DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

UNIVERSIDAD DE GRANADA

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

ALGORITMOS DE INTELIGENCIA COMPUTACIONAL PARA ABORDAR PROBLEMAS DE DETECCIÓN DE ANOMALÍAS EN ENTORNOS BIG DATA

Memoria presentada por

JACINTO CARRASCO CASTILLO

DIRECTORES

FRANCISCO HERRERA TRIGUERO
JULIÁN LUENGO MARTÍN

Granada, Diciembre 2022

Editor: Universidad de Granada. Tesis Doctorales
Autor: Jacinto Carrasco Castillo
ISBN: 978-84-1117-750-4
URI: https://hdl.handle.net/10481/80705

Esta tesis doctoral ha sido desarrollada con la financiación de una beca predoctoral FPU (código FPU16/04188) adscrita al Ministerio de Universidades.

El conocimiento científico pertenece a la humanidad.

ALEXANDRA ASÁNOVNA ELBAKIÁN

AGRADECIMIENTOS

En esta sección quiero agradecer el trabajo y apoyo de todas las personas por las cuales ha sido posible el desarrollo de esta tesis doctoral y su posterior culminación en esta memoria.

A mis directores, Paco y Julián, por la ayuda, guía y consideración para desarrollarme como investigador.

En el ámbito profesional he de mencionar también a todo el personal docente e investigador con quien he compartido proyectos, ya sea bajo su supervisión, como Javier, Alberto y especialmente Salva; o codo con codo, como Diego, Jesús, Sergio, Germán, David C., David L. y Nacho.

En el plano personal he de agradecer fundamentalmente a mi familia por su apoyo incondicional, a mis padres y hermana por comprender el no haber estado lo suficiente, y a mi abuela por la confianza en mí. A Anabel, que a pesar de todo aún le quedaban fuerzas para darme ánimos. A mis amigos por los ratos de terapia de grupo y a Ana, por la real.

Por último, en un plano meramente práctico, al Estado por las becas y la Educación Pública; al personal de conserjería, secretaría y limpieza del CITIC, la ETSIIT y la UGR, no por más prosaico su labor es menos fundamental para la existencia de esta tesis; y a los Hub, Sci y Git, porque no es investigación si no se difunde.

ABSTRACT

The proliferation of the use of computer systems in all kinds of fields, whether medical, industrial, economic or scientific, has brought with it the generation of ever-increasing volumes of data. This has led to the need to create new technologies that allow the storage and analysis of this data, as well as generating new circumstances in which the aim is to extract knowledge from it. One of the usual scenarios is that of anomaly detection, where the interest lies in the identification of a minority class of data, either because it may pose a threat to the system under study, as in the case of fraud detection or predictive maintenance of industrial systems, or in medical environments, where there are few samples of data from patients with a disease compared to the common healthy population and the aim is to detect that disease. The fact that the focus is on the minority class differentiates anomaly detection from noise detection, defined as an effect on the data that we want to mitigate in the data pre-processing phase but whose cause is not relevant to the investigation.

Therefore, we can identify different scenarios within the scope of anomaly detection depending on the availability of information at the time of learning the algorithm: supervised scenarios, assimilable to unbalanced classification problems; semi-supervised or novelty detection scenarios, where a normality model is generated based on the data of the majority class, the only ones available in the training phase; and unsupervised scenarios, where no information is available on the class of the instances. These differences result in the existence of different evaluation methods and in the need to resort to additional mechanisms for the extraction of interpretable knowledge in scenarios where the representation learned by the model is insufficient for the understanding of the problem.

In this thesis we focus on the study of the anomaly detection problem for unsupervised scenarios, both for time series problems and for static data. This study starts from the demarcation of the problem within the anomaly detection domain to move on to the design of a distributed algorithm for anomaly detection valid for both static and time series data focused on obtaining explanations to help decision making and understanding of the studied dataset. Finally, an evaluation model for unsupervised time series anomaly detection scenarios is proposed.

Specifically, the proposals made in the framework of the thesis are:

- A distributed anomaly detection model focused on explainability. For this model we rely on the HBOS algorithm, which performs univariate histograms for anomaly score assignment, and extend it to search for anomalies in higher dimensionality subspaces. The use of this algorithm as a basis is justified by the possibility of constructing a knowledge representation that allows in later phases to reconstruct histograms of higher dimensionality subspaces by taking advantage of certain calculations. Furthermore, the knowledge representation allows us to include a proposal for the construction of rules to describe the reasons for the categorisation of specific instances through counterfactuals, rules that justify why an instance belongs to one class and not to another. In the experimentation associated with this proposal, it can be seen that the results are not comparable to the state of the art in anomaly detection, the lower performance being the counterpart to the simplicity of the model that allows the rules to be obtained.

- A model for evaluating anomaly detection algorithms for time series. In the field of anomaly detection, there are multiple evaluation schemes. In particular, it is common to find in time series scenarios the application of anomaly score prediction models for time instances while identifying events of interest that occur subsequent to the anomalous predictions. However, these methods pose problems such as the need to set certain parameters for the evaluation such as the definition of a window prior to the event of interest or weights to reward fast detection or the multiplication of the effect of inter-class imbalance. Therefore, we propose a scoring mechanism based on the definition of multiple windows prior to the events of interest and the use of a generalised ROC curve for the different windows such that the aggregation of the instances by a function is the anomaly score for that interval. This proposal includes an implementation for classical environments and another for distributed environments and a comparison with a proposed evaluation measure for anomaly detection assimilated by its work with intervals, where we show not only the usefulness of our measure for evaluation in the described scenarios but also the computational efficiency of our measure versus this alternative.

The proposals made provide solutions to specific problems in anomaly detection research, such as the lack of models capable of working in distributed environments and offering explanations as to why an instance is classified as anomalous or normal,

and the dissociation of certain evaluation systems that consider specific instances for the evaluation of events that occur over a period of time.

RESUMEN

La proliferación del uso de sistemas informáticos en todo tipo de ámbitos, tanto médico, industrial, económico y científico ha traído consigo la generación de volúmenes cada vez mayores de datos. Esto ha provocado la necesidad de generar nuevas tecnologías que permitan el almacenamiento y análisis de dichos datos, a la par que generar nuevas circunstancias donde se pretende extraer conocimiento de los mismos. Uno de los escenarios habituales es el de la detección de anomalías, donde el interés reside en la identificación de una clase minoritaria de los datos, bien porque pueda suponer una amenaza al sistema estudiado, como en el caso de la detección de fraude o en el mantenimiento predictivo de sistemas industriales, o bien en entornos médicos, donde se disponen de pocas muestras de datos de pacientes con una enfermedad frente al común de la población sana y se pretenda detectar dicha enfermedad. El hecho de que el foco caiga sobre la clase minoritaria diferencia la detección de anomalías de la detección de ruido, definido como un efecto sobre los datos que querremos mitigar en la fase de preprocesamiento de los datos pero cuya causa no es relevante para la investigación.

Por tanto, podemos identificar dentro del ámbito de la detección de anomalías distintos escenarios en función de la disponibilidad de información en el momento del aprendizaje del algoritmo: escenarios supervisados, asimilables a problemas de clasificación desbalanceada; escenarios semisupervisados o de detección de novedad, donde se genera un modelo de normalidad en base a los datos de la clase mayoritaria, los únicos disponibles en la fase de entrenamiento; y escenarios no supervisados, donde no se dispone de información sobre la clase de las instancias. Estas diferencias derivan en la existencia de distintos métodos de evaluación y en la necesidad de recurrir a mecanismos adicionales para la extracción de conocimiento interpretable en escenarios donde la representación aprendida por el modelo sea insuficiente para la comprensión del problema.

En esta tesis nos centramos en el estudio del problema de detección de anomalías para escenarios no supervisados, tanto para problemas de series temporales como para datos estáticos. Este estudio parte de la demarcación del problema dentro del ámbito de la detección de anomalías para pasar al diseño de un algoritmo distribuido para la detección de anomalías válido tanto para datos estáticos como para series

temporales enfocado en la obtención de explicaciones para ayudar a la toma de decisiones y la comprensión del conjunto de datos estudiado. Finalmente, se propone un modelo de evaluación para escenarios no supervisados de detección de anomalías en series temporales.

En concreto, las propuestas realizadas en el marco de la tesis son:

- Un modelo distribuido de detección de anomalías enfocado en la explicabilidad. Para este modelo nos basamos en el algoritmo HBOS, que realiza histogramas univariantes para la asignación de puntuación de anomalía, y lo extendemos para la búsqueda de anomalías en subespacios de mayor dimensionalidad. El uso de este algoritmo como base viene justificado por la posibilidad de construir una representación del conocimiento que permite en fases posteriores reconstruir histogramas de subespacios de mayor dimensionalidad aprovechando ciertos cálculos. Además, la representación del conocimiento nos permite incluir una propuesta de construcción de reglas para describir los motivos de la categorización de instancias concretas a través de contrahechos, unas reglas que justifican por qué una instancia pertenece a una clase y no a la otra. En la experimentación asociada a esta propuesta se ve cómo los resultados no son asimilables al estado del arte en la detección de anomalías, siendo el menor rendimiento la contrapartida a la simplicidad del modelo que permite la obtención de reglas.

- Un modelo de evaluación de algoritmos de detección de anomalías para series temporales. En el ámbito de la detección de anomalías existen múltiples esquemas para la evaluación. En concreto, es habitual encontrar en escenarios de series temporales la aplicación de modelos de predicción de puntuación de anomalía para instancias temporales mientras que identifiquen eventos de interés que ocurren con posterioridad a las predicciones anómalas. Sin embargo, estos métodos plantean problemas como la necesidad de establecer ciertos parámetros para la evaluación como la definición de una ventana previa al evento de interés o pesos para recompensar una detección rápida o la multiplicación del efecto del desbalanceo entre clases. Por ello, proponemos un mecanismo de puntuación basado en la definición de múltiples ventanas previas a los eventos de interés y el uso de una curva ROC generalizada para las distintas ventanas de manera que la agregación de las instancias mediante una función es la puntuación de anomalía para ese intervalo. Esta propuesta incluye una implementación para entornos clásicos y otro para entornos distribuidos y una

comparación con una propuesta de medida de evaluación para detección de anomalías asimilable por su trabajo con intervalos, donde mostramos no solo la utilidad de nuestra medida para la evaluación en los escenarios descritos sino también la eficiencia del cómputo de nuestra medida frente a esta alternativa.

Las propuestas realizadas vienen a aportar soluciones a problemas concretos de la investigación en detección de anomalías como son la falta de modelos capaces de trabajar en entornos distribuidos y que ofrezcan explicaciones sobre el motivo de la clasificación de una instancia como anómala o normal, y la disociación de ciertos sistemas de evaluación que consideran instancias puntuales para la valoración de eventos que ocurren a lo largo de un período.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN

Desde hace unos años, se generan ingentes cantidades de datos día a día en empresas, centros de investigación y universidades. Dado el gran volumen de los datos que se están generando continuamente, se dice que ahora estamos en la era del *Big Data*. El *Big Data* se caracteriza por lo que se conoce como las 5 Vs del *Big Data*: Volumen, por la gran cantidad de datos; Velocidad, por la alta velocidad a la que se generan, generalmente de forma automática y constante; Variedad, por la característica de los datos de haber sido generados por distintos dispositivos; Veracidad, por los posibles errores que puedan contener los datos dadas el resto de características; y, por último, Valor, por el valor que tienen los datos y el conocimiento que se puede llegar a extraer de ellos.

Los datos por tanto, no solo por su gran volumen, son muy importantes porque contienen información valiosa para las entidades que los generan. De esta forma, es útil procesar estas cantidades de datos con el objetivo de obtener dicha información valiosa, o conocimiento, en un formato que sea legible y permita ser usado por las entidades con el objetivo de mejorar sus productos, métodos de producción, disminuir costes, etc.

La proliferación de sistemas que producen grandes cantidades de datos ha promovido el desarrollo de la Ciencia de Datos, un conjunto de técnicas provenientes de campos como la esdística, las matemáticas y la computación dedicadas a la extracción de conocimiento de los datos disponibles. Dichas técnicas se enmarcan dentro del proceso conocido como *Knowledge Discovery in Databases* (KDD) [HPK11], del que podemos ver un esquema en la Figura 1.1 y que está compuesto de ciertas fases que estructuran los pasos a seguir para la extracción de información valiosa de los datos:

- Especificación del problema: La demarcación de los objetivos del problema es un aspecto fundamental de cualquier proyecto de aplicación del KDD.
- Muestreo de datos: En función de los intereses y la naturaleza del proyecto habrá que adaptar la toma de muestras del suceso observado.

