



Auspicio Institucional

Estudio de tolerancia a errores en sensores empleados para detección de ocupación de habitantes de un recinto empleando técnicas de aprendizaje automático

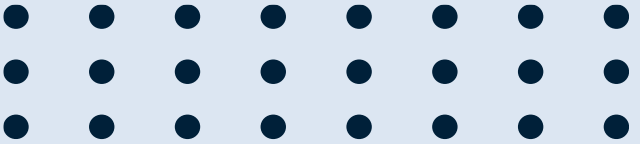
Carlos Alberto Binker, Hugo Tantignone, Lautaro Lasorsa

Guillermo Buranits, Eliseo Zurdo, Maximiliano Frattini

Departamento de Ingeniería e Investigaciones Tecnológicas, Universidad Nacional de La Matanza

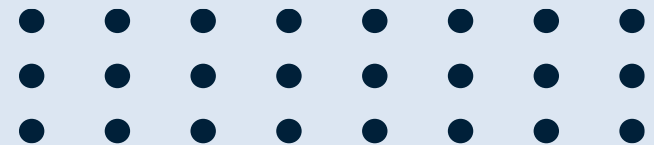
{cbinker, htantignone, laulasorsa, gburanits, eazurdo, mfrattini }@unlam.edu.ar

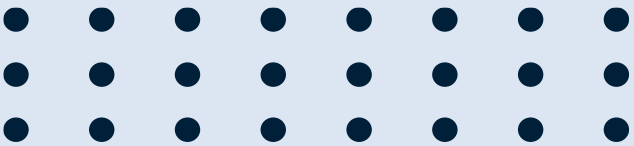




Índice

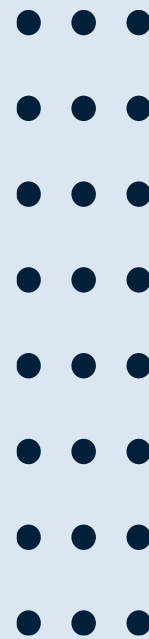
- 1 Introducción
- 2 Escenarios
- 3 Modelos
- 4 Falla total
- 5 Falla oculta
- 6 Conclusiones

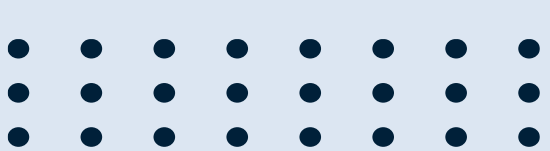




Introducción

La situación estudiada en este trabajo consiste en, dados sensores IoT en una habitación, decidir si hay personas en su interior. El presente trabajo busca estudiar el impacto de fallas en estos sensores usando distintos modelos de aprendizaje automático.

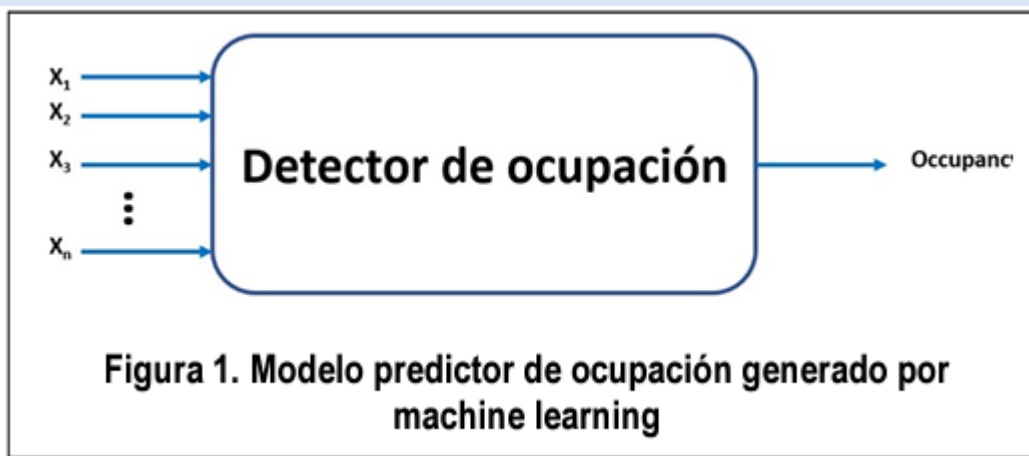




Escenarios propuestos

Se estudiarán dos tipos de situaciones:

1. Un subconjunto de los sensores falla de forma total, y se utilizan modelos entrenados únicamente con los sensores restantes.
2. Un subconjunto de los sensores tiene un ruido en las mediciones que es desconocido para los modelos, y se usa el modelo para el caso sin fallas.



