

Algoritmos supervisados de *machine learning* para determinar la ubicación de dispositivos wifi

Javier More Sánchez

javier.more@unmsm.edu.pe

<https://orcid.org/0000-0003-1313-6751>

Universidad Nacional Mayor de San Marcos, Perú

Recibido: 6 de agosto del 2022 / Aceptado: 23 de septiembre del 2022

<https://doi.org/10.26439/ciis2022.6074>

RESUMEN. El presente artículo tiene como objetivo elegir, bajo un determinado escenario, el mejor algoritmo supervisado de *machine learning* para localizar un terminal que soporte wifi. Se usa un dataset que cuenta con 2000 registros de Received Signal Strength Indicator (RSSI), obtenidos de 7 puntos de acceso (AP), los cuales se cargan en 8 algoritmos supervisados de machine learning. Luego se elige el algoritmo que realiza la predicción más precisa, incluso cuando se cuenta con un menor número de AP. La mayor precisión se logra con el algoritmo *naive Bayes*, tanto para el caso de 7 AP (99 % de precisión) como para cuando se cuenta con un número menor de AP. Asimismo, se observa que los algoritmos basados en redes neuronales presentan el peor rendimiento. Finalmente, se proponen trabajos futuros para continuar con la investigación sobre el tema de localización de dispositivos wifi en interiores.

PALABRAS CLAVE: aprendizaje automático, localización en interiores, RSSI, wifi

SUPERVISED MACHINE LEARNING ALGORITHMS TO DETERMINE THE LOCATION OF WI-FI DEVICES

ABSTRACT. This article aims to choose the best-supervised machine learning algorithm for locating a terminal that supports Wi-Fi in a specific scenario. It uses a dataset of 2000 received signal strength indicator (RSSI) records obtained from 7 access points (AP), loaded into eight supervised machine learning algorithms. The algorithm that produces the most accurate prediction is then chosen, even when fewer APs are available. The Naive Bayes algorithm achieved the highest accuracy for the 7AP (99 % accuracy) scenario and for a smaller number of APs. The algorithms based on neural networks had the worst performance. The article proposes future research on the location of Wi-Fi devices indoors.

KEYWORDS: indoor location, machine learning, RSSI, Wi-Fi

1. INTRODUCCIÓN

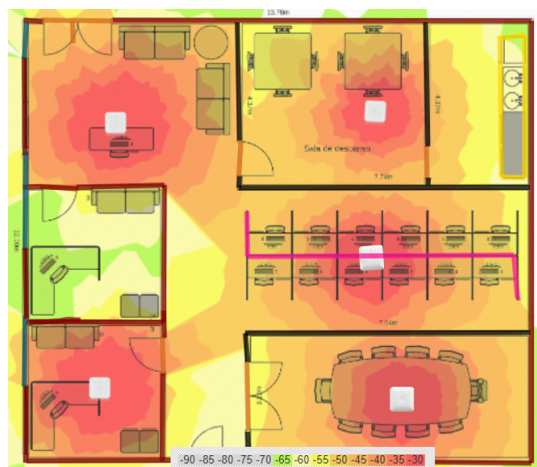
Determinar la ubicación de un objeto o un terminal en una zona de interiores (o exteriores) tiene múltiples aplicaciones, tales como monitoreo de productos, monitoreo de personal, monitoreo de personas en situaciones de emergencia, entre muchas otras. En los últimos años han surgido otras aplicaciones, diferentes del uso de sistemas de posicionamiento satelital, para definir la ubicación de un objeto en interiores usando redes inalámbricas del tipo WPAN (*bluetooth*), WLAN (*wifi*), WMAN (redes móviles 2G/3G/4G/5G, WiMAX) o LPWAN (LoRa WAN, SIGFOX). Dado el amplio uso de las redes wifi, el presente artículo se enfocará en esta tecnología.

Un aspecto clave es el parámetro que se utilizará para realizar la predicción en un escenario basado en redes wifi. La literatura muestra la presencia de dos parámetros relevantes: el RSSI y el CSI (*Channel State Information*). No obstante, como indican Xun et al. (2020), para trabajar con el CSI, se requiere de *hardware* específico y, además, realizar modificaciones en el *firmware* de este. Por tanto, en el presente trabajo se usará el RSSI, el cual puede ser obtenido fácilmente del *smartphone* y de cualquier otro dispositivo que se conecte a una red wifi.

