

# Eje Temático: Aplicaciones Informáticas y de Sistemas de Información

12°  
CONGRESO  
NACIONAL  
CoNaII SI  
2024



## Detección de fallas en sensores IoT mediante Machine Learning

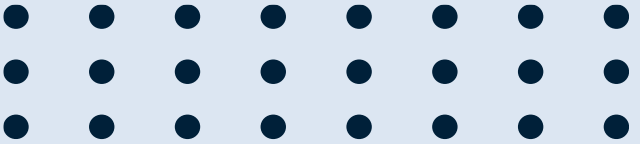
Carlos Alberto Binker, Hugo Tantignone, Lautaro Lasorsa

Guillermo Buranits, Eliseo Zurdo, Maximiliano Frattini

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

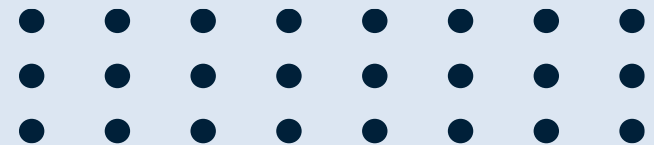
{cbinker, htantignone, laulasorsa, gburanits, eazurdo, mfrattini }@unlam.edu.ar

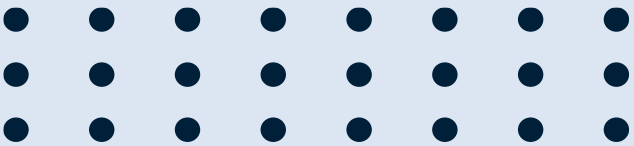




# Índice

- 1 Introducción
- 2 Escenarios propuestos
- 3 Modelos bicategoricos
- 4 Modelos multicategoricos
- 5 Conclusiones





# Introducción

En los dispositivos IoT es los sensores pueden medir de forma incorrecta los datos. Esto lleva a dos necesidades:

1. La robustez del sistema frente a los errores en los datos.
2. La capacidad de detectar estos fallos.

El presente trabajo busca abordar el segundo punto utilizando técnicas de aprendizaje automático.



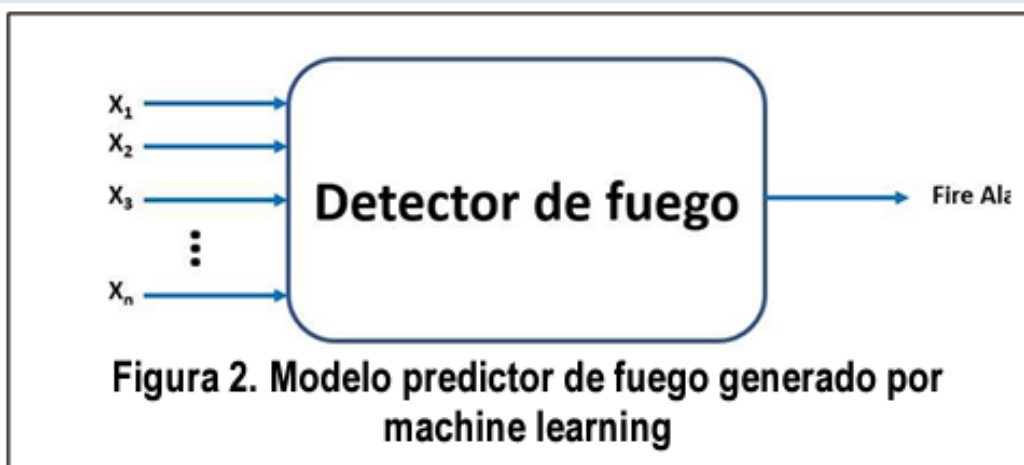
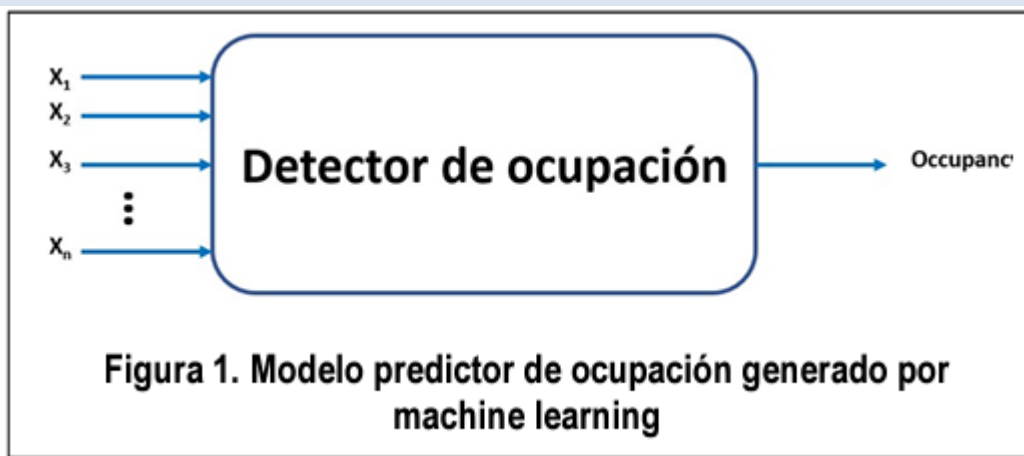
# Escenarios Propuestos