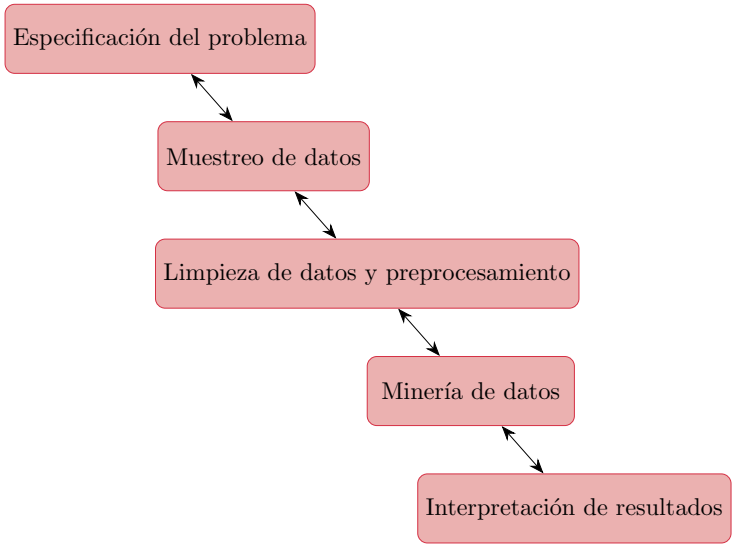


Figura 1.1: Resumen del proceso KDD

- Limpieza de datos y preprocesamiento: En el proceso de recopilación de datos se pueden introducir inconsistencias que deben arreglarse en este paso del proceso para no afectar a la extracción de conocimiento. Entre estos procesos se incluyen el filtrado de ruido o la imputación de valores perdidos [LGR+20].
- Minería de datos: En este paso se aplican algoritmos de aprendizaje automático para la extracción de conocimiento a partir de los datos limpios. Por ser la parte del proceso en el que se extrae la información, a veces aparece el todo el proceso KDD con el nombre de minería de datos.
- Interpretación de resultados: En el último paso se aplican técnicas de visualización para ayudar a la interpretación de los resultados.

La definición del problema dentro del proceso KDD determina el enfoque de los sucesivos pasos. Cuando se trabaja con cantidades de datos lo suficientemente grandes como para hablar de *Big Data*, es necesario crear y utilizar nuevos algoritmos que sean capaces de tratar y manejar tales datos. Uno de los primeros modelos de

programación propuestos fue *Map Reduce* [DG04], consistente en la realización del cómputo de manera distribuida para una posterior agregación mediante unas claves intermedias generadas en la primera parte del cómputo, de manera que se abstrae el problema de lectura y escritura en disco. El *framework* Hadoop hace uso de un sistema de ficheros propio robusto a fallos y el paradigma *Map Reduce* para realizar este análisis en entornos distribuidos [Whi12]. El siguiente nivel en el procesamiento de datos de gran escala fue provocado por Spark, que introdujo el concepto de RDDs (*Resilient Distributed Datasets*), unas colecciones de objetos distribuidas y que pueden ser reconstruidas en caso de fallo [ZCF+10]. Las mejoras introducidas se notan especialmente en tareas iterativas, algo especialmente utilizado en aprendizaje automático, y en tareas analíticas interactivas.

Una de las tareas fundamentales en la fase de preprocesamiento es la de filtrar o corregir las observaciones que dificulten el proceso de aprendizaje de los algoritmos, aunque la identificación de dichas instancias puede representar el conocimiento que se espera extraer del problema abordado, como ocurre en los escenarios de detección de anomalidad. Esta situación se da en tareas como la detección de fraude, ciberseguridad o el mantenimiento predictivo de sistemas industriales. Estas situaciones donde las observaciones que se desvían del comportamiento esperado son las relevantes para el problema estudiado se recogen bajo el término paraguas de detección de anomalías, siendo la definición clásica de anomalía la de una observación cuyas diferencias con el resto hacen pensar que han sido generadas por un mecanismo alternativo [Haw80]. En este caso, este mecanismo es el que nos resulta de interés, pues en lugar de resultar ruido aleatorio, las diferencias podrían provenir de algún fallo en el sistema estudiado que querremos identificar y corregir, y no se trataría de un ruido que fuera necesario filtrar en la fase de preprocesamiento por ser información irrelevante.

Dentro de la fase de minería de datos, los algoritmos de aprendizaje que se pueden usar se clasifican de manera general en función de si la variable objetivo que se quiere predecir es conocida o no, obteniendo las siguientes tres grandes clases:

- Aprendizaje supervisado: en el conjunto de datos del que disponemos, la variable objetivo es conocida, por lo que estamos interesados en encontrar una relación entre los datos y la variable objetivo, de forma que podamos usar dicha relación para predecir la variable objetivo cuando no sea conocida. Dentro del aprendizaje supervisado podemos distinguir:
  - Regresión: cuando la variable objetivo es continua.

    - Clasificación: cuando la variable objetivo es discreta, de forma que cada instancia del conjunto de datos pertenece a una clase dentro de un conjunto de clases previamente definido.

- Aprendizaje no supervisado: la variable objetivo no es conocida en el conjunto de datos del que disponemos. En este caso estamos interesados en encontrar relaciones implícitas entre los propios datos. Dentro del aprendizaje no supervisado podemos distinguir:
    - *Clustering*: se busca agrupar las instancias del conjunto de datos en distintos grupos de forma que las instancias que pertenecen al mismo grupo sean lo más parecidas entre sí a la vez que se busca que las instancias de distintos grupos sean los más distintas entre sí.
    - Reglas de asociación: se buscan patrones o asociaciones frecuentes entre las instancias del conjunto de datos del que se dispone, llamadas reglas.
- Aprendizaje semisupervisado: cuando en el conjunto de datos del que se dispone hay un subconjunto de instancias para el que se conoce la variable objetivo y otro subconjunto para el que no se conoce dicha variable.

Sin embargo, el problema abordado en esta tesis de detección de anomalías recoge distintos escenarios de aplicación que se enmarcan dentro de los tres grandes tipos de aprendizaje: Una vertiente de la detección de anomalías se enmarca dentro de la clasificación supervisada, donde unas pocas instancias conocidas representan los eventos de interés, mientras que otros modelos siguen un enfoque no supervisado aunque el objetivo sea también el de clasificación. En el siguiente capítulo profundizaremos sobre la taxonomía de escenarios que reciben el nombre de detección de anomalías.

Como hemos visto, la diversidad de escenarios es una cuestión fundamental tanto para la realización de propuestas algorítmicas como para la evaluación de las mismas. Además, la difuminación de las fronteras entre los distintos tipos de aprendizaje ha resultado en una falta de propuestas para escenarios con unas características y problemáticas espefícicas como el de *Big Data*.

En esta tesis nos centraremos en el estudio de los métodos no supervisados para la detección de anomalías, su extensión a entornos *Big Data*, la inclusión de mecanismos de explicabilidad y la evaluación de los algoritmos en escenarios de series temporales.

2 FUNDAMENTOS DE LA DETECCIÓN DE ANOMALÍAS

En este capítulo abordaremos con más profundidad el problema de detección de anomalías, tanto la propia definición del problema más allá de la simplificación de la búsqueda de la anormalidad, como las familias de algoritmos principales y las técnicas específicas para la extracción de conocimiento interpretable en entornos generalmente no supervisados.

CLASIFICACIÓN DE ESCENARIOS El término de detección de anomalías aún agrupa multitud de escenarios y enfoques para el estudio del problema, por lo que se ha propuesto una nomenclatura que sirve a la vez de taxonomía de las propuestas algorítmicas para la detección [CIL19]. Principalmente hay dos características que nos afectan a la hora de categorizar los escenarios:

- El conocimiento sobre las clases representadas en los datos disponibles en el momento de entrenamiento, ya sea porque únicamente dispongamos de instancias que sabemos normales (**Detección de novedad** o **Escenarios semisupervisados**), o porque no tengamos información sobre la naturaleza de las observaciones (**Detección de *outliers*** o **Detección no supervisada**).
- La temporalidad de los datos, es decir, si existe una relación temporal entre las observaciones. Esto discrimina entre los escenarios donde se clasifican series temporales completas (**Clasificación de eventos raros**) y aquellos escenarios donde se clasifican observaciones independientes con un alto grado de desbalanceo entre clases, generalmente en datos estáticos (**Detección de anomalías**).

En el desarrollo de esta tesis nos centraremos en escenarios no supervisados tanto para datos estáticos como para datos no etiquetados con una componente temporal y una información limitada sobre la presencia de instancias anómalas.

2.1 TAXONOMÍA DE ALGORITMOS

En esta sección hacemos un repaso por diferentes algoritmos de detección de anomalías categorizados según el mecanismo para identificar las diferencias entre las clases y las instancias anómalas [Agg17].

2.1.1 *Modelos probabilísticos*

Los modelos probabilísticos representan un primer enfoque para la detección de anomalías, basando su comportamiento en la consideración de que la población normal sigue una distribución conocida, con lo que se puede estimar la probabilidad de pertenencia de las instancias sospechosas a la distribución ajustada.

ECOD La idea del ajuste de distribuciones se realiza en el modelo ECOD [LZH+22] para cada característica, construyendo la puntuación de anomalía en base a la función de distribución acumulada empírica. Con la construcción propuesta de la función de distribución se evita la parametrización, que restringe el modelado a distribuciones concretas. La simplicidad de la base del funcionamiento permite extraer explicaciones interpretables.

2.1.2 *Modelos Lineales*

Habitualmente en los datos estudiados existen correlaciones entre las diferentes variables estudiadas en un problema, especialmente si los datos provienen de experimentación real. Basándose en esta hipótesis surgen varios modelos de aplicación usual en problemas de regresión y clasificación, como los modelos lineales, *Principal Component Analysis* (PCA) o *Support Vector Machine* (SVM). Usando estos modelos, para la detección de anomalías se ha utilizado el error de reconstrucción o la distancia a la frontera de decisión como puntuación de anomalías. De estos modelos, el más utilizado es el de *OneClass-SVM* para la construcción de modelos de normalidad en escenarios de detección de novedad o detección de anomalías semisupervisada [SPS+01].

2.1.3 *Modelos basados en proximidad*

Los modelos basados en proximidad utilizan para la definción del grado de anomalía la distancia a las observaciones más cercanas, así como la densidad del vecindario de un item. De esta manera, instancias muy alejadas o con cambios en la densidad de observaciones son susceptibles de identificarse como anómalas. Una desventaja conocida de estos métodos es la pérdida de significado y rendimiento en escenarios de alta dimensionalidad, pues a medida que esta crece, las distancias entre las observaciones tienden a ser homogéneas, con lo que esta información deja de discriminar entre instancias normales y anómalas [HAK00]. Los métodos actuales basados en proximidad mitigan este efecto mediante la agregación de las puntuaciones obtenidas en subespacios de menor dimensionalidad:

k-NN Para problemas de cualquier índole, uno de los enfoques de resolución consiste en la comparación de la instancia a evaluar con las observaciones más cercanas. En escenarios no supervisados, el enfoque propuesto por el algoritmo *k*-NN consiste en el cómputo de la distancia a la *k*-ésima instancia más cercana como puntuación de anomalía [RRS00]. En función del conjunto de datos, otros mecanismos de agregación de la distancia a los *k* vecinos más cercanos pueden proporcionar mejores resultados, como la media o la mediana [AP02].

Para series temporales también se ha adaptado este algoritmo, basándose en la distancia de las observaciones agrupadas en ventanas deslizantes a las ventanas previas, incluyendo también una ventana adicional como calibración para disminuir la tasa de falsos positivos [INB+17].

LOF Aunque la distancia a los vecinos más cercanos es una medida relevante sobre el grado de anomalía, regiones con distintas densidades de puntos pueden aportar más información. En este sentido, LOF propone el concepto de *alcanzabilidad*, referida al máximo entre la distancia entre dos observaciones, y la distancia de uno de ellos a sus *k* vecinos más cercanos [BKN+00]. De esta manera, se define el factor *outlier* local como una relación entre la densidad de un punto frente a sus vecinos.

HBOS Una simplificación del modelo es la propuesta en HBOS [GD12], que construye un histograma por cada variable y usa el número de observaciones en cada intevalo por cada dimensión para el cálculo de la puntuación de anomalía.