Esto abre dos variables:

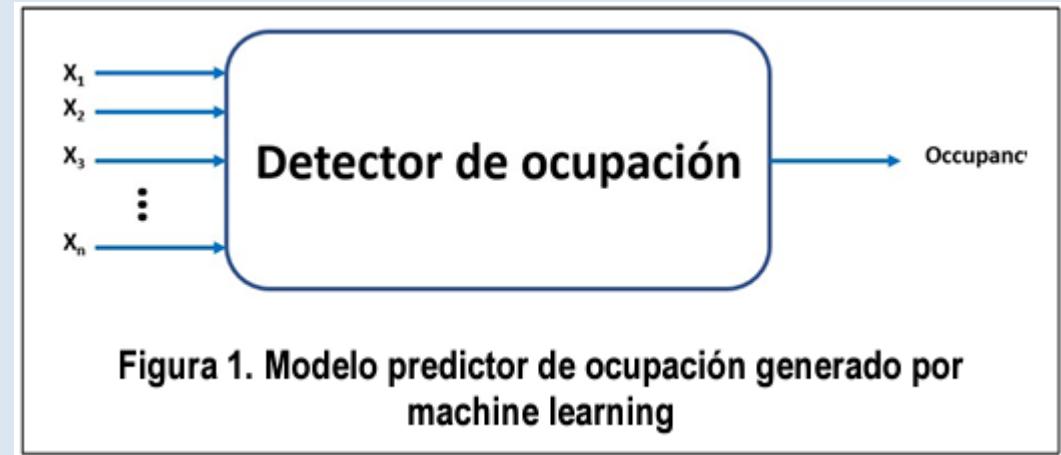
1. La probabilidad de que un subconjunto de sensores falle.
2. En el escenario 2, la magnitud del ruido introducido en el sensor.



Modelos

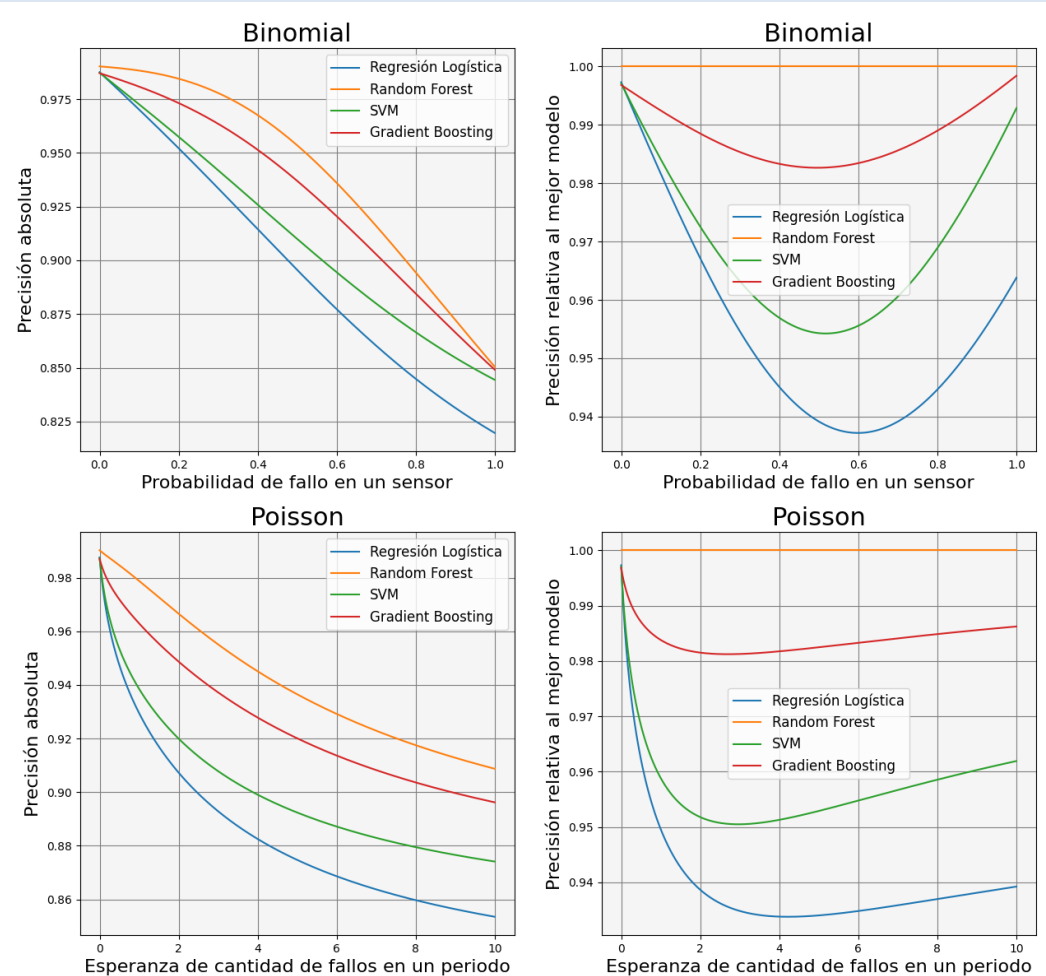
En este trabajo se utilizan 4 modelos:

1. Regresión Logística (RL)
2. Random Forest (RF)
3. Support Vector Machine (SVM)
4. Gradient Boosting (GB)



Falla total

En estos escenarios un subconjunto de los sensores falla de forma total. Se evaluó a cada modelo promediando su desempeño en cada posible subconjunto propio de sensores fallados, ponderado por la probabilidad de cada subconjunto.



Se utilizaron dos distribuciones para modelar la probabilidad de fallo de un subconjunto de sensores:

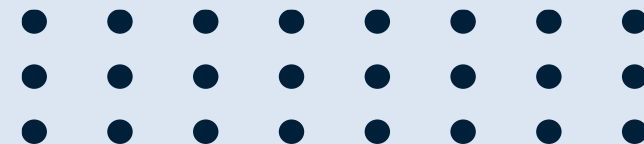
1. Binomial(n, p)
2. Poisson(k, p)

Donde el valor p parametriza ambas distribuciones, y es el eje X de los gráficos. El eje Y es la precisión promedio ponderada de cada modelo.

En la primera columna los gráficos muestran la precisión en valores absolutos, y en la segunda relativa al mejor modelo para ese valor del parámetro.

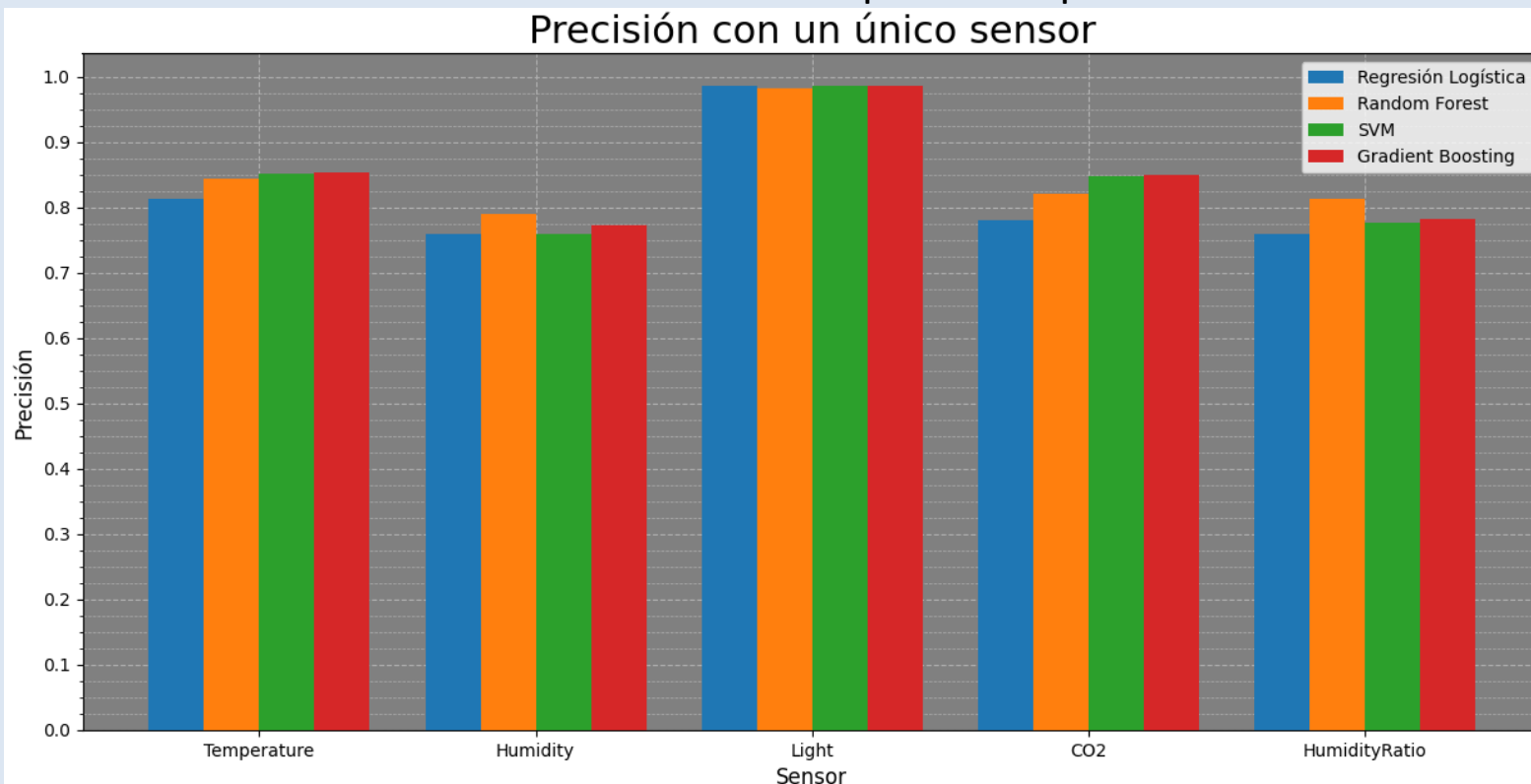
Se puede apreciar siempre el mismo orden:

RF > GB > SVM > RL



Falla total

Que incluso con una probabilidad de falla significativamente alta se obtengan buenos valores de precisión es algo llamativo. Por eso es interesante observar la precisión que tienen los modelos utilizando un único sensor.

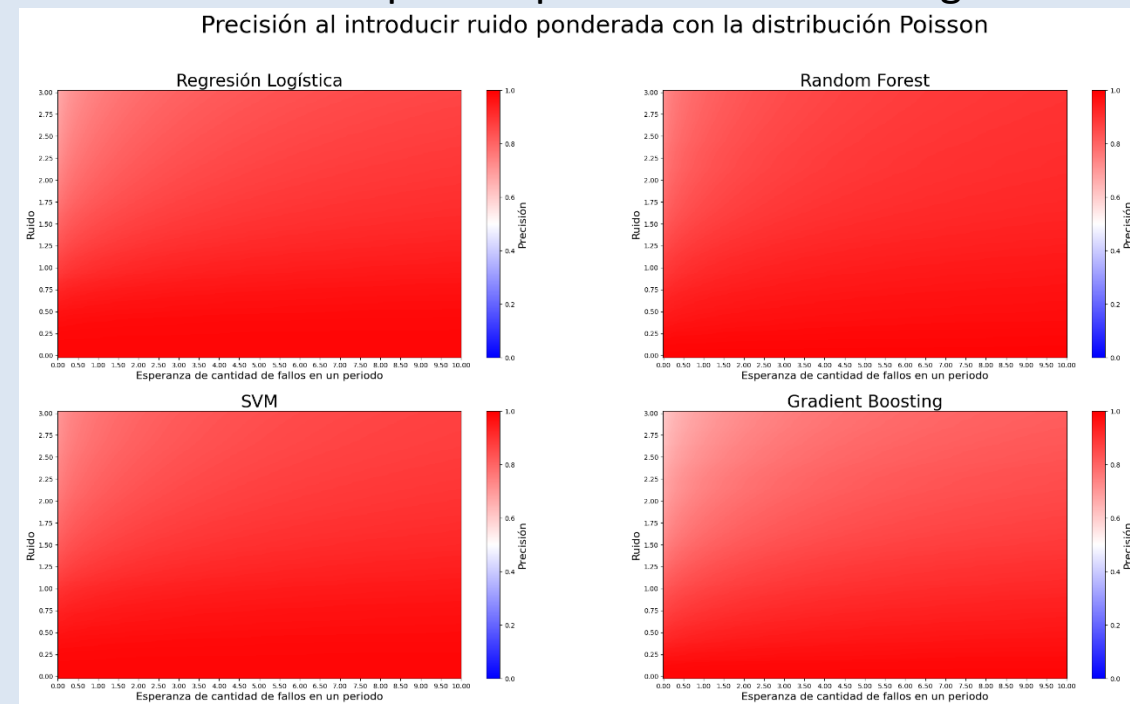
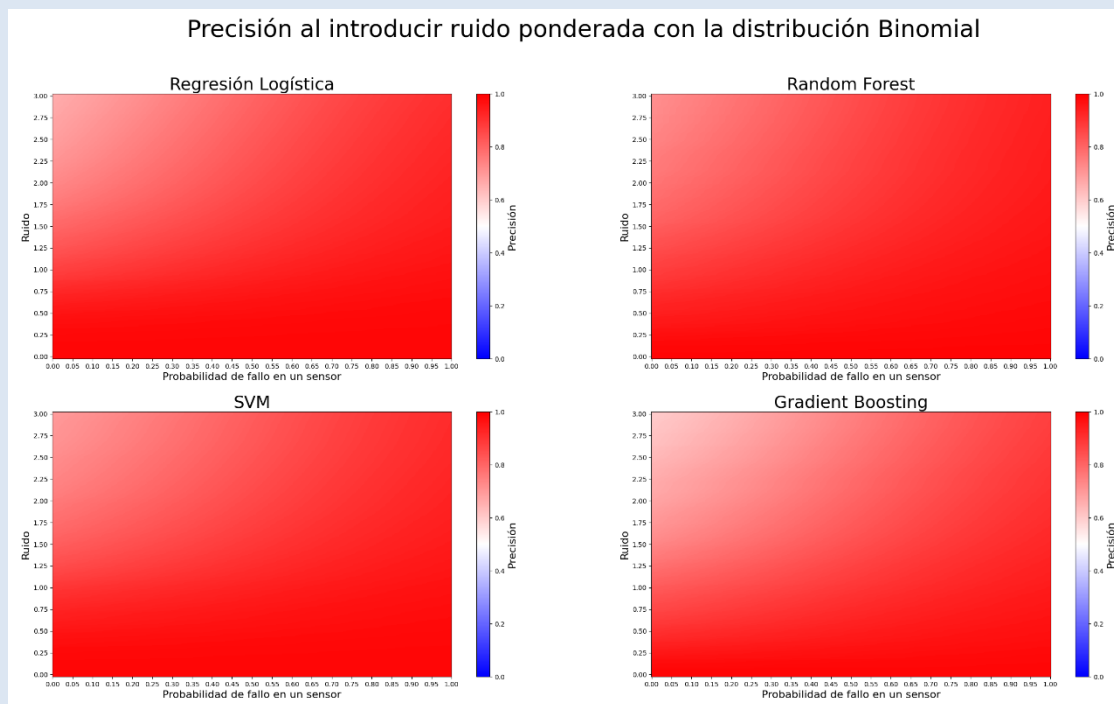


Se puede ver que con solo utilizar el sensor de Luz todos los modelos tienen una precisión de casi el 100%.

Además, los demás sensores permiten de forma individual valores de precisión mayores al 70%.