Se estudiaron dos casos de uso con los que habíamos trabajado anteriormente:

- Detección de ocupación en una habitación.
- Detección de fuego.

A su vez, se plantean dos enfoques posibles:

1. Modelos especializados para detectar si un sensor en particular ha fallado o no.
2. Un único modelo que debe decidir si algún sensor fallo y cuál.

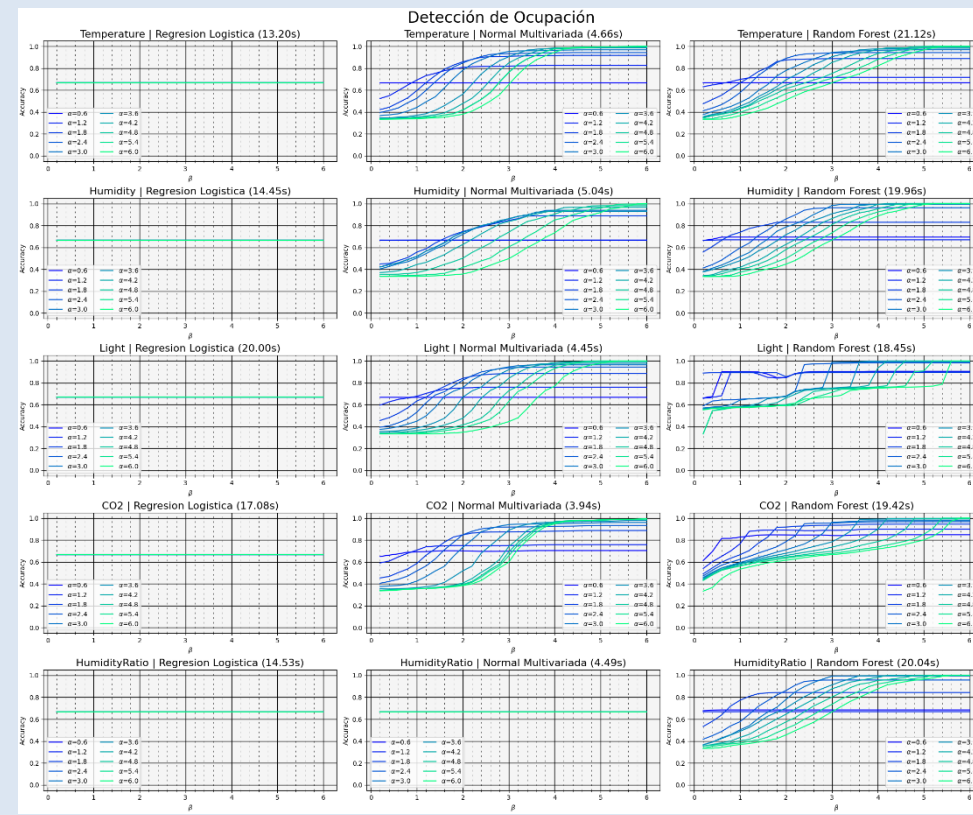
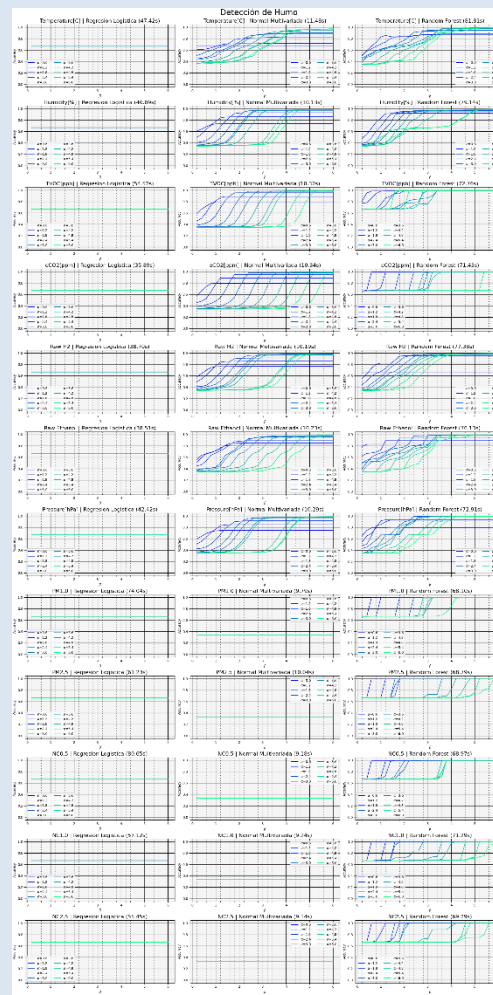


