuntos de datos, entrenamiento y validación.



(a) (b)

Figura 7 – Datos con eliminación de valores NaN de los sensores. (a) Conjunto de datos de entrenamiento. (b) Conjunto de valores de sensores del conjunto de validación.

Se introducen de nuevo ráfagas de ataques de duración 150 marcas temporales, fijando el valor 0 a los sensores durante dichos ataques. La Figura 8(a) muestra los intervalos de ataque y la Figura 8(b) muestra los valores de los sensores en el conjunto de validación tras dichos ataques. La aplicación de la GNN a este conjunto de datos proporciona los resultados de la Tabla 2, notablemente mejores que en el caso anterior, con métricas muy cercanas a la unidad.



(a) (b)

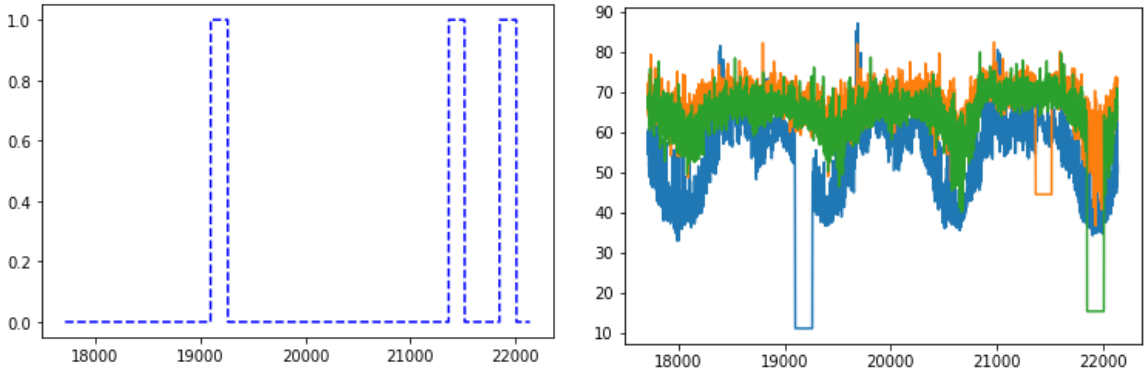
Figura 8 – (a) Ataques aplicados sobre el sistema. (b) Valores de cada uno de los sensores en el conjunto de validación.

Tabla 2 – Resultado de la aplicación de la GNN a los datos en Experimento B

F1 score	precision	recall
0.989	0.985	0.994

5.3. Experimento C

Con el fin de comprobar la robustez de la predicción, cuando los sensores proporcionan valores anormales, pero distintos del valor nulo, se realizan dos conjuntos de pruebas, en ambos casos utilizando una GNN entrenada con datos sin datos NaN, como en el experimento B. En primer lugar, los ataques se fijan de nuevo con duraciones de 150 marcas temporales, pero ahora se fija durante cada ráfaga de error, un valor aleatorio entre [10 dB, 60 dB]. La Figura 9(a) muestra los periodos de los ataques, mientras que la Figura 9(b) muestra los valores de los sensores, tras estos ataques en el conjunto de validación de los datos. La Tabla 3 muestra los resultados de dos pruebas realizadas. En función del valor proporcionado por el sensor de los ataques las métricas son variables, siendo mayores, cuando más diferencia aparece entre el nivel del ataque y los valores medios de cada sensor.



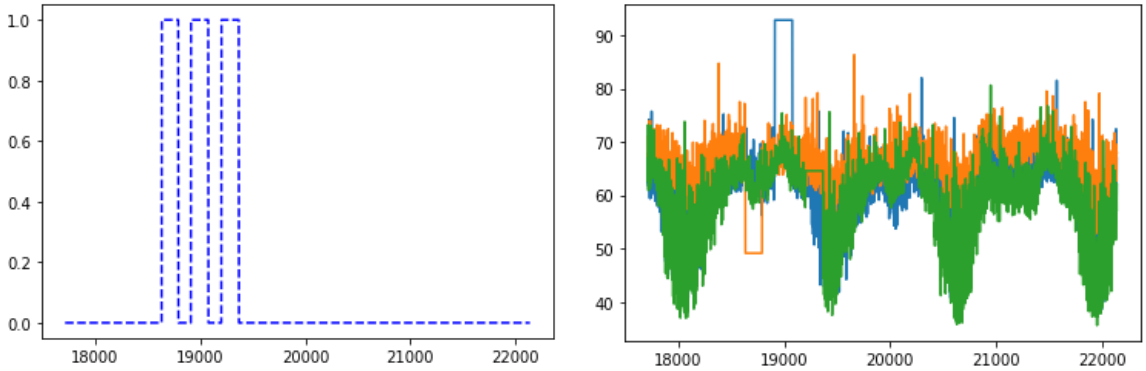
(a) (b)

Figura 9 – (a) Ataques aplicados sobre el sistema. (b) Valores de cada uno de los sensores en el conjunto de validación.