Cuando un terminal (*smartphone*, tableta, *laptop*, *smart TV* o similar) se conecta a una red wifi, estandarizada mediante IEEE 802.11 y sus evoluciones (George & Rao, 2017), específicamente a un punto de acceso (AP), recibe un determinado nivel de intensidad de señal (medido en dBm), denominado RSSI. De acuerdo con los principios de radiopropagación, cuando el terminal se aleja del AP, el RSSI disminuye, y aumenta cuando el terminal se acerca al AP. Si bien el nivel de RSSI en un determinado punto puede ser estimado usando diversas herramientas de predicción (por ejemplo, véase la Figura 1), en la vida real, el valor de RSSI diferirá de la predicción, en la medida en que existen diversos obstáculos que no pueden ser modelados.

Figura 1

Ejemplo de simulación de cobertura de cinco puntos de acceso usando el aplicativo Wi-Fi Designer



Nota. Simulación en la herramienta Wi-Fi Designer, desarrollada por Cambium Networks (<https://wfd.cloud.xirrus.com/wfdc/>).

Para el desarrollo del presente trabajo, se utilizó la herramienta KNIME (*Konstanz Information Miner*), la cual cuenta con licencia Open Source GPLv3, que facilita su libre uso (KNIME, s. f.). Es preciso señalar que esta herramienta ha sido empleada para la elaboración de diversos artículos orientados a *machine learning*, tales como Muenzberg et al. (2019), Çelik y Çinar (2021), Feltrin (2015), y Chauhan y Sehgal (2018). Sin embargo, en la revisión bibliográfica realizada, no se ha encontrado el uso de KNIME para determinar la ubicación de un *smartphone* conectado a la red wifi. Por tanto, dada la facilidad de uso de KNIME, los interesados en aprender sobre algoritmos de *machine learning* y que, además, cuenten con algunos conocimientos de telecomunicaciones podrán tomar el presente artículo como un punto de partida para una mejor comprensión del comportamiento de los algoritmos.

Para tal efecto, aprovechando la versatilidad de KNIME, se propone usar y comparar la precisión de diversos algoritmos de *machine learning* supervisados, para determinar la ubicación de un terminal conectado a la red wifi. En específico, se usarán ocho algoritmos: *decisión tree* (DT), *naive Bayes* (NB), *k-nearest neighbors* (k-NN), *random forest* (RF), *support vector machine* (SVM), *gradient boosted* (GB), así como algoritmos basados en redes neuronales: RProp MLP (*multilayer perceptron*) y PNN (*probabilistic neural network*). Con el fin de profundizar en los algoritmos utilizados, se sugiere revisar KNIME (2020).

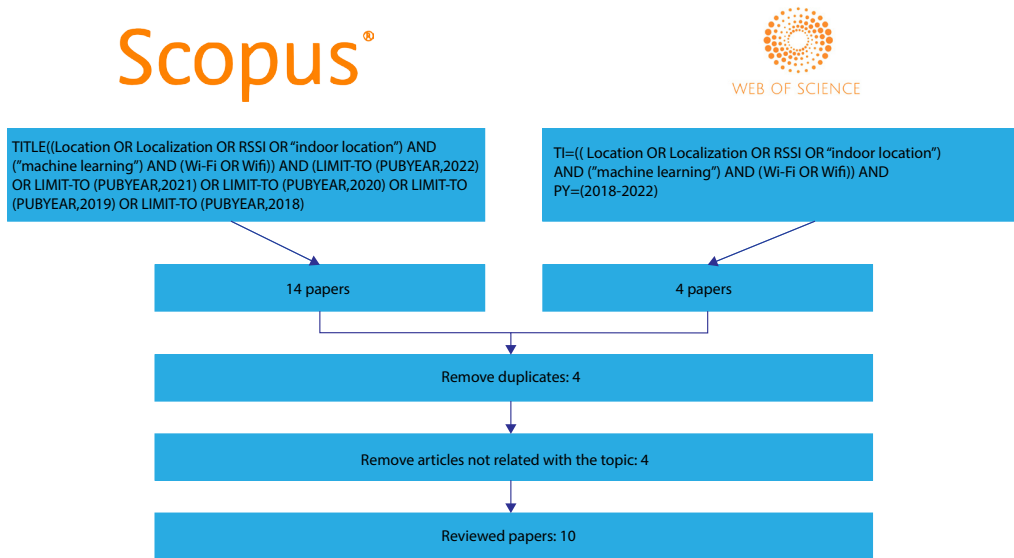
El presente artículo parte de describir los trabajos realizados sobre el tema en los últimos cinco años. Luego se presenta la metodología utilizada, partiendo de la obtención de los datos, seguido por el uso de los algoritmos, hasta implementar el modelo en KNIME. Finalmente, se describen los resultados, así como las conclusiones y las recomendaciones para los trabajos futuros.