# Modelos Bicategóricos

- Para detectar si un input es consistente se crearon 3 modelos distintos, y a cada uno se lo entreno con los datos
- originales y además datos sintéticos creados en base a los datos originales incorporando ruido en uno de los sensores
- por vez.

En las figuras:

- La columna indica qué modelo se utilizó
- La fila indica en qué sensor se introdujo el error.
- La serie indica la intensidad del error en los datos de entrenamiento.
- El eje X es la intensidad de error en los datos de validación.
- El eje Y es la precisión.

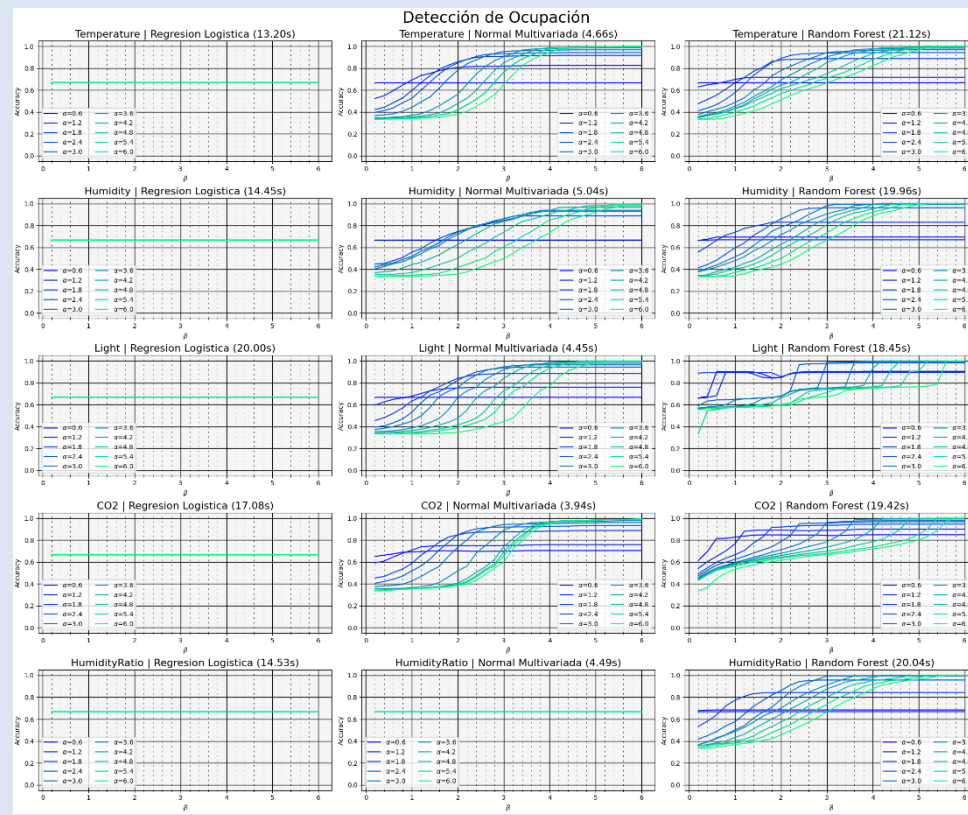