LODA Como hemos mencionado anteriormente, una mayor dimensionalidad se traduce en una menor significancia de las distancias entre observaciones. Por ello, el algoritmo LODA realiza una selección aleatoria de las proyecciones unidimensionales para el cálculo de la puntuación de anomalías y así mitigar el impacto de variables con una menor importancia [Pev16].

ROD Otros métodos aptos para escenarios de alta dimensionalidad están basados en ángulos, trabajando con la hipótesis de que los ángulos formados por la mayoría de instancias con los elementos anómalos como vértice serán menores que los ángulos con vértice en las instancias normales [KZ+08].

La propuesta de Almardeny *et al.* [ABC20] se basa también en los ángulos, pero en esta ocasión realiza una media entre los ángulos que forma la mediana de cada subespacio de 3 dimensiones con la proyección de cada instancia. De esta manera, podríamos identificar instancias anómalas en estos subespacios y además dar una explicación sobre la localización de la alarma.

ONECLASS-SVM Una propuesta fundamental en la detección de anomalías, tanto por ser la adaptación de *Support Vector Machine* (SVM) a escenarios de detección de anomalías como por ser la propuesta más conocida para clasificación *One Class* o detección de novedad es *OneClass SVM* [SPS+01]. Este método construye un hiperplano que cubre los puntos incluidos en el conjunto de entrenamiento y espera dejar fuera aquellas instancias que se alejen del comportamiento normal, de igual manera que SVM calcula un hiperplano para la separación de las clases.

2.1.4 *Ensamblado de modelos*

Un mecanismo de diseño de algoritmos que ha tenido un gran crecimiento en los últimos años es el ensamblado de modelos. Esto explota el buen desempeño de modelos simples en regiones concretas del espacio de entrada para construir una mezcla que sea robusta [TD01]. Esta metodología ha sido ampliamente usada en entornos supervisados por la disponibilidad de etiquetas para seleccionar el mejor modelo base en cada caso, sin embargo hay más dificultades para realizar propuestas en el caso de detección de anomalías o clasificación no supervisada, pues la decisión de qué modelo aplicar se realiza sin información sobre la calidad de los mismos [WM19]. Además, la puntuación de anomalía es generalmente una

medida construida de manera relativa a las instancias contempladas en la fase de entrenamiento y propia de cada método, por lo que será necesario darle un significado probabilístico para poder realizar una combinación entre métodos con puntuaciones heterogéneas [GT06].

Una posibilidad es la creación de una pseudoobservación mediante la agregación de múltiples modelos y la evaluación de los detectores base en función de la correlación con esta pseudoobservación en entornos de la instancia base [WM19; ZLN+19].

ISOLATION FOREST La propuesta de *Isolation Forest* [LTZ08] es una adaptación a entornos no supervisados del modelo de *Random Forest*, que basa su funcionamiento en la construcción de árboles usando características y valores aleatorios, de manera que se realiza una partición del conjunto de entrada. En el método para detección de anomalías, la construcción de los árboles se realiza hasta que cada instancia del conjunto de datos permanezca en un nodo hoja. Posteriormente, para cada observación se calcula la profundidad media de los nodos en los que se encuentra, de manera que las instancias que son fácilmente aislables son identificadas como anómalas.

RANDOM CUT FOREST El modelo de *Random Cut Forest* [GMR+16] está enfocado a entornos de *data stream*, donde se actualiza una representación del flujo de entrada y el grado de anomalía se basa en la novedad que representa la instancia para dicha representación. La construcción de árboles en este algoritmo difiere del anterior en que la elección de la característica involucrada en la partición se hace de manera proporcional al tamaño del espacio para dicha característica, algo enfocado en evitar la proliferación de falsas alarmas cuando existen variables irrelevantes.

STACKING Aprovechando la información que pueden dar modelos supervisados, se han realizado varias propuestas que conectan varios algoritmos basándose en la hipótesis de que la salida de un algoritmo puede representar información relevante para que la aproveche otro, generalmente estando la pareja de algoritmos formada por un método supervisado y otro no supervisado [MMA14]. Este método de ensamblado se conoce como *stacking*.

Para entornos supevisados, Zhao *et al.* proponen un método de *stacking* para utilizar la puntuación de modelos de clasificación desbalanceada para utilizar las puntuaciones de anomalía como entrada para un algoritmo XGBOOST [ZH18].

SUOD Adicionalmente a los modelos basados en detectores simples, el *framework* SUOD [ZHC+21] propone la aplicación de sistemas de resumen de datos y la combinación de varios modelos de aprendizaje para acelerar la obtención de resultados utilizando modelos supervisados de regresión sobre pseudoetiquetas construidas con la combinación de modelos no supervisados.

2.1.5 *Modelos basados en redes*

DEEPSVDD De manera similar al método OneClass SVM, el método DeepSVDD aprende una función de transformación para la construcción de una hiperesfera de volumen mínimo[RVG+18], con la salvedad de que en este caso el aprendizaje se realiza mediante redes de *deep learning*. Este método parte de la hipótesis de que sólo se tienen datos normales en el conjunto de entrenamiento, algo que generalizan los autores en la siguiente propuesta realizada DeepSAD [RVG+19].

DEEP AUTOENCODING GMM El mecanismo habitual en las propuestas de detección de anomalías mediante redes *autoencoders* está basado en la compresión de los datos a una menor dimensión (conocido como espacio latente) mediante una red *encoder*. Posteriormente, se reconstruyen los datos de entrada con la red *decoder* y en el entrenamiento de ambas redes se busca la minimización del error de reconstrucción. De esta manera el algoritmo aprende las características de las instancias normales y el error de reconstrucción sirve como puntuación de anomalía [SY14].

En la propuesta de Zong *et al.* [ZSM+18], se añade una segunda fase donde se entrena un modelo de mezcla de gausianas con el error de reconstrucción a la par que la red *autoencoder* para facilitar el ajuste de hiperparámetros del modelo estadístico con la intención de salir de óptimos locales.

VARIATIONAL AUTOENCODER En las propuestas basadas en redes *autoencoder* variacionales se estima una red *encoder* probabilística que se encarga de realizar una aproximación al modelo generativo y de aprender sus parámetros [KW14].

GANOMALY En los últimos años han tenido gran auge las propuestas basadas en *Generative Adversarial Networks* (GAN), especialmente para problemas que tratan con imágenes. La idea consiste en entrenar dos redes adversarias durante el proceso de entrenamiento de manera que compitan, con una generando imágenes que se

hagan pasar por las del espacio de entrada, y la segunda que trate de discriminar su origen. La propuesta para detección de anomalías GANomaly [AAB18] consiste en que la primera red sea un modelo *autoencoder*, de manera que en la fase de reconstrucción, difícilmente se reflejen las anomalías de la instancia original. La segunda red comprime la salida de la red *autoencoder*, con lo que debería quedar algo similar al vector del espacio latente de la primera red. De esta manera, la tercera red que discrimina en los modelos GAN sirve como predictor de la clase de anomalía.

2.1.5.1 *Clasificación de series temporales*

Como comentamos anteriormente, un escenario habitual dentro del ámbito de la detección de anomalías es la clasificación de series temporales. Aunque en la literatura la detección de eventos raros pueda tratarse de la clasificación de series temporales completas, la consideración de tramos de dichas series temporales como instancias independientes lo hace un problema más entendible, por ejemplo en la detección de arritmias, como vemos en la Figura 2.1. En este caso, la serie temporal completa puede trocearse de manera que que una arritmia sea detectada en un momento dado significa que se está produciendo, no será algo de un único instante ni de la serie temporal completa.

Para dicho cometido, incluimos en esta sección algunos métodos basados en redes neuronales que utilizan una mezcla de redes convolucionales (CNN, por sus siglas en inglés) y redes *Long Short-Term Memory* (LSTM). Dichos tipos de redes emplean arquitecturas y capas que las hacen propicias para trabajar en el ámbito de las redes neuronales. En concreto, las operaciones de convolución sirven como agregación de características, de manera que se pueden aprender propiedades de la entrada a distintos niveles, que serían diferentes escalas de tiempo en las series temporales [HW20]. La extracción de características mediante capas convolucionales es también la base del modelo YiboGao [GWL21], que incluye en los bloques de su red unas capas para detectar posible ruido en la señal y filtrar esa entrada.

De igual manera que una entrada ruidosa puede afectar a la salida, las LSTM poseen mecanismos para detectar cuándo el ajuste del modelo decae y conviene olvidar información anterior, a la vez que incorpora la nueva información del ajuste, con lo que la evolución del modelo las hace también candidatas idóneas para escenarios de series temporales.

En la clasificación de eventos raros la entrada consiste en una serie temporal, en ocasiones de longitud variable. Sin embargo, los modelos normalmente reciben una

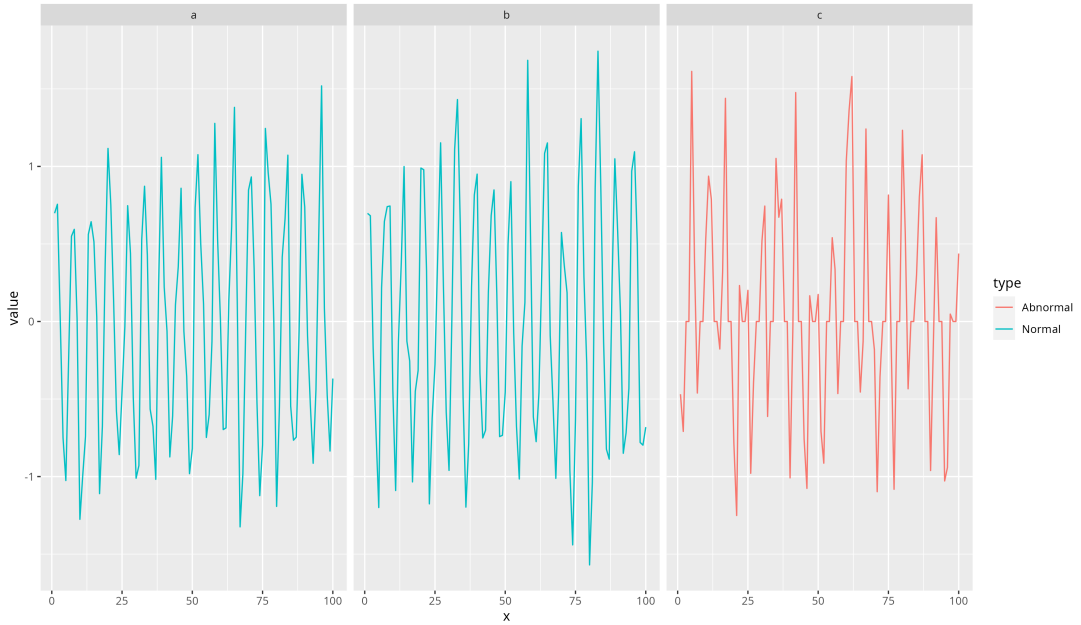


Figura 2.1: Ejemplo de clasificación de series temporales

entrada de tamaño fijo, con lo que puede ser necesario una tarea de preprocesamiento, como rellenar la serie temporal, como en el modelo propuesto por Oh *et al.* [ONT+18]. En este modelo también se incluye en las capas de la red CNN una operación de *max pooling*, una operación que selecciona el máximo de una región de la capa anterior y que es habitual también en problemas que tratan con imágenes.

Uno de los problemas de las redes neuronales, y en especial de las LSTM, es su elevado coste computacional, debido a la gran cantidad de parámetros que deben aprender. Además, la dificultad en muchas ocasiones para obtener un gran conjunto de entrenamiento que sea suficiente para su entrenamiento puede hacer que los modelos sobreajusten, lo que llevaría una pérdida de eficacia en entornos reales. Por ello, una de las técnicas es la de *drop out*, que son capas que desactivan ciertas neuronas de forma aleatoria para evitar que adquieran un sesgo excesivo hacia la entrada. Otra técnica para tal efecto es la normalización por lotes de la entrada, de manera que no tengan un peso injustificado por razones aleatorias en la configuración del modelo, y esta técnica es la usada en el trabajo de Wei [WZZ+19], junto con una mayor tasa de aprendizaje, que permite reducir el tiempo de entrenamiento.