2. ESTADO DEL ARTE

Se han encontrado diversos artículos que tienen como objetivo determinar la ubicación de un objeto (que sea compatible con wifi) usando la información de diversos puntos de acceso (AP), a los que dicho objeto tiene alcance. En efecto, Bellavista-Parent et al. (2021) contabilizaron y revisaron un total de 64 artículos entre el 2019 y el 2021, que usan diversos datasets y que de una u otra forma están orientados a determinar el posicionamiento *indoor* utilizando las redes wifi y que a la vez hacen uso de algoritmos de *machine learning*.

No obstante, para el presente artículo se realizó una nueva búsqueda en los portales especializados Scopus y Web of Science (WoS), tomando como punto de partida los títulos de los trabajos (dejando de lado el contenido del *abstract* o de las palabras clave). Se encontró un total de 18 artículos cuyos títulos guardan relación con la búsqueda; sin embargo, finalmente se seleccionaron 10 artículos que guardan estricta relación con posicionamiento vía wifi usando algoritmos de *machine learning* y que han sido publicados en los últimos cinco años (véase la Figura 2).

Figura 2
Artículos revisados en Scopus y Web of Science



Para determinar la ubicación de un terminal en cuatro habitaciones, Sabanci et al. (2018) compararon seis algoritmos utilizando redes neuronales artificiales (ANN), k-NN, DT, NB, *extreme learning machine* (ELM) y SVM, y encontraron que el algoritmo k-NN presenta el mejor rendimiento. Por su lado, Tahat et al. (2021) usaron un dataset que contiene información en las bandas de 2,4 GHz y 5 GHz. Emplearon los algoritmos k-NN, SVM, DT y ANN, y hallaron que el algoritmo basado en redes neuronales presenta el mejor resultado (no obstante, señalan que es necesario un mayor tiempo de entrenamiento).

Se han encontrado investigaciones que usan el dataset UJIndoorLOc, el cual contiene mediciones en tres edificios con diversos pisos (3 a 5). En el estudio de Wadhwa et al. (2019), para la detección del edificio se tiene una precisión del 100 % usando k-NN, mientras que para la predicción del piso se logran precisiones de hasta 97,98 % usando el algoritmo RF. Asimismo, en el estudio de Cao et al. (2021), se emplean los algoritmos *eXtreme gradient boosting* (XGBoost) y *gradient boosting decision tree* (GBDT), y se encontró que para la detección del piso se tienen precisiones de entre 99,54 % a 99,22 %. Por su parte, Maaloul et al. (2022) usan los algoritmos NB, SVM, k-NN, RF y GB, entre los cuales el algoritmo GB presenta la mayor precisión.

Un nuevo enfoque se presenta en el estudio de Xue et al. (2020), al que denominan *localización indoor de alta adaptabilidad* (HAIL), para lo cual se diseña una red neuronal de

retropropagación (BPNN). Los autores indican que queda como trabajo pendiente continuar investigando nuevos algoritmos para mejorar la precisión.

Finalmente, Singh et al. (2021) realizaron una revisión de los artículos publicados entre los años 2007 al 2020 sobre el uso de algoritmos de *machine learning* para determinar la ubicación *indoor*. Asimismo, considerando que se vive en un mundo conectado, se plantean cuestiones sobre la privacidad en la recolección de información.

Tabla 1

Aspectos clave de los artículos revisados

Título	Año	Puntos clave
A Universal Wi-Fi Fingerprint Localization Method Based on Machine Learning and Sample Differences	2021	Uso de algoritmos XGBoost y GBDT. En la detección del piso se logra una precisión de entre 99,54 % a 99,22 %.
An Empirical Evaluation of Machine Learning Algorithms for Indoor Localization Using Dual-Band WiFi	2021	Uso de banda de 2,4 y 5 GHz. Los algoritmos basados en ANN presentan los mejores resultados.
Experimental Machine Learning for RSSI Fingerprint in Indoor WiFi Localization	2021	Uso de tres <i>routers</i> en una habitación.
Using Machine Learning Techniques and Wi-Fi Signal Strength for Determining Indoor User Location	2021	Uso de los algoritmos KNN y SVM. Para el caso de KNN se logra una precisión de 93 %.
Machine Learning for Location Prediction Using RSSI on Wi-Fi 2.4 GHz Frequency Band	2021	Se logra una precisión de 96 %.
Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview	2021	Realiza una revisión de la literatura desde el 2007 al 2020. Plantea cuestiones sobre la privacidad en la recolección de información.
A WiFi Fingerprint Based High-adaptability Indoor Localization Via Machine Learning	2020	Uso de algoritmo HAIL basado en redes neuronales.
Machine Learning Based Indoor Localization Using Wi-Fi Fingerprinting	2019	Base de datos UJIndoorLOc: tres edificios. Uso de dos algoritmos: k-NN y RF.
WiFi Based Indoor Localization: Application and Comparison of Machine Learning Algorithms	2018	Base de datos de UCI: cuatro habitaciones. Uso de seis algoritmos. Mejor precisión: k-NN (98,75 %).