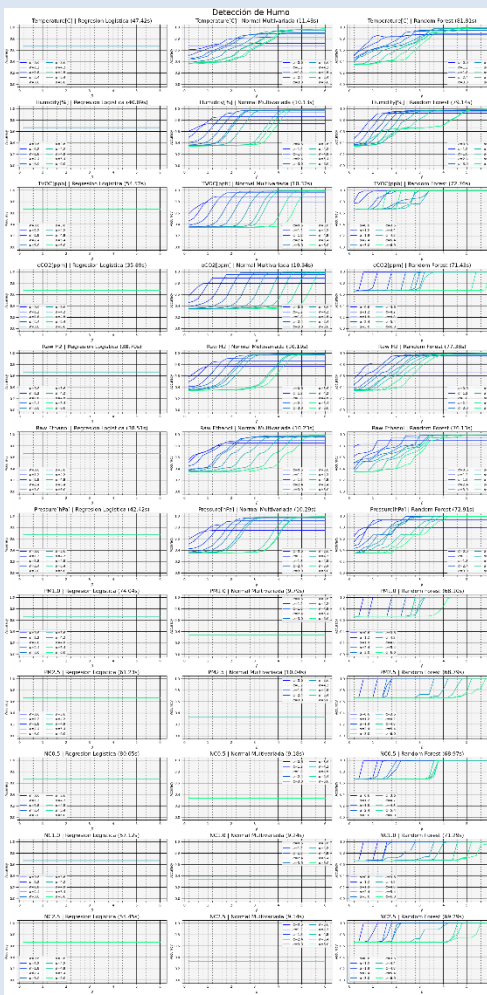


# Modelos Bicategóricos

Los modelos utilizados fueron: Regresión Logística (RL) simple, Normal Multivariada (NM) y Random Forest (RF). Para el caso de Normal Multivariada se decidió un nivel de verosimilitud que maximice la precisión en los datos de entrenamiento.

Se puede apreciar que :

- RL es incapaz de detectar patrones no lineales en los datos, dando un output constante.
- NM y RF son ambos capaces de capturar patrones no lineales y son eficaces.
- NM es ineficaz en algunos sensores, mientras que RF es eficaz en todos los sensores.