En ocasiones la entrada de las redes LSTM que aprenden la relación temporal no viene determinada sólo por la salida de la red CNN que extrae las características, sino que se le puede introducir también información específica del problema extraída mediante conocimiento experto u otras técnicas. Este es el caso de la red *ad hoc* para un problema de detección de arritmias de Chen *et al.* [CHZ+20].

2.1.6 *Algoritmos de predicción de series temporales*

La predicción o *forecasting* para series temporales se puede concebir como un problema de regresión que incluye un horizonte temporal de predicción. Estos modelos se usan también en detección de anomalías usando el error en la predicción como puntuación de anomalía.

MÉTODOS PARA SERIES UNIVARIANTES La descomposición de las series temporales en componentes de tendencia y estacionalidad es la base para algunos métodos clásicos de predicción como ARIMA, y por tanto también de detección de anomalías basados en él [BGM+01]. Más recientemente, empleados de Facebook propusieron el método Prophet para solventar algunos problemas de ARIMA como la necesidad de ajustar los parámetros, una tarea difícil para investigadores sin experiencia [TL18].

Otro modelo con base en modelos estadísticos es el de ventanas móviles ponderadas exponencialmente (*exponentially weighted moving average,* EWMA), utilizado para detección de cambio en el comportamiento general de una serie temporal [RPL15; CS12].

VECTOR AUTOREGRESSION MODEL Los modelos de vectores autorregresivos (*vector autoregressive*, VAR) [Lüt05] se usan para la predicción y estudio estructural de series temporales. Este modelo parte de una representación del modelo de orden p como:

$$y_t = \nu + A_1 y_{t-1} + \cdots + A_p y_{t-p} + u_t, t = 0, \pm 1, \pm 2, \ldots,$$

donde y_t y ν son vectores de dimensión K (las variables de nuestro problema), A_i son matrices de coeficientes y u es un vector K-dimensional de ruido blanco. De esta manera, se modela la observación t como una combinación lineal de las p observaciones anteriores. Se debe tener en cuenta que en la capa de datos se permitía distinta frecuencia de muestreo en las diferentes dimensiones, de manera similar a

la que los modelos VAR tratan como sucesivas las observaciones sin considerar el tiempo de muestreo. El modelado matricial de la serie temporal permite estudiar propiedades como la estabilidad y la estimación mediante el método de mínimos cuadrados.

GRADIENT BOOSTING FORECASTER En este trabajo sobre árboles de decisión con *gradient boosting* [KMF+17] se incluyen dos propuestas para realizar de manera eficiente el muestreo con el que obtener la información necesaria para realizar el particionamiento del espacio de características. La primera da más peso en el submuestreo a las instancias que contribuyen más al error (aquellas que están infraaprendidas por el algoritmo). En la segunda, se agrupan las características para traducir el problema de selección de características a uno de un coloreado de un grafo, que se resuelve mediante un algoritmo *greedy*.

RANDOM FOREST FORECASTER El modelo original de *Random Forest* [Ho95] se usa también en problemas de regresión, por lo que se puede utilizar también desde la perspectiva de predicción y posterior detección de anomalías usando el error cometido.

2.2 TÉCNICAS DE EXPLICABILIDAD PARA DETECCIÓN DE ANOMALÍAS

La naturaleza no supervisada del problema de detección de anomalías incrementa la necesidad de que los sistemas de aprendizaje ofrezcan detalles sobre la motivación de su salida para mejorar el conocimiento extraído de los datos. Cuestiones como la confianza en la robustez del modelo frente a entradas similares, la capacidad de explicar las causas de las decisiones de manera que sean interpretables por humanos o la capacidad de interactuar con estos son cualidades deseables de los modelos [BDD+20]. Este conjunto de técnicas se enmarcan dentro de la *eXplainable Artificial Intelligence* (XAI), que tiene como objetivo la obtención de modelos interpretables, bien mediante explicaciones generadas por el propio modelo sobre las decisiones tomadas, o bien mediante modelos *post-hoc* que se basan en la aproximación a los modelos preexistentes ya entrenados.

Ciertas propuestas existentes para la explicación de la detección de anomalías pasan por la búsqueda de características que identifican las instancias como anómala, bien mediante la construcción de un retículo con los subespacios donde se identifican

las anomalías [KN99] o la búsqueda de proyecciones que pongan de relieve estas instancias anómalas [DAN+14]. Otras propuestas buscan estos subespacios para construir una regla de asociación basada una puntuación de anomalía de una instancia con respecto a la distribución de las observaciones normales para ese subespacio [AFP13]. Parecida a estas técnicas de búsqueda de los subespacios, el algoritmo LookOut [GES+19] busca el espacio de dos dimensiones que mejor representa un conjunto de anomalías para producir el gráfico que los aísla del resto de puntos normales.

Otros métodos de explicabilidad actúan con posterioridad al método de aprendizaje, una característica que los hace propicios para su uso cuando el modelo de aprendizaje es difícil de modificar una vez aprendido [RYS+21]. Uno de estos métodos es *Layer Relevance Propagation* (LRP) [BBM+15], que usa los pesos de capas sucesivas en modelos basados en redes para determinar la influencia en la clase predicha. El método de *Local Interpretable Model-agnostic Explanation* (LIME) [RSG16] ha ganado gran popularidad por la representación gráfica que ofrece de las explicaciones. Este método crea modelos sustitutivos más simples que imiten el comportamiento en entornos locales para identificar las variables (o píxeles en el caso de imágenes) cuyo cambio aleatorio genera un cambio en la determinación de la clase. El cambio en características concretas como explicación es también el fundamento del funcionamiento de SHAP [LL17]. Otro enfoque de explicaciones válido para distintos modelos es el de *counterfactuals* o contrahechos [WMR17], consistente en la búsqueda de las variaciones de una instancia que provocan el cambio de clase de la misma. Este método, que se puede ejecutar con posterioridad a cualquier modelo puede sin embargo aprovecharse de las representaciones usadas por el propio algoritmo para facilitar el cómputo de las reglas y explicaciones.

2.3 METODOLOGÍA DE EVALUACIÓN

El proceso de extracción de conclusiones de la experimentación con algoritmos de ciencia de datos incluye la evaluación de los mismos para seleccionar aquellos que obtengan un mejor rendimiento en la resolución del problema abordado. En el caso de la detección de anomalías, el enfoque usual es la consideración del mismo como un problema de clasificación desbalanceado, donde se disponen de datos etiquetados, y cuya etiqueta puede ser tomada en cuenta o no por los algoritmos en el período de entrenamiento, pertenecientes a dos clases: la clase mayoritaria o clase negativa, y

la clase minoritaria o clase positiva. Al igual que en la clasificación desbalanceada, el interés no solo reside en el acierto global del algoritmo sino en la capacidad de identificación de las instancias de la clase minoritaria.

EVALUACIÓN DE LA DETECCIÓN DE ANOMALÍAS PARA DATOS NO TEMPORALES La precisión en n (*precision at n*, *P@n*) se define para los métodos que proporcionan un orden (o una puntuación de anomalía que permita ordenarlos) de anomalía sobre las instancias como la proporción de las primeras n instancias que son efectivamente anomalías [Cra09]. Si n coincide con el número de anomalías en el conjunto de datos, el autor denomina *P@n* como la *R*-precisión. La fiabilidad de esta medida queda perjudicada por tanto por el parámetro n, especialmente en escenarios no supervisados [XLY19]. La medida *P@n* ignoraría la componente temporal en escenarios de mantenimiento predictivo, considerando las observaciones como instancias aisladas.

El problema del balanceo de clases se aborda en la clasificación desbalanceada con la curva ROC [FGG+18]. El espacio ROC se define como un espacio $[0,1] \times [0,1]$ usando la tasa de verdaderos positivos (*True Positive Rate*, TPR o sensibilidad, representada en el eje Y), y la tasa de falsos positivos (*False Positive Rate*, FPR o uno menos la especificidad, representada en el eje X) [HM82]. Por ejemplo, un algoritmo con unas TPR y FPR perfectas se situaría en el punto $(0,1)$. La extensión de este concepto se usa generalmente para algoritmos que proporcionan puntuaciones o probabilidades en escenarios de clasificación desbalanceada. Entonces, se puede conseguir puntos ROC diferentes para cada posible umbral y tanto TPR y FPR se incrementan a la par que este umbral.

TRANSFORMACIÓN EN REGRESIÓN Como hemos descrito anteriormente, los datos temporales representan una gran proporción de los escenarios reales de aplicación. La tarea más común en los problemas de series temporales es modelar la serie temporal estudiada, un escenario donde la componente temporal es intrínseca a los datos estudiados, de manera que la medida de evaluación no necesita ser consciente de esta naturaleza temporal y usan una medida de regresión como la raíz del error cuadrático medio (*Root Mean Squared Error*, RMSE) o el ratio de error [ZWL+13; ZWZ16]. Sin embargo, para el caso específico de la detección de anomalías en series temporales, el interés recae en el evento anómalo que ocurre en un punto concreto de la serie temporal, no los valores de la serie temporal en sí.

Uno de los enfoques para la evaluación de detección de anomalías en series temporales consiste en la transformación del problema de detección de eventos raros

en un problema de regresión, donde para cada observación $i = 1, \ldots, N$, el objetivo es el tiempo restante r_i hasta el fallo o la parada [GSY+19]. Esta propuesta es interesante, puesto que en el mantenimiento predictivo se pretende maximizar la productividad con los mínimos costes de reparación mediante el conocimiento de nuestro sistema. Así pues, buenas predicciones del tiempo restante de funcionamiento hasta el fallo ($R(x_i) = \hat{r}_i, i = 1, \ldots, N$) nos ayudarán en esta tarea. La evaluación de medida es el RMSE, al igual que en muchos problemas de regresión. Este enfoque incrementa la relevancia de la componente temporal en la detección de fallos, y usar el RMSE como la medida de calidad resuelve el desacople entre las predicciones individuales y la naturaleza continua del problema.

DETECCIÓN DE EVENTOS RAROS Y PRONTITUD En el escenario de problemas de clasificación de series temporales la prontitud se define como la media del porcentaje de la longitud t_j de cada serie temporal X_j necesitado para proporcionar una etiqueta de clase Y_j. La necesidad de tener en cuenta la distancia hasta el evento predicho se aborda también por Zhang *et al*. [ZBR+17]. En este trabajo, transforman la clasificación de eventos raros en un problema de clasificación a través de la definición de un horizonte temporal, en el cual las instancias deben clasificarse como anómalas. Además, se asigna un peso a estas instancias para dar más relevancia a aquellas más cercanas al evento.

MEDIDAS DE DETECCIÓN DE ANOMALÍAS BASADAS EN INTERVALOS Un escenario relevante para la detección de anomalías temporales es la consideración de los intervalos de tiempos como anomalías [TLZ+19; LGT+18]. En estos trabajos, Tatbul *et al*. proponen una generalización basada en intervalos para los conceptos de *precision*, *recall* y la medida F_1. La salida esperada de los algoritmos es una etiqueta para cada instancia, que se transforma en intervalos de instancias anómalas contiguas. Estas medidas evalúan cada intervalo anómalo (R_i para los eventos realmente anómalos y P_i para los intervalos anómalos predichos) usando la puntuación basada en la detección ($E(R_i, P)$, la existencia de al menos una instancia temporal que pertenece a R_i y cualquier $P_i \in P$), y la superposición ($O(R_i, P)$, la proporción de las instancias detectadas en R_i). Por tanto, con la aplicación de este método el foco recae en los intervalos anómalos etiquetados como tal. Esta propuesta permite la modificación de ciertos parámetros para recompensar diferentes comportamientos, como una detección de anomalías temprana o más cercana al evento dentro del intervalo de anomalía.

Esta medida tiene una directa modificación para evaluar a los algoritmos para el escenario estudiado, definiendo los intervalos temporales anómalos como las ventanas que preceden los eventos de interés.