Así, con base en las investigaciones previas, el aporte del presente trabajo de investigación es evaluar el comportamiento de los distintos algoritmos ante cambios en las condiciones del

experimento (reducción del número de AP utilizados). Así se determinará que, pese a que algunos AP dejen de funcionar, el algoritmo elegido siempre será el que presente la mayor precisión. Además, se plantean trabajos futuros para continuar investigando sobre el tema.

3. METODOLOGÍA

Para la comparación de los algoritmos, se usó el dataset denominado *Wireless Indoor Localization*, disponible en la página web de la Universidad de California en Irvine (UCI, 2017). El dataset contiene 2000 mediciones de intensidad de señal de recepción en un *smartphone*, distribuidas en 4 habitaciones con 500 muestras cada una. Para ello, el *smartphone* se desplazó por 4 habitaciones, recogiendo las mediciones del RSSI de 7 AP. Cabe señalar que no se tiene mayor detalle del dataset, por ejemplo, banda de frecuencia, modelo del terminal móvil, canales utilizados, entre otros. Para más información, un extracto del dataset se presenta en la Tabla 2.

Tabla 2

Extracto del dataset utilizado

AP_01	AP_02	AP_03	AP_04	AP_05	AP_06	AP_07	Room
-64	-56	-61	-66	-71	-82	-81	Room_1
-68	-57	-61	-65	-71	-85	-85	Room_1
-63	-60	-60	-67	-76	-85	-84	Room_1
-34	-53	-52	-19	-65	-75	-69	Room_2

Nota. Reproducido de *Wireless Indoor Localization*, by Rajen Bhatt [Dataset], por Universidad de California en Irvine, 2017, Machine Learning Repository (<https://archive-beta.ics.uci.edu/ml/datasets/wireless+indoor+localization>).

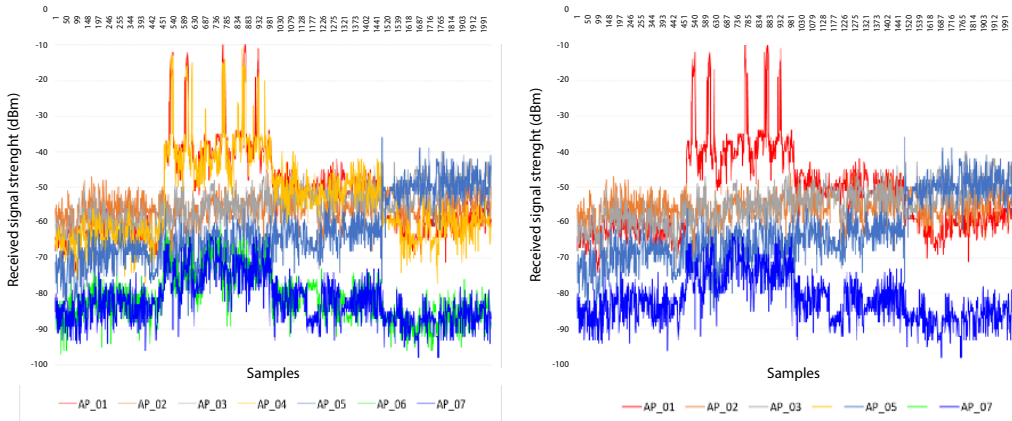
Es preciso señalar que, según el autor del dataset, la información se utilizó en Jayant et al. (2017) aplicando *Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm* (FPSOGSA); no obstante, al revisar dicho artículo, se observa que solo usa tres habitaciones. Asimismo, tal como se señaló previamente, Sabanci et al. (2018) también usan este dataset, entrenando la data con seis algoritmos de *machine learning*.