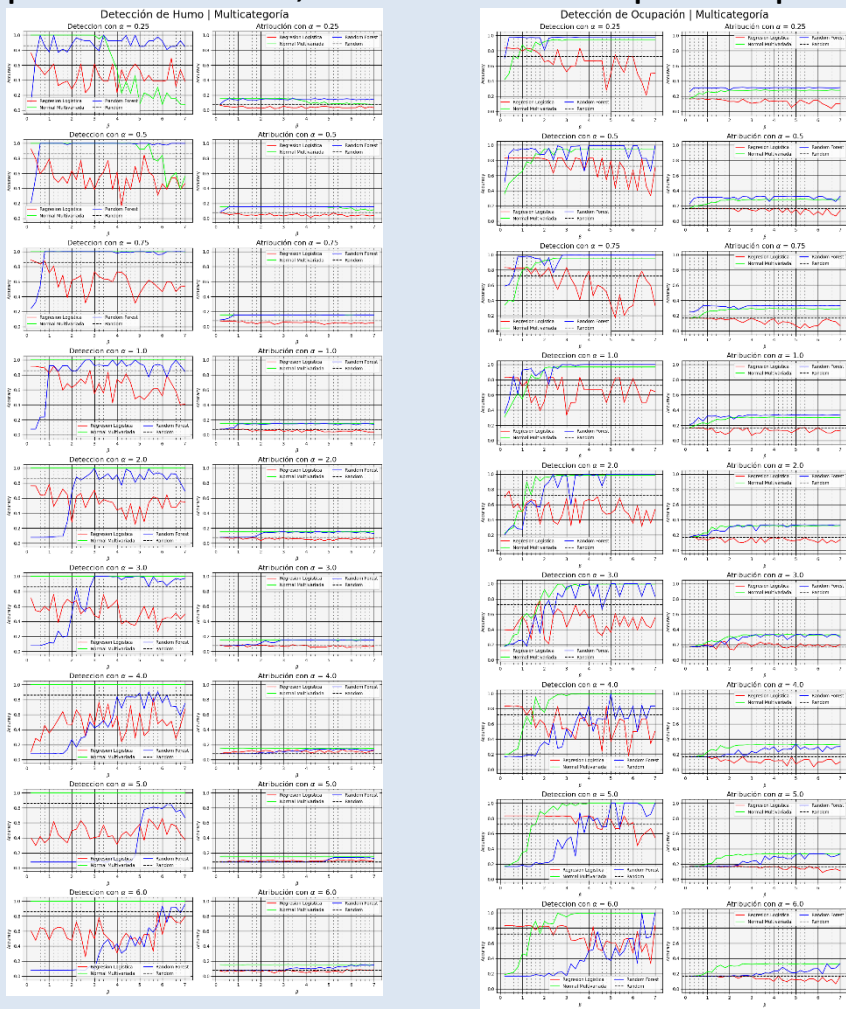


# Modelos Multicategóricos

Se utilizaron los mismos 3 modelos que en el caso bicategorico pero en sus versiones multi categoricas. En el caso puntual de NM, se creó una NM para la presencia de errores en cada sensor y se decide por la de mayor verosimilitud.