BENCHMARK DE DETECCIÓN DE ANOMALÍAS DE NUMENTA (*numenta anomaly benchmark*, NAB) Un problema esencial para el estudio de series temporales es que no existen *benchmarks* para la comparación del desempeño de algoritmos de detección de anomalías en *streaming*. Una propuesta fundamental es *Numenta Anomaly Benchmark* [LA15] (NAB). Este *benchmark* propone una función de puntuación y proporciona un conjunto de series temporales etiquetadas manualmente provenientes de entornos de aplicación real. La propuesta se basa en la definición de cuatro pesos $A_{TP}, A_{FP}, A_{TN}, A_{FN}$ para ponderar los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, respectivamente. El valor de estos pesos se sitúa en el intervalo $[-1, 1]$, con $A_{TN}, A_{TP} \geq 0$ y $A_{FN}, A_{FP} \leq 0$ para penalizar los errores. El escenario por defecto descrito en NAB usa los pesos $A_{TP} = A_{TN} = 1$ y $A_{FP} = A_{FN} = -1$, aunque se pueden definir distintos perfiles que los cambien. Entonces, se define una ventana previa a cada anomalía. En estas ventanas se mantiene sólo la primera alarma detectada por el algoritmo, mientras que en la ventana que precede una anomalía que no tiene una alarma se considera como falsos negativos, lo que podría llamarse como una ventana perdida. Siendo y la relativa posición de la alarma dentro del intervalo, se define la función de puntuación como:

$$\sigma^A(y) = (A_{TP} - A_{FP}) \left(\frac{1}{1 + e^{5y}} \right) - 1. \tag{2.1}$$

Por tanto, las detecciones al final del intervalo ($y = 0$) se evalúan con 0, y una detección justo después de un intervalo recibe una pequeña penalización. Entonces, la puntuación inicial por cada conjunto de datos se define como la suma de las puntuaciones de las detecciones en las ventanas positivas (Y_d) más el producto de A_{FN} y el número de ventanas perdidas f_d:

$$\text{Score} : \sum_{y \in Y_d} \sigma^A(y) + A_{FN} f_d. \tag{2.2}$$

Este *benchmark* está sujeto a ciertas críticas hechas por Singh y Olinsky [SO17]. Algunas de estas críticas se refieren a la falta de aplicación práctica del sistema en un escenario de aplicación real de flujo de datos, pues el supuesto buen funcionamiento de los algoritmos no es suficiente para aplicaciones reales. Además, sería necesario

un mayor detalle en la definición de la función de puntuación para determinados escenarios. En concreto, en el análisis de este artículo se muestra cómo la función de puntuación premia de manera excesiva evitar los falsos positivos y permite un bajo *recall* de anomalías.

RESUMEN DE MEDIDAS DE CALIDAD Incluimos en la Tabla 2.1 un resumen de los diferentes escenarios y las medidas utilizadas para ellos:

Tarea	Descripción	Evaluación	Fórmula
Clas. de series temporales	Clasificación de series temporales completas.	*Accuracy*	$\frac{\sum_{X \in \mathbf{X}_{test}} \mathbb{I}[c(X)=Y]}{\lvert\mathbf{X}_{test}\rvert}$
	Porcentaje de la longitud de la serie temporal necesitada para la clasificación.	Prontitud	$\frac{\sum_{x \in \mathbf{X}_{test}} \frac{l_x^*}{L}}{\lvert\mathbf{X}_{test}\rvert}$
Regresión	Tiempo hasta la parada.	RMSE	$\sqrt{\sum_{i=0}^{N}(\hat{r}_i - r_i)^2}$
Outlier/ Eventos raros	No supervisado/ Supervisado	AUC	Área bajo la curva ROC
		P@n	$\frac{\lvert\{\sum_{i=1}^{n} \mathbb{I}[y_i=O]\}\rvert}{n}$
Medidas basadas en intervalos	Intervalos supervisados	Precisión/ Exhaustividad	$\alpha E(R_i, P) + (1-\alpha)O(R_i, P)$
NAB	Sistema de puntuación propio. Detección ponderada en ventanas previas a las anomalías.	Puntuación propia	$\sigma^A(y) = \frac{A_{TP}-A_{FP}}{1+e^{5y}} - 1$
			$\sum_{y \in Y_d} \sigma^A(y) + A_{FN} f_d$

Tabla 2.1: Medidas de evaluación para algoritmos de detección de anomalías

2.3.1 *Benchmarking*

Las características propias del problema de detección de anomalías y la influencia de los escenarios de series temporales ha promovido la existencia de *benchmarks* específicos para los algoritmos descritos:

ADBENCH ADBench es un *benchmark* de evaluación ideado para métodos estáticos partiendo de conjuntos de datos desbalanceados donde se trata la clase minoritaria como la clase anómala [HHH+22]. Este banco de pruebas permite la experimentación con algoritmos no supervisados, semisupervisados y supervisados e incluye la posibilidad de añadir ruido a los conjuntos de datos para comprobar la robustez de los algoritmos al mismo.

NAB Como hemos descrito en el apartado anterior el *Numenta Anomaly Benchmark* (NAB) incluye un conjunto de series temporales con anomalías puntuales etiquetadas, además de un sistema de puntuación para la evaluación de algoritmos que premia la detección temprana de las anomalías [LA15].

TIMEEVAL A diferencia de las anomalías puntuales identificadas en el NAB, TimeEval realiza una evaluación sobre secuencias anómalas [WSP22].

2.4 BIBLIOTECAS SOFTWARE

En esta sección se incluyen algunas bibliotecas recientes propuestas para la detección de anomalías en diferentes contextos. Prestaremos especial atención a la biblioteca Merlion al tratar escenarios de detección de anomalías en series temporales multivariantes.

ANOMALIB Anomalib[1] [AAV+22] es una biblioteca *software* enfocada en la detección de anomalías en imágenes. Los métodos que se incluyen en esta biblioteca realizan tareas de clasificación de imágenes y de segmentación, es decir, particionamiento y clasificación de secciones dentro de una imagen. Al igual que en otros escenarios donde se manejan imágenes, los métodos con un mejor desempeño están basados en redes neuronales. Además, los incluidos en esta biblioteca aportan también explicaciones sobre las decisiones tomadas al identificar las partes de la imagen que han influido en dichas decisiones.

1 https://openvinotoolkit.github.io/anomalib/

Las propuestas algorítmicas incluidas en esta biblioteca se tratan de `PatchCore` [RPZ+22], un método que extrae características de imágenes de la clase normal y las almacena para comparar las secciones de imágenes de test, identificando como anomalía cualquiera con alguna región anómala; `CFLOWAD` [GIK22], que entrena una red autoencoder e identifica las regiones anómalas usando decodificaciones a diferentes escalas; y `PaDi` [DSL+21], que pasa las imágenes por una red para trasladarlas al espacio latente, donde calcula la distancia de Mahalanobis a las instancias normales sobre cada región de la imagen a evaluar. Como vemos, los métodos recogidos en esta biblioteca se centran en la clasificación de imágenes, y más concretamente en la identificación de regiones anómalas partiendo de imágenes normales.

La implementación de los modelos se ha realizado con `PyTorch`, y la biblioteca contiene las herramientas necesarias para la visualización, ajuste de parámetros y evaluación tanto de los algoritmos incluidos como de otros que se puedan ajustar a los mismos escenarios.

KATS Kats[2] es la herramienta *software* de Facebook escrita en `Python` enfocada al estudio de series temporales univariantes, en concreto para las tareas de predicción, detección de anomalías y extracción de características. Esta biblioteca también incluye una API (*Application Programming Interface*) para la implementación de algorimtos siguiendo sus directrices y funcionalidad adicional como la generación de series temporales sintéticas.

La biblioteca contiene el algoritmo Prophet para la predicción de series temporales univariantes [TL18] comentado en la Subsección 2.1.6; el algoritmo CUSUM [Rob59] para detección de anomalías o cambios en la serie temporal univariante, y un modelo de extracción de características de series univariantes para su incorporación en modelos de *stacking*.

OTSAD Otsad[3] recoge una serie de propuestas algorítmicas para la detección de anomalías en series temporales univariantes [ICC+20]. Este paquete `R` incluye una funcionalidad para la evaluación del benchmark de Numenta [LA15].

Se recogen algunos de los algoritmos con mejores resultados en la competición en la fecha de publicación de la biblioteca, como son los basados en métodos estadísticos usando ventanas móviles ponderadas [RPL15; CS12], una adaptación de k-NN a

2 https://github.com/facebookresearch/kats
3 https://github.com/alaineiturria/otsad

entornos de series temporales [INB+17], y un detector de anomalías contextuales[4], basado en realizar una partición del espacio en regiones discretas y el almacenamiento de las secuencias, de manera que se identifican como anomalías las instancias que suponen el almacenamiento de información no vista con anterioridad.

ALIBI Alibi[5] es una biblioteca `Python` de detección de anomalías, métodos estadísticos para la detección de *drift* y detección de ataques adversarios [VVK+19].

La detección de anomalías se realiza para escenarios estáticos con algoritmos como *Isolation Forest* [LTZ08], algoritmos basados en *autoencoders* (Subsección 2.1.5) y basados en la distancia de Mahalanobis. Los métodos basados en redes pueden aplicarse también a problemas con imágenes. Para series temporales usa métodos estadísticos como el *Likelihood Ratios* [RLF+19], un método basado en redes LSTM para la predicción de la siguiente secuencia [SVL14], o combinando el espectro residual con una red convolucional [RXW+19].

Para la detección de *drift* se utilizan diferentes test de hipótesis nula y métodos estadísticos para identificar el desplazamiento en la distribución de los datos en entornos de *streaming* [BZA17].

PYOD PyOD[6] [ZNL19] es una biblioteca de algoritmos escrita en `Python` que recoge numerosos algoritmos, tanto clásicos como basados en redes neuronales, principalmente enfocados a escenarios estáticos. Esta biblioteca también incluye una funcionalidad para realizar *benchmarking* con datos de ODDS, una adaptación de *data sets* públicos de la UCI para la detección de anomalías. Esta biblioteca plantea un enfoque similar en la construcción de los modelos a la de `sklearn`, proveyendo a los métodos de funciones para entrenar y que proporcionen probabilidades y etiquetas, así como guardar los modelos.

Los modelos incluidos pertenecen a diferentes familias como métodos probabilísticos como ECOD [LZH+22] o métodos basados en ángulos [KZ+08]; métodos lineales como basados en PCA o SVM [SPS+01]; métodos basados en proximidad como HBOS [GD12] o [BKN+00]. También incluyen métodos basados en ensamblados como XGBOD [ZH18] o *Isolation Forest* [LTZ08] y métodos basados en redes *autoencoders*. En resumen, esta biblioteca ofrece numerosos métodos interesantes para entornos estáticos no distribuidos.

4 https://github.com/smirmik/CAD/
5 https://github.com/SeldonIO/alibi-detect
6 https://github.com/yzhao062/pyod

TSFE-DL TSFEDL[7] [AGL+22] es una biblioteca `Python` para algoritmos de detección de eventos raros, al incluir algoritmos basados en redes neuronales para la clasificación temprana de series temporales multivariantes. Algunos de los algoritmos incluidos son los descritos en la Subsección 2.1.6, aunque el total son 21 algoritmos.

Los modelos están implementados tanto en `PyTorch` como en `Keras`, y se componen de una primera parte con capas convolucionales seguidas por redes neuronales de tipo LSTM o *Gated Recurrent Units* (GRU), para la extracción de características espaciotemporales de alto nivel. Por la naturaleza de estas redes, la biblioteca incluye un caso de estudio para la predicción en series temporales, clasificación de series temporales y detección de anomalías en datos de tráfico de red.

MERLION Merlion[8] [BKL+21] es una biblioteca que recoge propuestas algorítmicas para el estudio de series temporales y que ofrece una interfaz unificada para la predicción o *forecasting* y la detección de anomalías para escenarios univariantes y multivariantes.

Una de las características fundamentales para el estudio que no se incluyen en otras propuestas es la definición de una arquitectura que aporta una interfaz común para algoritmos de predicción y detección de anomalías. Otra característica relevante es la inclusión de conjuntos de datos que se vienen usando en la experimentación de las propuestas algorítmicas de la literatura, de manera que se facilita la experimentación para la evaluación de nuevas propuestas.

Los métodos incluidos en la biblioteca para detección univariante están basados en el residuo espectral [RXW+19], mientras que los métodos para entornos multivariantes están basados en *Random Forest* [LTZ12; GMR+16] o en modelos de *Deep Learning* y redes *autoencoder* [ZSM+18; SVL14; KW14].

Otra contribución relevante de esta biblioteca es la inclusión de un *benchmark* que facilita la tarea de investigación de nuevas propuestas al incluir conjuntos de datos y un diseño de experimentos con los que evaluarlas. Al incluir algoritmos para detección de anomalías y predicción para series uni y multivariantes, los *data sets* y medidas de evaluación cubren los mismos escenarios:

- La predicción de series temporales univariantes es el escenario contemplado que ha disfrutado de mayor atención a lo largo de los años, con lo que ya existen

7 https://github.com/ari-dasci/S-TSFE-DL
8 https://github.com/salesforce/Merlion

benchmarks y la competición M4, con 100000 series temporales de distintos dominios [MSA18].