El dataset, que contiene mediciones de diversos AP, se dividirá en tres tipos. En primer lugar, se usará el dataset original (con 7 AP), luego con 6 AP y, finalmente, solo considerando 5 AP. Esto se realiza con el fin de validar el comportamiento del algoritmo ante fallas de un determinado AP. En la Figura 3, se observa que los niveles de RSSI oscilan entre -10 dBm (nivel de señal muy bueno) y -98 dBm (nivel de señal muy malo). Luego la información de cada uno de ellos se normalizará en el rango de 0 a 100, con el fin de poder utilizar todos los algoritmos indicados en la sección previa. Después, se particionará en una proporción de 70 %

a 30 %; el 70 % servirá para realizar el entrenamiento del algoritmo y el 30 % para calcular la precisión del algoritmo. Esta información se ingresará a los bloques que contienen los diversos algoritmos y se elegirá el que arroje como resultado el menor error (mayor precisión).

Figura 3

Niveles de intensidad de señal recibidos en el smartphone usando datos de 7 y 5 AP

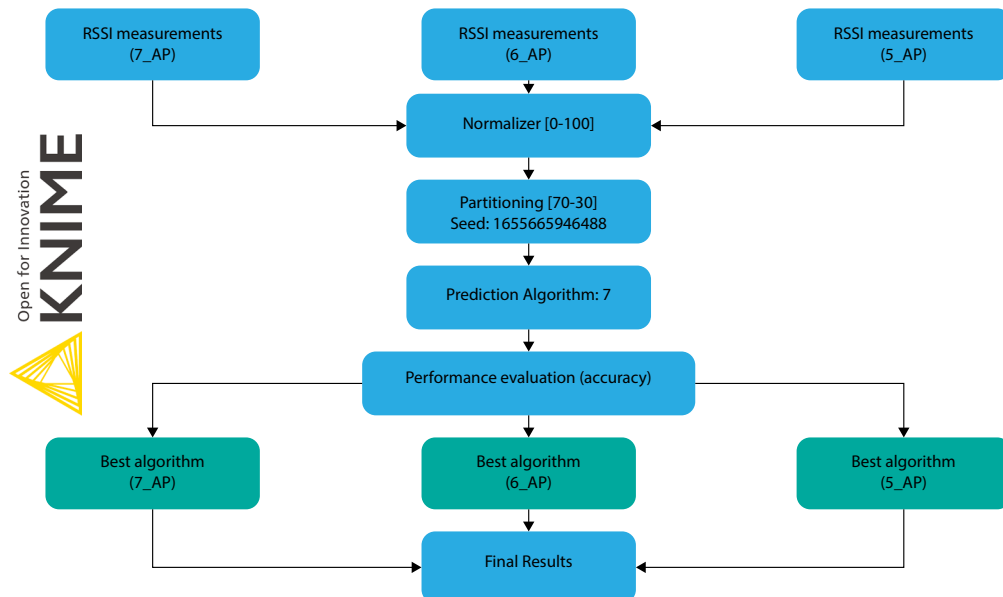


Nota. Elaboración propia usando el dataset de UCI (2017).

En la Figura 4, se muestra en forma de esquema el modelo utilizado en el presente artículo, partiendo desde la formación de tres grupos de mediciones (datos de ingreso del modelo) hasta llegar al algoritmo que presente el menor error de predicción (datos de salida del modelo).