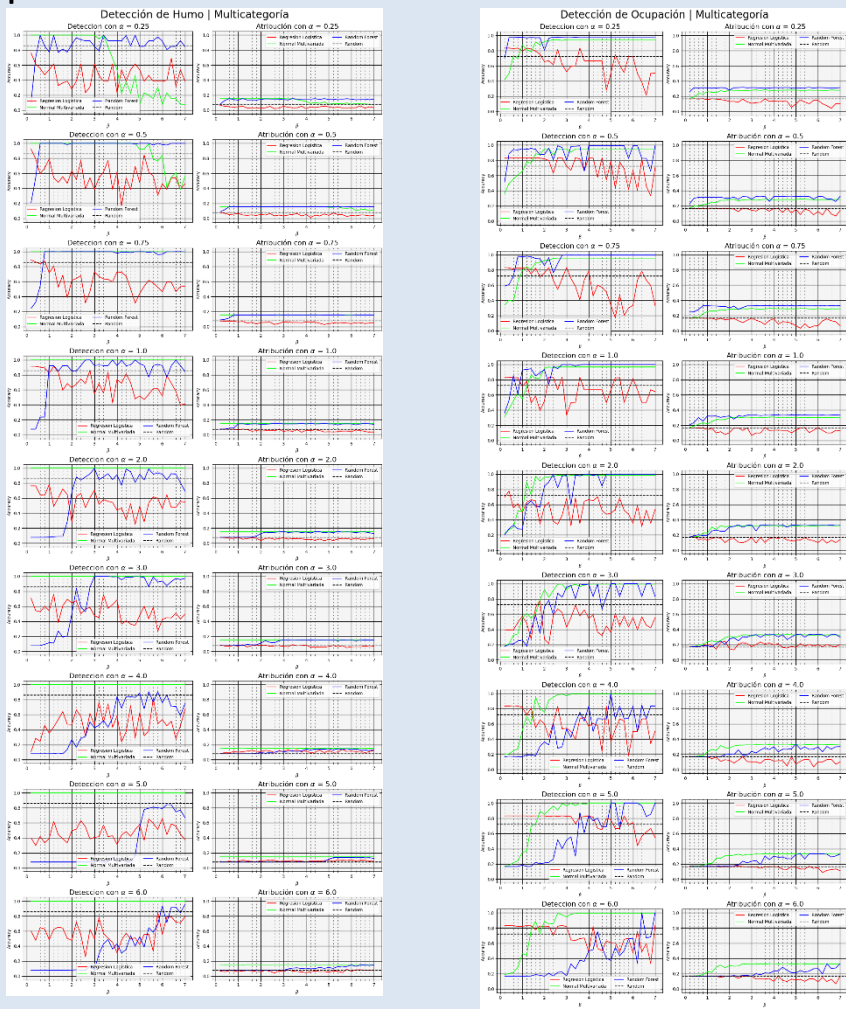
En las figuras:

- Cada fila corresponde a una distinta intensidad del error en los datos de entrenamiento.
- La primera columna es la precisión en distinguir si hay error o no en los datos (es decir, detectar correctamente la clase “sin error”)
- La segunda columna es la precisión detectando la categoría exacta (“sin error” o especificar el sensor en el que hay error)
- Cada serie corresponde a uno de los 3 modelos (RL : rojo, NV : verde , RF : azul) El eje X es la intensidad del error en los datos de validación
- El eje Y es la precisión en los datos de validación



# Modelos Multicategóricos

Lo que se puede apreciar es que en general RL es menos eficaz que los otros modelos tanto para detectar errores como para atribuirlos correctamente.



También podemos ver:

- Incluso cuando se detectan correctamente los errores, se los atribuye de forma generalmente errónea.
- Tomando la magnitud del error apropiada durante los datos de entrenamiento, en un caso RF llega a tener casi detección perfecta y en el otro NM llega a detección perfecta.
- En general tienen performances similares.

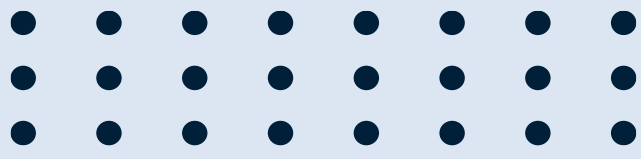




# Conclusiones

Random Forest y Normal Multivariada son dos modelos ligeros de entrenar y de almacenar que sin embargo son sumamente efectivos para la tarea de detección de errores en los datos medidos por los sensores, aunque en su atribución al detector que causo el error.

La regresión logística, si bien también es un modelo ligero, es incapaz de cumplir las tareas satisfactoriamente.

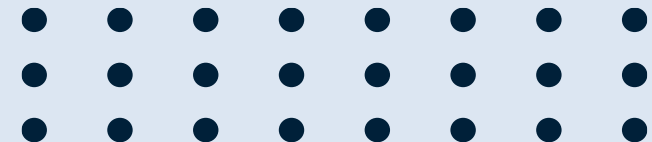




# Comentarios adicionales

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/LautaroLasorsa/CONAIISI-2024>





# iGracias!

Contacto:

- [laulasorsa@unlam.edu.ar](mailto:laulasorsa@unlam.edu.ar) | [lautarolasorsa@gmail.com](mailto:lautarolasorsa@gmail.com)
- lautaro-lasorsa @ LinkedIn