- Para predicción de series multivariantes se usan varios *data sets* públicos, tomando la primera de las dimensiones como *target*, obteniendo las predicciones usando una ventana deslizante [SSL19].
- Para detección en series univariantes, se utiliza el *benchmark* de Numenta [LA15]. Dicho banco de pruebas tiene una puntuación propia que recibe algunas críticas porque el beneficio de una detección temprana puede enmascarar un elevado número de falsas anomalías [SO17].
- La detección de anomalías en entornos multivariantes se realiza utilizando los conjuntos de la predicción multivariante, de manera no supervisada.

3

JUSTIFICACIÓN

En este capítulo presentamos el problema que justifica esta memoria de tesis.

En el capítulo introductorio hemos indicado que la detección de anomalías presenta varias diferencias con la categorización clásica de los modelos de aprendizaje automático. En concreto, un escenario habitual catalogado como detección de anomalías es el de un problema de clasificación donde no se dispone de la etiqueta de las instancias de entrenamiento. Por ello, planteamos las dos líneas de trabajo incluídas en la tesis:

1. El uso de algoritmos de detección de anomalías capaces de trabajar con grandes volúmenes de datos y a la vez representar el conocimiento extraído de manera que podamos ofrecer explicaciones que ayuden a la toma de decisiones: En el ámbito de la detección de anomalías existen ciertos algoritmos que presentan mecanismos básicos con en cuanto a su funcionamiento y que siguen siendo representativos en el estado del arte. Adicionalmente, existen propuestas que buscan explicar la determinación de una instancia como anómala en base a su grado de anomalía en ciertos subespacios. Por eso, basamos nuestra propuesta en uno de estos métodos clásicos, realizando una adaptación para poder buscar las anomalías en subespacios de una mayor dimensionalidad a la de este algoritmo base, y lo combinamos con una propuesta de un mecanismo de explicabilidad utilizando la representación aprendida por el modelo, además de proporcionar una implementación para entornos distribuidos.

2. La evaluación de los algoritmos de detección de anomalías en escenarios temporales: Un escenario de uso habitual de los algoritmos de detección de anomalías es el mantenimiento predictivo, en el cual algoritmos identifican observaciones de una serie temporal como anómalas, mientras las anomalías pueden venir dadas como eventos que se tratan de predecir o evitar con las predicciones en los intervalos previos al mismo. Debido a esta inconsistencia en la naturaleza de las predicciones y los eventos a predecir, se propone un *framework* de evaluación de algoritmos de detección de anomalías para series temporales que generaliza la definición de la curva ROC en función de una ventana de precedencia a los

eventos de interés y así también la curva ROC a una superficie que muestra la curva para diferentes ventanas temporales previas al evento.

4

OBJETIVOS

Tras identificar los problemas a abordar, en este capítulo presentamos los objetivos de esta tesis asociados con dichos problemas. La realización de estas tareas parten de un estudio inicial de la detección de anomalías y las propuestas realizadas para la identificación de anomalías en entornos no supervisados y los diferentes esquemas de evaluación de las propuestas. Los objetivos concretos que presentamos en esta tesis son:

1. La propuesta de un algoritmo de detección de anomalías para escenarios estáticos y de series temporales enfocado en la explicabilidad:
   a) El estudio de los métodos de detección de anomalías y la representación aprendida por los modelos.
   b) El estudio de los métodos de generación de explicaciones basados en la representación de los modelos de detección de anomalías.
   c) La implementación distribuida del método diseñado.
   d) La comparación con métodos del estado del arte en *benchhmarks* diseñados tanto para series temporales como para datos estáticos.
2. La propuesta de un modelo de evaluación de algoritmos de detección de anomalías para series temporales:
   a) El estudio de diferentes medidas de evaluación para detección de anomalías tanto para entornos estáticos como para series temporales.
   b) El diseño de una propuesta de un sistema de evaluación para la evaluación de algoritmos en problemas de series temporales donde no haya instancias etiquetadas como anomalías sino que se pretenda predecir eventos anómalos mediante las instancias en una ventana previa.
   c) La generalización en este sistema de métodos populares de evaluación que premian la prontitud en la detección.
   d) La implementación de una versión distribuida para ser capaces de evaluar rápidamente algoritmos en escenarios con grandes volúmenes de datos.

5

ALGORITMO DE DETECCIÓN DE ANOMALÍAS CON HISTOGRAMA MULTINIVEL PARA ENTORNOS *BIG DATA* ENFOCADO EN LA EXPLICABILIDAD

5.1 INTRODUCCIÓN

La detección de anomalías para problemas no supervisado supone un reto para los investigadores al no estar disponibles las etiquetas en la fase de entrenamiento de los algoritmos. Esto hace que la frontera que separa las instancias anómalas de las normales en la fase de predicción resulte difusa, una circunstancia que puede parecer resuelta en la fase de diseño de algoritmos por el uso de *benchmarks* y conjuntos etiquetados para la evaluación de las propuestas. Sin embargo, a menudo nos enfrentamos a escenarios reales, con grandes volúmenes de datos no etiquetados, y podemos comprobar que una etiqueta, o incluso una puntuación de anomalía que dicotomicemos posteriormente mediante un umbral, es una información insuficiente para considerar que hemos extraído conocimiento del problema, pues no se pueden asociar las etiquetas de las observaciones a eventos concretos o problemática conocida en la base de datos.

Esta circunstancia es la que se da en el proyecto MAPRE de colaboración con Navantia y el CESADAR sobre mantenimiento predictivo para detección de anomalías en grandes volúmenes de datos enfocada en la explicabilidad de las anomalías encontradas y ha motivado la inclusión en este capítulo de una propuesta algorítmica de detección de anomalías basada en histogramas multinivel. Con este algoritmo y su implementación distribuida se propone el uso de histogramas multinivel para realizar una búsqueda eficiente de anomalías en subespacios de dimensión mayor que uno y a la vez mantener una representación del modelo aprendido que permita extraer explicaciones sobre la determinación de una instancia como anómala.

5.2 PROPUESTA

Nuestro objetivo para la realización de la propuesta ha consistido en la obtención de un método capaz de encontrar anomalías en combinaciones de características, de manera que podamos explicar la identificación de las anomalías como tal mediante las variables que han influido en esta decisión. Además el algoritmo debe poder adaptarse para su ejecución distribuida para poder manejar grandes volúmenes de datos. Un enfoque existente que permite abordar el problema de detección de anomalías desde estas dos perspectivas es el algoritmo HBOS [GD12]. Por una parte, la creación de histogramas univariantes y la asignación de puntuaciones de anomalías según la frecuencia permite la combinación de características para la construcción de la puntuación de anomalías en subespacios de más de una dimensión y realizar el cómputo de estos histogramas de manera distribuida. Por otra, esta asignación de puntuaciones atendiendo a la localización en proyecciones de menor dimensionalidad permite calcular la influencia de estas variables e incluirlas en el mecanismo de explicabilidad.

El algoritmo propuesto recibe el nombre de *Histogram-based Outlier Score - MultiLevel* (HBOS-ML) al ser una adaptación del algoritmo clásico que incluye la combinación de los histogramas univariantes en subespacios de mayor dimensionalidad. La ventaja directa es la posibilidad de identificar en estos subespacions puntos anómalos que son normales en entornos univariantes, con la desventaja de un crecimiento cuadrático del número de histogramas según el número de variables, resultando en $\binom{m}{k}$ histogramas, con m el número de variables del conjunto de datos y k el número máximo de variables escogidas para la realización del subespacio. Frente a este crecimiento del número de histogramas, será necesaria la selección de las proyecciones que realmente aporten a la identificación de anomalías, un problema abordado también por el algoritmo OUTRES [MSS11]. Este algoritmo construye un retículo de proyecciones y busca las proyecciones relevantes para la identificación de instancias anómalas. Por una parte, las proyecciones donde las instancias están agrupadas no aportan información al no haber instancias discordantes. Esta será una circunstancia que se dé principalmente en subespacios de dimensionalidad menor. En cambio, en los subespacios de mayor dimensionalidad, por la conocida maldición de la dimensión (*dimensionality curse*), las distancias entre observaciones seguirán una distribución uniforme y por tanto estas proyecciones no aportarán tampoco a la identificación de instancias anómalas. El enfoque aquí, puesto que lo que disponemos es de los histogramas por cada combinación, será el de comparar este histograma con las

observaciones que obtendríamos en una muestra de una distribución multinomial equiprobable, donde cada clase se corresponde con cada intervalo k-dimensional. En el caso de que mediante la realización de un test χ^2 se rechace la equivalencia de las distribuciones, se incluirá la puntuación de ese subespacio en el cómputo de la puntuación para las instancias.

Algorithm 1: Pseudocódigo HBOS-ML

Data: Número de atributos a agrupar k, número de intervalos n_{bins}, *data set* $X \in \mathrm{R}^{n \times m}$
Result: *Scores*
Estandarización de las características en el intervalo $[0, 1]$
Construcción de una matriz $D \in \{1, \ldots, n_{bins}\}^{n \times m}$ para la posterior reconstrucción de histogramas multivariables:
for $\{i = 1, \ldots, n\}$ **do**
  for $\{j = 1, \ldots, m\}$ **do**
    $d_{i,j} \leftarrow$ Intervalo correspondiente a la localización de $x_{i,j}$
  end
end
for $c \in \{\binom{k}{m}\}$ **do**
  Construcción del histograma k-dimensional H^c con las variables incluidas en la combinación c partiendo de D
end
for $\{i = 1, \ldots, n\}$ **do**
  for $c \in \{\binom{k}{m}\}$ **do**
    $Score_i = Score_i + \log 1 / H^c_{d^c_i}$
  end
end

5.2.1 *Adaptación a entornos distribuidos*

De cara a la distribución del cómputo, la estrategia seguida ha consistido en realizar el proceso completo mediante pasos independientes y reutilizables de manera que se puedan aplicar a particiones de los datos de forma independiente del cómputo para el resto.

El principal paso para conseguir esta reusabilidad es la construcción de un *data set* paralelo que contiene el intervalo en el que se encuentra cada observación para cada variable. Una vez calculado el intervalo para cada variable y observación, para cada combinación de variables estudiada se construirá el histograma realizando un conteo y la obtención de la puntuación para cada intervalo k-dimensional. De esta manera, en el último paso únicamente se tiene que asignar a cada observación la puntuación del intervalo en el que se encuentra para cada combinación de variables.

Todas estas operaciones son, o bien modificaciones del conjunto de entrada que no dependen de otras observaciones, como la construcción del *data set* con el intervalo o el cálculo de la puntuación de anomalía, o bien operaciones fácilmente asimilables al paradigma *Map-Reduce*, como el cálculo del histograma, que se puede hacer para cada instancia independientemente y posterior agregación con la suma de los histogramas para cada observación.

5.2.2 *Inclusión de mecanismo de explicabilidad*

En esta sección se describe la propuesta para la inclusión de un mecanismo de explicabilidad dentro del algoritmo de detección de anomalías en base a histogramas multivariantes.

5.2.2.1 *Problemática en el ámbito no supervisado*

Al igual que nos encontramos a la hora de evaluar los algoritmos de detección de anomalías, en el ámbito de la explicabilidad, la mayor parte de las propuestas se centran en escenarios supervisados. Además, las pocas propuestas existentes para clasificación no supervisada o detección de anomalías generan una representación gráfica, que es insuficiente para abordar problemas de naturaleza *Big Data*, además de ser propuestas *post-hoc* [XXM+19], por lo que necesitarían ser capaces de aproximar algoritmos distribuidos con mecanismos que no necesariamente lo son. Por ello, se puede justificar la necesidad de un mecanismo específico implementado en el ámbito distribuido basado en reglas.

Una de las propuestas más relevantes para la explicabilidad basada en reglas es la propuesta de Guidotti *et al*. [GMR+18], la cual presenta un modelo *a posteriori* para la aproximación basada en reglas locales en base a la generación mediante un algoritmo genético de elementos representativos de la frontera de decisión. Con ello, se permite la construcción de un árbol de decisión de poca profundidad que dé

simultáneamente una explicación de por qué se ha tomado esa decisión, y una serie de *counterfactuals* o *contrahechos*, consistentes en los cambios mínimos de ese árbol de decisión que llevarían a la decisión contraria, como vimos en el capítulo anterior.