Figura 4

Modelo utilizado para evaluar la precisión de los algoritmos

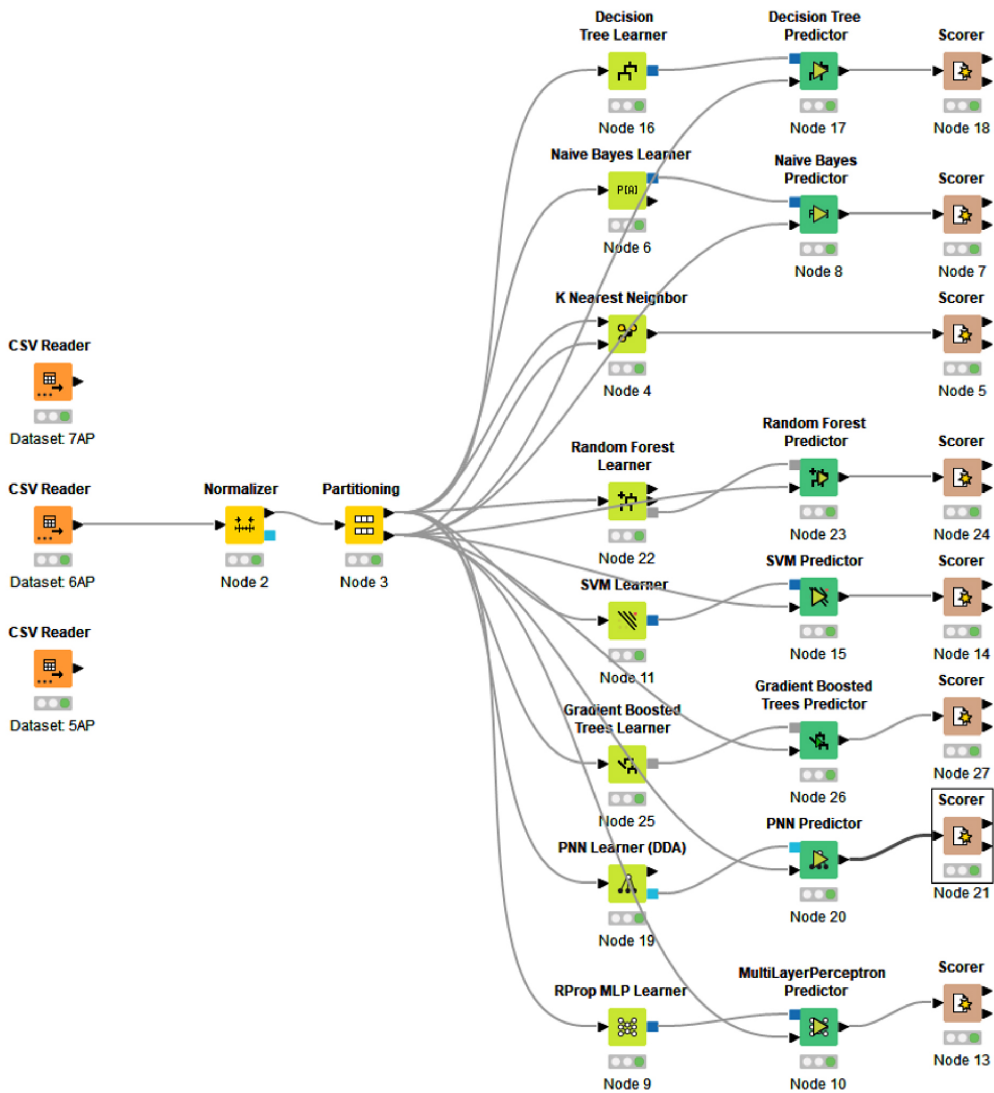


4. IMPLEMENTACIÓN DE LOS ALGORITMOS DE MACHINE LEARNING EN KNIME

Los tres datasets se cargan de manera independiente en el flujo de trabajo (*workflow*) elaborado en KNIME (véase la Figura 5). KNIME trabaja con bloques, en los cuales cada uno cumple una determinada función. El dataset ingresa al flujo de trabajo por medio del módulo “CSV Reader” y los resultados se obtienen en el bloque “Scorer”.

Figura 5

Flujo de trabajo implementado en KNIME con ocho algoritmos de machine learning



Es importante resaltar que, previo a realizar el particionamiento, el dataset debe ser normalizado (en este caso, de 0 a 100). Luego se realiza el particionado (70/30) de tal forma que cada habitación tenga la misma proporción del número de muestras. Asimismo, es preciso resaltar que, con el fin de que los resultados puedan ser replicados, en el bloque de particionado de KNIME se usó la siguiente semilla: 1655665946488.

El *workflow* contempla el uso de ocho algoritmos no supervisados aplicados a un problema de clasificación, en específico, para detectar la habitación en la que se encuentra un terminal móvil. Así se implementan los algoritmos DT, NB, k-NN, RF, SVM, GB, así como algoritmos basados en redes neuronales artificiales (ANN), en particular RProp MLP y PNN.

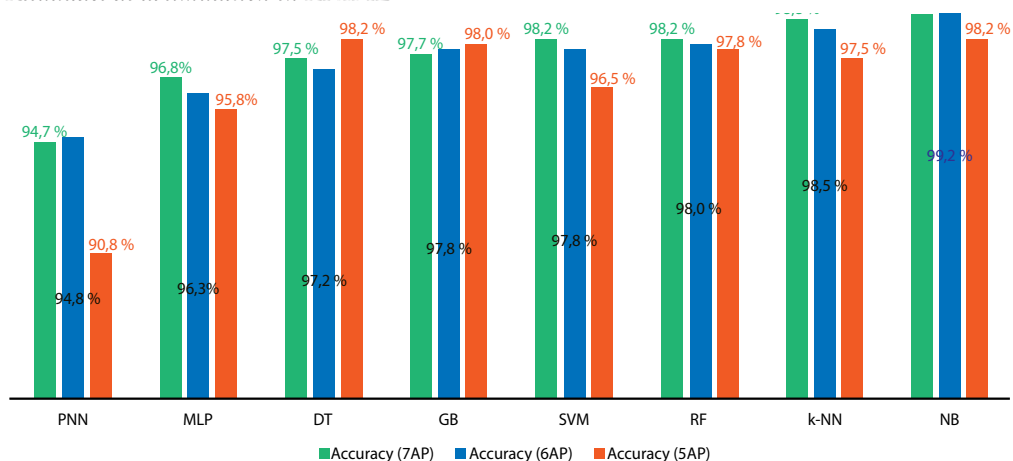
Dada la facilidad de uso de KNIME, cada algoritmo de *machine learning* se carga como un bloque independiente. Posteriormente, la precisión de cada algoritmo se obtiene en el módulo “Scorer” de KNIME, con base en la matriz de confusión. Para el caso del algoritmo MLP, se usan los siguientes parámetros: (i) número máximo de iteraciones: 1000; (ii) número de capas ocultas: 2; y (iii) número de neuronas ocultas por capa: 10. Asimismo, para el algoritmo PNN se usan 2000 épocas. En el caso de los demás algoritmos, se emplea la configuración predeterminada de cada módulo.