Tabla 3 – Resultado de la aplicación de la GNN a los datos en Experimento C1

F1 score	precision	recall
0.857	0.965	0.770
0.984	0.981	0.987

De nuevo se modifica el experimento anterior, cambiando el rango de valores de los sensores atacados a [50 dB, 95 dB], valores más cercanos a los que proporcionan la mayoría de los sensores. La Figura 10 muestra estos datos. Como se puede observar en los datos de la Tabla 4, aunque la precisión se mantiene, el valor de *recall* disminuye como era de prever, dado que hay valores fijados como fallos que, la GNN no es capaz de detectar porque los asume como valores habituales.



(a) (b)

Figura 10 – (a) Ataques aplicados sobre el sistema. (b) Valores de cada uno de los sensores en el conjunto de validación.

Tabla 4 – Resultado de la aplicación de la GNN a los datos en Experimento C2

F1 score	precision	recall
0.785	0.985	0.653
0.773	0.987	0.635

6. CONCLUSIONES

El crecimiento de la demanda a escala mundial de soluciones para la detección temprana de anomalías en IoT se atribuye al aumento de los incidentes de amenazas internas y fraudes de datos. Por ejemplo, el mercado de la detección de anomalías en Estados Unidos se estima en 2.100 millones de dólares en el año 2022. Europa representa el segundo mercado más importante en la detección de anomalías debido a los avances tecnológicos y a las importantes inversiones de las empresas para crear nuevas soluciones.

Como tecnologías globales de cara a dar respuesta a esta gran demanda, las tecnologías de aprendizaje automático y de IA sirven en gran medida para dar soluciones destinadas a ayudar a los usuarios a identificar rápidamente los cambios bruscos en los patrones y el comportamiento.

En los últimos años, se están desplegando numerosas redes de sensores acústicos en grandes ciudades. Estas redes, como cualquier sistema que genera datos a gran escala, también requieren de sistemas de seguridad y protección de los datos por lo que es interesante la aplicación de métodos de detección de anomalías. En este trabajo, se han evaluado las prestaciones de las redes neuronales gráficas (GNN), utilizando datos capturados en la red de sensores de la ciudad de Barcelona. Los resultados aportados en este artículo dan fiel reflejo de las capacidades de estos paradigmas de cara a la detección temprana de anomalías en entornos donde la IoT posee una importancia fundamental.

AGRADECIMIENTOS

Este trabajo ha recibido financiación del programa de investigación e innovación Horizon 2020 de la Unión Europea bajo el proyecto NGI OntoChain bajo el acuerdo de ayuda Nº 957338

REFERENCIAS

[1] Bellman, R. (1978). An introduction to artificial intelligence: can computers think?. Thomson Course Technology.

[2] Rich, E., & Knight, K. (1991). Artificial Intelligence McGraw–Hill. New York.

[3] Russell Stuart, J., & Norvig, P. (2009). Artificial intelligence: a modern approach. Prentice Hall.

[4] Nilsson, N. J., & Nilsson, N. J. (1998). Artificial intelligence: a new synthesis. Morgan Kaufmann.

[5] Torres-Domínguez, O., Sabater-Fernández, S., Bravo, L., Martín, D. & García-Borroto, M. ResearchGate. (2019). Detección de anomalías en grandes volúmenes de datos.

[6] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. Statistical science, 17(3), 235-255.

[7] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

[8] Warren, J., & Marz, N. (2015). Big Data: Principles and best practices of scalable realtime data systems. Simon and Schuster.

[9] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29.

[10] Deng, A., & Hooi, B. (2021, May). Graph neural network-based anomaly detection in multivariate time series. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 5, pp. 4027-4035).

[11] Camps, J. (2015, May–June). Barcelona noise monitoring network. In Proceedings of the EuroNoise, Maastricht, The Netherlands, pp. 218–220.