PROPUESTA En nuestro caso, partimos del algoritmo implementado para la inclusión del mecanismo de explicabilidad, sin necesidad de recurrir a una propuesta *post-hoc*. Como se ha comentado, muchos algoritmos se basan en la construcción o búsqueda de instancias representativas de las fronteras de decisión. Por la naturaleza *Big Data* del algoritmo y la representación de la información en forma de histogramas, se opta por usar dichos histogramas y la selección de variables representativas para la construcción del árbol de decisión y obtención de los contrahechos que identifiquen las causas de la identificación de la instancia como anomalía.

Llamaremos E al espacio de explicaciones interpretables por un humano, b al algoritmo de decisión, x a la instancia a explicar e y a la etiqueta o puntuación de anomalía, de manera que buscamos una explicación $e \in E$ que explique $b(x) = y$.

Definición 1 (Explicación) *Definimos los componentes de una explicación como un par de objetos:*

$$e = \langle r = p \rightarrow y, \Phi \rangle.$$

El primer componente es una regla de decisión que describe el razonamiento para el valor de decisión $y = b(x)$. El segundo componente es un conjunto de contrahechos, es decir, el mínimo número de cambios que debe hacerse en los valores de entrada de manera que cambie la decisión tomada para x.

En una regla de decisión p, asumimos que p es una conjunción de condiciones de particionamiento sc de la forma $[a \in v_1, v_2]$, donde a es una característica y v_1, v_2 son valores en el dominio de a. Diremos que una regla $r = p \rightarrow$ cubre a x si $sc(x)$ es cierto para cada $sc \in p$, o que x satisface p.

Cuando la instancia x satisface p, la regla $p \rightarrow y$ representa una *motivación* para la toma de una decisión, es decir, p explica localmente por qué b devuelve y. Para relacionar el cumplimiento de las reglas, diremos que r es α consistente con c si $c(x) \in [y - \alpha, y + \alpha]$ para todo x que satisfaga r. La inclusión de un umbral viene dada porque estamos enfocado a mantener todo lo posible el *score* de anomalía y y la minimización de las reglas a introducir en la consulta r, por lo que así permitimos

pueda cambiar y dentro de unos márgenes si no se modifican las características de la regla.

Consideramos ahora δ unas condiciones de particionamiento. Denotamos la modificación de p por δ como:

$$p[\delta] = \delta \cup \{(a \in [v_1, v_2] \in p| \nexists [w_1, w_2], (a \in [w_1, w_2]) \in p\}.$$

En definitiva, es el cambio de al menos un atributo de p.

Definición 2 (Contrahecho) *Una regla contrafactual, o de contrahechos, es una regla de la forma $p[\delta] \rightarrow \hat{y}, \hat{y} \neq y$. Llamaremos contrahecho a δ.*

Los contrahechos que esperamos obtener serán mínimos con respecto a la instancia x, es decir, cambiando el mínimo número de condiciones de x: $nf(p[\delta], x) = |\{sc \in p[\delta]| \notin sc(x)\}|$.

Definición 3 (Explicación local) *Sea x una instancia, y $b(x) = y$ una decisión de b. Una explicación local $e = \langle r, \Phi \rangle$ es un par formado por una regla de decisión, α consistente con c y satisfecha por x; y un conjunto $\Phi = \{p[\delta_1] \rightarrow \hat{y}_1, \ldots, p[\delta_v] \leftarrow \hat{y}_v\}$.*

Una vez descrita la notación con respecto a las reglas, pasamos a la formalización de la propuesta en los términos establecidos. Para ello, comenzamos con definir cuáles son las variables que nos importarán en la definición del conjunto de reglas que conformen la explicación local.

Dada la instancia $x_i \in \mathbb{R}^k$ el histograma $\mathbf{H}$ calculado, las posiciones en el histograma $d_{i,\cdot}$, denominamos $\mathcal{E}_\alpha$ asociada a la instancia como la α esfera alrededor de x_i:

$$\begin{aligned}\mathcal{E}_\alpha(x_i) = \big\{(d_q, j) : |s_{q,j} - s_{d_{i,j},j}| \leq \alpha, j = 1, \ldots, k,\\ q \in \{1, \ldots, n_{bins}\} - \{d_{i,j}\}\big\}\end{aligned}$$

Una vez tenemos la mayor regla α consistente con el algoritmo, buscamos la regla mínima podando aquellas variables que el modelo ha seleccionado como irrelevantes, además de las que incluyan todas las posibles q. Si este conjunto fuera demasiado grande como para representar una explicación clara, se podrían seleccionar las variables para las que el conjunto de pares $(intervalo, variable)$ fuera menor dentro de $\mathcal{E}_\alpha$, así como comprobar que los intervalos sean consecutivos.

Una vez formada la esfera, podemos formar la regla contrafactual basándonos en los pares que quedan fuera de la α esfera de entre las variables que aún permanecen, pues serán los que cambian el score de x_i en más que α.

Esta es una definición válida tanto si la instancia ha sido considerada como anómala como si es normal, una cuestión que sólo se puede concretar si se ha definido un umbral de anomalía. Es en este momento cuando, si lo que queremos es determinar cuándo pasa de ser anómala a no serlo, fijaremos α como la diferencia entre el umbral y la puntuación actual de anomalía. De esta manera, hemos postergado la concretización de un parámetro que es imprescindible para el funcionamiento de los métodos supervisados, y que en nuestro caso podremos fijar según la necesidad y la información que queramos obtener.

5.3 EXPERIMENTACIÓN

Para la experimentación usaremos dos *benchmarks* para testar algoritmos de detección de anomalías tanto en datos estáticos como en datos temporales.

5.3.1 *Anomaly Detection Benchmark*

El primer conjunto de pruebas utilizado será *Anomaly Detection Benchmark* (ADBench) [HHH+22]. Este *benchmark* está diseñado para la detección de anomalías en datos estáticos y tiene como apoyo la biblioteca PyOD [ZNL19] para la ejecución de los algoritmos. Además del escenario para la comparación de algoritmos no supervisados, este trabajo también incluye funcionalidad para realizar la comparación según la disponibilidad de etiquetas, bien para algoritmos supervisados o para algoritmos de detección de novedad. Otra aportación es la de la clasificación de los conjuntos de datos según las características de las instancias anómalas para la identificación del mejor escenario para cada algoritmo y una funcionalidad para probar los algoritmos con distintos niveles de ruido.

5.3.1.1 *Conjuntos de datos*

Este *benchmark* dispone de 57 conjuntos de datos, haciendo uso de los existentes en repositorios como ODDS [Ray16], que adaptaba conjuntos de datos del repositorio de datos de UCI para entornos de detección de anomalías, partiendo de conjuntos desbalanceados y tratando la clase minoritaria como la clase positiva.

5.3.1.2 *Algoritmos*

La implementación de los algoritmos es la incluida en la biblioteca PyOD. Incluimos aquí algunos algoritmos involucrados en la comparación.

PCA Clasificador basado en componentes principales.

OCSVM Clasificador basado en *Support Vector Machines* que considera el conjunto de entrenamiento como la clase normal para proyectar los puntos en un espacio de dimensión superior en el que poder separar las clases. En esta adaptación para detección no supervisada se asume la existencia de una cierta contaminación con instancias que quedarán fuera de esta hipersuperficie.

k-NN Método basado en vecinos más cercanos.

LODA Método basado en distancias usando proyecciones aleatorias.

COF Este método basado en el esquema de LOF [TCF+02]

CBLOF Este algoritmo parte de una *clusterización* para calcular un grado de anomalía dentro del clúster [HXD03].

ISOLATION FOREST Método basado en la construcción de árboles que identifican como anómalas instancias que se separan fácilmente del resto.

HBOS Método basado en histogramas que sirve como base para la realización de la propuesta.

COPOD Método basado en la agregación de una puntuación según la función de distribución acumulada para todas las variables [LZB+20].

DEEPSVDD Método para detección de novedad basado en redes de *deep learning* basado en el aprendizaje de una función de transformación para la representación de los datos en una hiperesfera de volumen mínimo [RVG+18].

5.3.1.3 *Resultados*

La Figura 5.1 muestra un resumen de las puntuaciones obtenidas por los algoritmos en los distintos conjuntos de datos. Las medidas utilizadas han sido el área bajo la curvva tanto de la curva de precisión frente a exhaustividad como de la curva

ROC. Para ambas medidas el método obtiene peores resultados que los métodos del estado del arte, además de ser el único método junto a DeepSVDD que no obtiene puntuaciones cercanas al máximo en ningún conjunto de datos.

Para realizar una comparación estadística general de los métodos nos basamos en el gráfico de diferencias críticas (*Critical Differences Plot*, o CDPlot). Este método realiza un test de hipótesis sobre el ránking medio de cada algoritmo sobre todos los conjuntos de datos y agrupa aquellos para los que la hipótesis nula, consistente en la equivalencia de los métodos, no ha sido descartada por el test. Además, se ordenan de manera gráfica, pudiendo ver el ránking medio de cada algoritmo. De esta manera, poodemos comprobar que efectivamente HBOS-ML obtiene los peores resultados (muy cerca de DeepSVDD en la comparación con respecto al AUC de la ROC), aunque la equivalencia no puede ser descartada con respecto a los algoritmos LODA, COF y LOF.

5.3.2 *TimeEval: Time Series Anomaly Detection Benchmark*

La biblioteca TimeEval [WSP22] proporciona diferentes herramientas para la experimentación de algoritmos de detección de anomalías para series temporales. En primer lugar, recoge numerosas series temporales etiquetadas, tanto univariantes como multivariantes, además de proporcionar la API GutenTAG para la generación de conjuntos de datos sintéticos con el tipo de anomalía temporal deseado y una API para la evaluación de manera gráfica tanto de adaptaciones propias de algoritmos clásicos a entornos temporales como de nuevas propuestas.

5.3.2.1 *Conjuntos de datos*

TimeEval incluye 1354 conjuntos de datos agrupados en 24 colecciones atendiendo a su origen y naturaleza. En esta prueba usaremos únicamente los conjuntos reales multivariantes por la naturaleza del algoritmo. Las colecciones usadas en la experimentación son CalIt2, Daphnet, MITDB, SVDB y SMD, haciendo un total de 182 conjuntos de datos.

5.3.2.2 *Algoritmos*

Se incluyen en esta experimentación versiones adaptadas para trabajar con series temporales de los algoritmos PCA, LOF, *k*-NN, IsolationForest, HBOS, COPOD y CBLOF comentados en la subsección previa.

5.3.2.3 *Resultados*

Se incluyen en la Figura 5.3 los resultados de la experimentación en el *benchmark* de TimeEval, incluyendo como medidas el área bajo la curva ROC y el área bajo la curva adaptada a intervalos, una medida más propicia para detección de anomalías en entornos de series temporales [TLZ+19]. Al igual que en el *benchmark* ADBench, los resultados de HBOS-ML en TimeEval distan de ser asimilables a los de los algoritmos del estado del arte. En este caso sí hay mayor semejanza en cuanto a los resultados observado en el *boxplot*, aunque en el CDPlot mostrado en la Figura 5.4 constatamos que sigue quedando en última posición, en este caso sin poder descartar la equivalencia con los algoritmos HBOS, LOF y COPOD con respecto al AUC con respecto a la precisión y la exhaustividad basada en intervalos y en la penúltima posición, con unos resultados asimilables al algoritmo de detección basado en PCA, COPOD y LOF con respecto a los resultados de AUC-ROC.

5.4 CONCLUSIONES

En este capítulo hemos comprobado la necesidad de la inclusión de mecanismos de explicabilidad en el diseño de modelos de Inteligencia Artificial, especialmente en escenarios donde la explicabilidad del modelo puede ofrecer un conocimiento no disponible *a priori*, como es el caso de los escenarios no supervisados. Esta falta de información es un problema aún más acuciante en entornos *Big Data*, puesto que el gran volumen de datos suele generar modelos complejos.

La contraprestación a esta funcionalidad de explicabilidad se aprecia en los resultados, con un rendimiento inferior al de los algoritmos del estado del arte, por lo que seguiremos trabajando en esta propuesta para mejorar los resultados aprovechando el conocimiento adquirido en el proyecto MAPRE sobre datos temporales de gran volumen no etiquetados.