5. RESULTADOS

Se observa que, usando la herramienta KNIME, el algoritmo *naive Bayes* presenta la mejor precisión: 99,00 % usando 7 AP y 98,167 % usando 5 AP. En segundo lugar, queda el algoritmo *k-nearest neighbors*. También se observa que los algoritmos basados en ANN, tales como el MLP y el PNN, presentan la precisión más baja. En efecto, con MPL se logra una precisión de 96,833 % y para PNN tan solo se llega a 94,667 % (para el caso de 7 AP). Todos los resultados de las diversas simulaciones se visualizan en la Figura 6.

Figura 6

Resultados de la simulación en KNIME



Es preciso indicar que el presente análisis de clasificación tiene la particularidad de ser multiclase, es decir que la variable que se predice no es binaria (por ejemplo, SÍ/NO o 1/0), sino que más bien se trata de cuatro valores (Room_1, Room_2, Room_3 y Room_4).

6. CONCLUSIONES, TRABAJOS FUTUROS Y LIMITACIONES

El algoritmo *naive Bayes* presenta la precisión más alta, pues llega hasta un 99 % de precisión, seguido por el algoritmo k-NN con 98,833 % (para el caso de 7 AP). Asimismo, cuando el sistema es forzado a trabajar con un número menor de AP, se tiene una precisión del 98,167 % para NB. Además, se observa que los algoritmos basados en ANN (en específico, MLP y PNN) presentan el peor rendimiento. En efecto, para el caso de 7 AP, con MLP se tiene una precisión del 96,833 % y con PNN solo se llega a 94,667 %.

Diversos trabajos futuros se desprenden de la presente investigación. Por el lado de las redes wifi, resulta necesario crear un dataset propio que contenga mediciones de todas las bandas de espectro utilizadas para redes wifi: 2,4 GHz, 5 GHz y 6 GHz, de tal forma que se pueda determinar cuál es la banda que permite tener una mejor predicción. Asimismo, por el lado de las redes móviles, también se puede hacer pruebas georreferenciadas (con información de latitud y longitud) de *drive test* o *walk test* para identificar los Cell_ID vecinos, y sobre esa base predecir la ubicación de un terminal móvil usando algoritmos de *machine learning*.

Finalmente, los resultados se deben tomar con mucha precaución en la medida que cada ambiente interior es único y, por tanto, para que el sistema de localización funcione correctamente, se requiere que previamente se haya realizado una adecuada recolección de datos. Es decir, para otro escenario, los resultados podrían variar, lo cual evidentemente es una restricción de los trabajos que usan algoritmos para localizar dispositivos sin usar sistemas de posicionamiento satelital.

REFERENCIAS

- AlQahtani, A. A. S., & Choudhury, N. (2021). Machine learning for location prediction using rssi on Wi-Fi 2.4 GHz frequency band. En *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0336-0342). DOI: 10.1109/IEMCON53756.2021.9623104
- Bellavista-Parent, V., Torres-Sospedra, J., & Perez-Navarro, A. (2021). New trends in indoor positioning based on WiFi and machine learning: A systematic review. En *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (pp. 1-8). <https://doi.org/10.1109/IPIN51156.2021.9662521>