Falla oculta

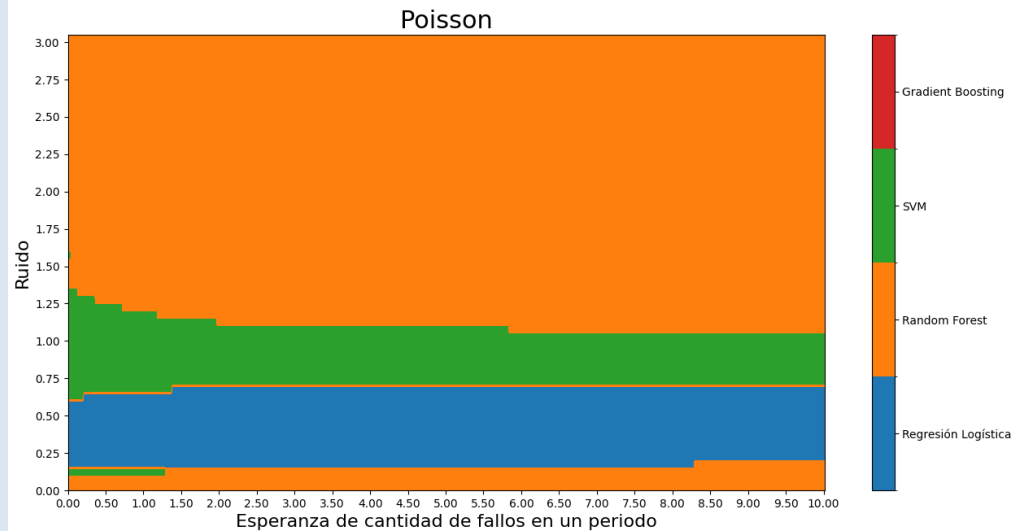
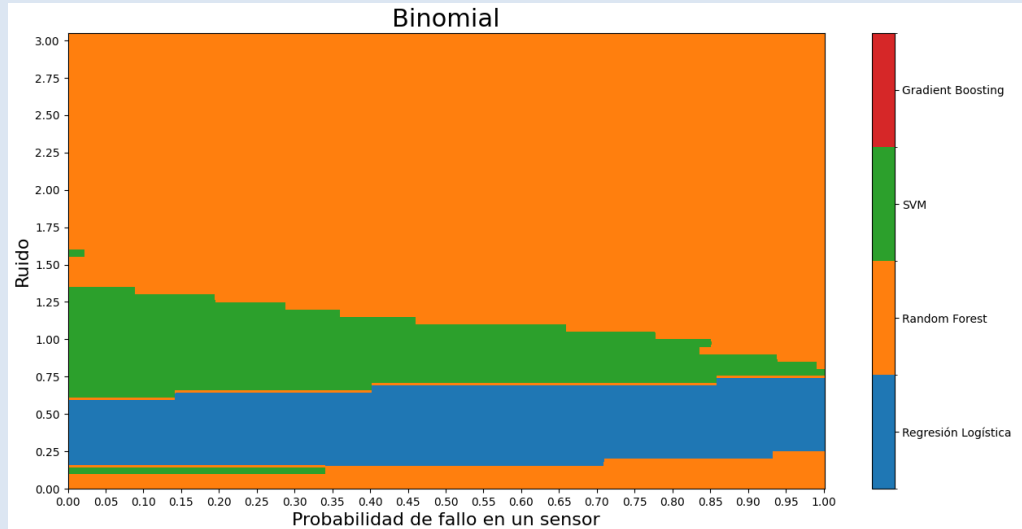
El otro posible escenario es que, al subconjunto con fallos, en lugar de inhabilitarlo, se le introduce un error de una distribución normal con la varianza del sensor en los datos del entrenamiento multiplicada por la variable de magnitud.



En estas figuras, cada una utiliza un tipo de distribución distinta, y cada gráfico de cada figura corresponde a un modelo. En estos, el eje X es el parámetro que controla la probabilidad de fallo en los sensores y el Y la intensidad del error introducido. El color indica la precisión promedio ponderada, que como se ve en todos los casos es alta.

Falla oculta

Visto lo anterior es razonable preguntarnos cuál modelo es mejor para cada caso



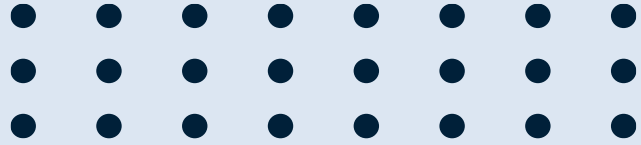
Se ve que para la mayoría del área el Random Forest es superior a los demás modelos, pero es interesante notar que hay valores de los parámetros para los cuales es óptimo utilizar SVM o Regresión Logística.

El modelo que nunca es superior es Gradient Boosting.



Conclusiones

Se pueden extraer dos conclusiones principales de este trabajo:

- Random Forest es el modelo con mejor desempeño en el caso estudiado.
 - Los sensores disponibles contienen mucha información sobre la variable a predecir incluso de forma individual.
- 



Comentarios adicionales

Por ser un framework ligero se utilizó Sklearn para la realización de este trabajo.

Todo el trabajo se encuentra disponible en GitHub:

<https://github.com/carlucho1/CONAIISI-2024-2>





iGracias!

Contacto:

- laulasorsa@unlam.edu.ar | lautarolasorsa@gmail.com
- lautaro-lasorsa @ LinkedIn