- Cao, X., Zhuang, Y., Yang, X., Sun, X., & Wang, X. (2021). A universal Wi-Fi fingerprint localization method based on machine learning and sample differences. *Satellite Navigation*, 2, 27. <https://doi.org/10.1186/s43020-021-00058-8>
- Çelik, H., & Çinar, A. (2021). An application on ensemble learning using KNIME. En *2021 International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 400-403). <https://doi.org/10.1109/ICDABI53623.2021.9655815>
- Chauhan, C., & Sehgal, S. (2018). Sentiment classification for mobile reviews using KNIME. En *2018 International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 548-553). <https://doi.org/10.1109/GUCON.2018.8674946>
- Feltrin, L. (2015). KNIME an open source solution for predictive analytics in the geosciences [Software and datasets]. *IEEE Geoscience and Remote Sensing Magazine*, 3(4), 28-38. <https://doi.org/10.1109/MGRS.2015.2496160>
- George, D., & Rao, S. (2017). Enabling rural connectivity: long range Wi-Fi versus super Wi-Fi. En *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)* (pp. 1-4). <https://doi.org/10.1109/ICIC.2017.8524142>
- Insany, G. P., Ayu, M. A., & Mantoro, T. (2021). Using machine learning techniques and Wi-Fi signal strength for determining indoor user location. En *2021 IEEE 7th International Conference on Computing, Engineering and Design (ICCED)*. doi: 10.1109/ICCED53389.2021.9664859
- Jayant G., R., Perumal, B., Narayanan, S., Thakur, P., & Bhatt, R. (2017). User localization in an indoor environment using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks. En *Proceedings of Sixth International Conference on Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing* (vol. 546). https://doi.org/10.1007/978-981-10-3322-3_27
- KNIME. (s. f.). KNIME Open Source Story. <https://www.knime.com/knime-open-source-story>
- KNIME. (2020, 25 de junio). Classification_and_Predictive_Modelling. https://hub.knime.com/knime/spaces/Examples/latest/04_Analytics/04_Classification_and_Predictive_Modelling
- Koovimol, P., & Pattaramalai, S. (2021). Experimental machine learning for RSSI fingerprint in indoor WiFi localization. En *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 1018-1021). DOI: 10.1109/ECTI-CON51831.2021.9454865
- Maaloul, K., Abdelhamid, N. M., & Lejdel, B. (2022). Machine learning based indoor localization using Wi-Fi and smartphone in a shopping malls. En B. Lejdel, E.

- Clementini & L. Alarabi (Eds.), *Artificial Intelligence and Its Applications. AIAP 2021. Lecture Notes in Networks and Systems* (vol. 413). https://doi.org/10.1007/978-3-030-96311-8_1
- Muenzberg, A., Sauer, J., Hein, A., & Roesch, N. (2019). Checking the plausibility of nutrient data in food datasets using KNIME and big data. En *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. <https://doi.org/10.1109/WiMOB.2019.8923233>
- Sabanci, K., Yigit, E., Ustun, D., Toktas, A., & Aslan, M. (2018). WiFi based indoor localization: Application and comparison of machine learning algorithms. En *2018 XXIIIrd International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED)* (pp. 246-251). <https://doi.org/10.1109/DIPED.2018.8543125>
- Singh, N., Choe, S., & Punmiya, R. (2021). Machine learning based indoor localization using Wi-Fi RSSI fingerprints: An overview. *IEEE Access*, 9, 127150-127174. <https://doi.org/10.1109/ACCESS.2021.3111083>
- Tahat, A., Awwad, R., Baydoun, N., Al-Nabih, S., & Edwan, T. A. (2021). An empirical evaluation of machine learning algorithms for indoor localization using dual-band WiFi. En *2021 2nd European Symposium on Software Engineering* (pp. 106-111). <https://doi.org/10.1145/3501774.3501790>
- Universidad de California en Irvine. (2017, 4 de diciembre). *Wireless Indoor Localization, by Rajen Bhatt* [Dataset]. Machine Learning Repository. <https://archive-beta.ics.uci.edu/ml/datasets/wireless+indoor+localization>
- Wadhwa, S., Rai, P., & Kaushik, R. (2019). Machine learning based indoor localization using wi-fi fingerprinting. *International Journal of Recent Technology and Engineering*, 8(3), 502-506. <https://doi.10.35940/ijrte.A2133.098319>
- Xue, J., Liu, J., Sheng, M., Shi, Y., & Li, J. (2020). A WiFi fingerprint based high-adaptability indoor localization via machine learning. *China Communications*, 17(7), 247-259. <https://doi.org/10.23919/J.CC.2020.07.018>
- Xun, W., Sun, L., Han, C., Lin, Z., & Guo, J. (2020). Depthwise separable convolution based passive indoor localization using CSI Fingerprint. En *2020 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1-6). <https://doi.org/10.1109/WCNC45663.2020.9120638>