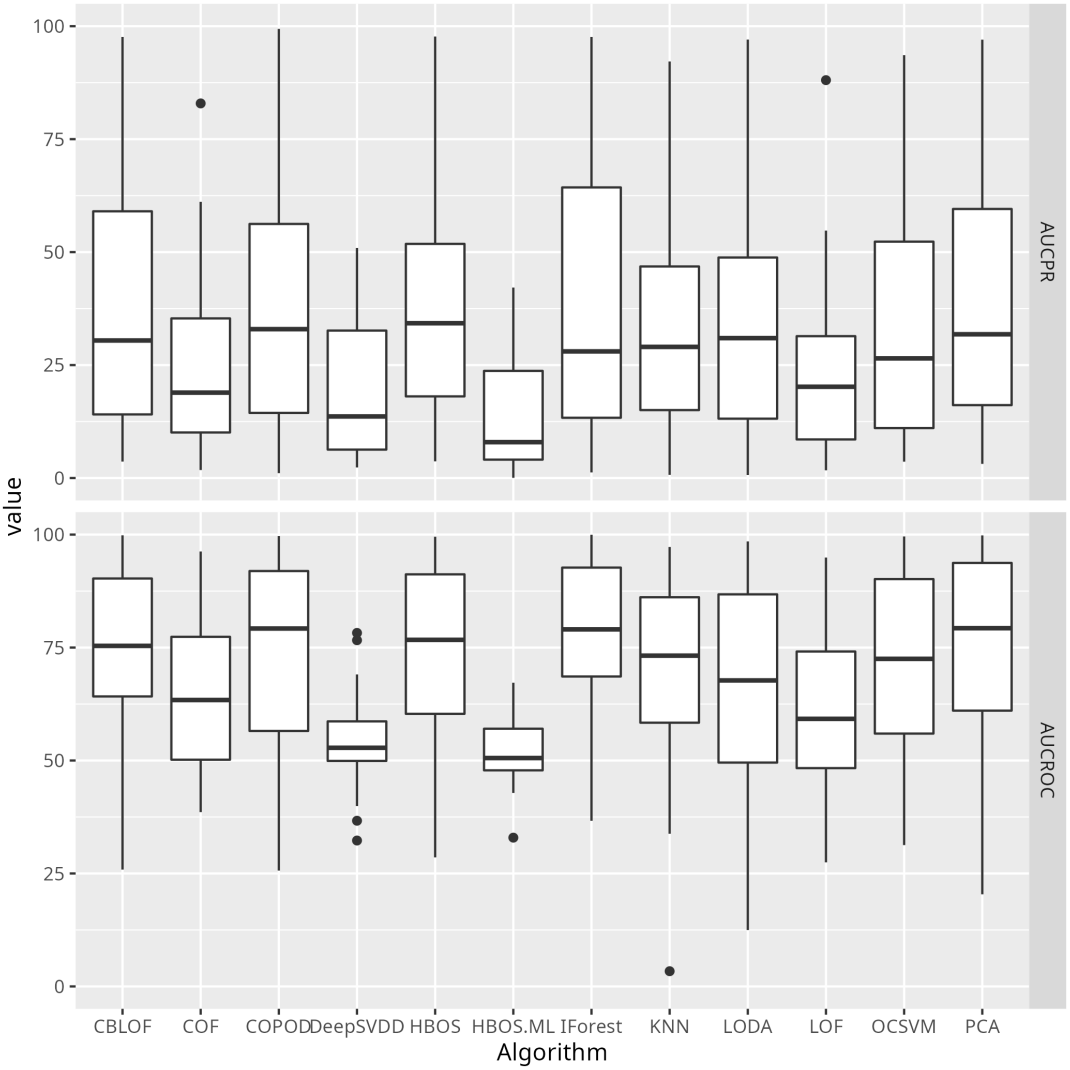
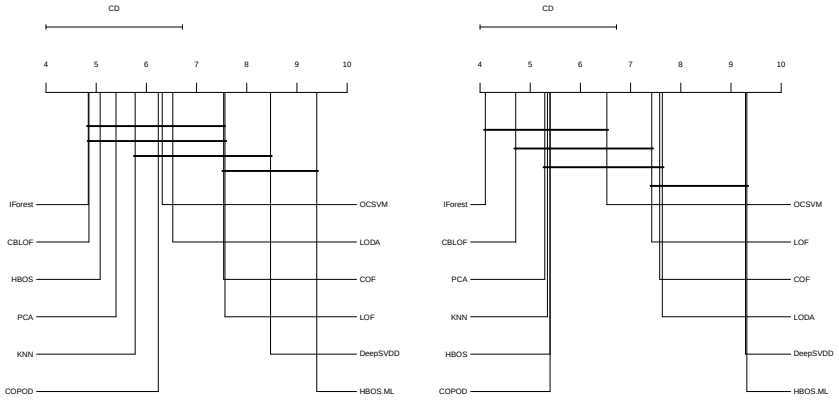


Figura 5.1: Boxplot con resultados en PR-AUC y ROC-AUC



(a) Comparación RangePR-AUC (b) Comparación ROC-AUC

Figura 5.2: CD Plot en ADBenchh

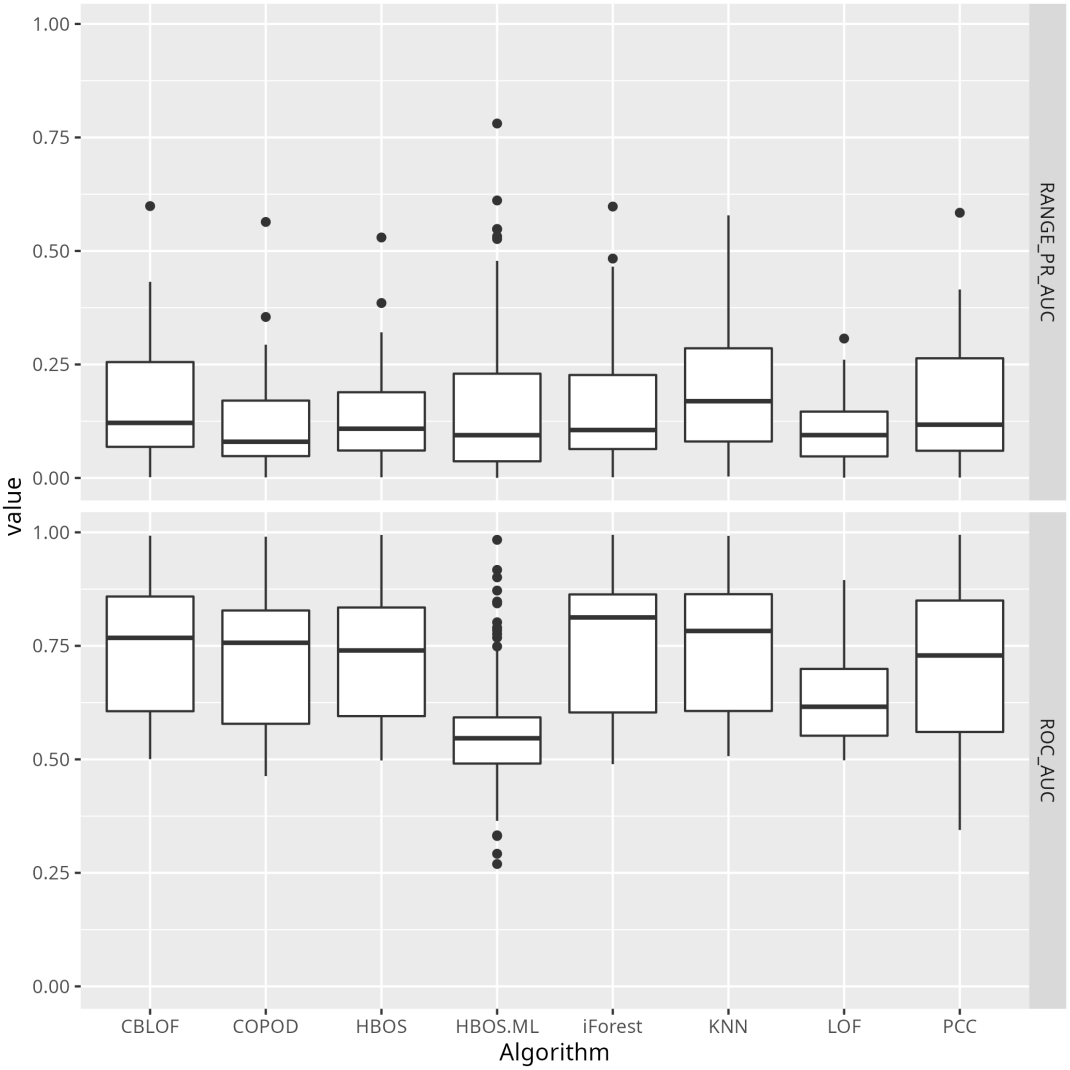
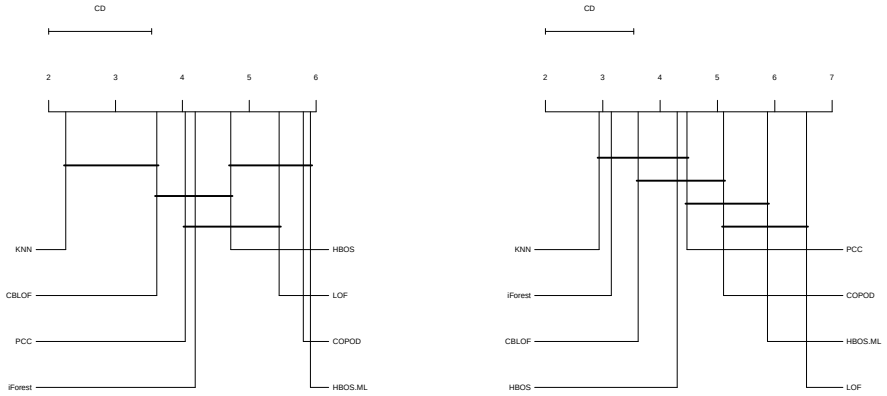


Figura 5.3: Resultados TimeEval en Range-PR-AUC y ROC-AUC



(a) Comparación RangePR-AUC (b) Comparación ROC-AUC

Figura 5.4: CD Plot en TimeEval

6

DETECCIÓN DE ANOMALÍAS EN MANTENIMIENTO PREDICTIVO: UN NUEVO SISTEMA DE EVALUACIÓN PARA DETECCIÓN DE ANOMALÍAS EN ENTORNOS TEMPORALES NO SUPERVISADOS

6.1 INTRODUCCIÓN

La denominación de detección de anomalías se usa para referirse a problemas de diferente naturaleza y casos de uso [CBK09]. Por ejemplo, las anomalías en series temporales propias del mantenimiento predictivo o detección de fallos [WM19], procesamiento de datos en seguridad [ZG20] o en grafos en datos provenientes de redes sociales [NHH18]. En otros escenarios, los nombres utilizados son detección de valores atípicos, excepciones, eventos raros o detección de novedad, dependiendo de la intención del estudio o del campo de investigación. La existencia de múltiples contextos viene aparejada a diferentes modelos de evaluación de las propuestas, lo que puede significar una incorrecta evaluación de las mismas.

Las diferencias en la nomenclatura utilizada para referirse a los escenarios de investigación ha sido abordada por otros investigadores [CIL19]. En su taxonomía, proponen *eventos raros* para la clasificación supervisada de series temporales donde existe un desbalanceo de clases y la tarea de clasificar estas series temporales en clases conocidas [TYK+16]. Por otro lado, el amplio término de anomalía se reserva para la clasificación supervisada de datos no temporales con un alto desbalanceo entre las clases [RPG16]. En el esecenario semisupervisado [Agg17], donde sólo se dispone de datos normales en el período de entrenamiento, algunos autores usan el término de *clasificación de una clase* [FFA20], mientras que otros prefieren el término *detección de novedad*, remarcando el interés en las instancias no vistas [OCF08]. Otro término ampliamente extendido es el de detección de valores atípicos [Agg17], habitualmente asociado a la clasificación no supervisada y frecuentemente en relación con el término *ruido*, más relacionado con las observaciones que difieren del normal comportamiento, pero no lo suficiente como para pensar que han sido producidas por un mecanismo distinto al de la generación de las instancias, lo que es en última instancia la tarea de los algoritmos de detección de datos anormales.

En el caso específico del mantenimiento predictivo, como un caso especial de detección de anomalías para series temporales y la situación considerada en este trabajo, los eventos de interés se representan por puntos instantáneos que requieren una intervención. Una circunstancia similar ocurre con los algoritmos de detección de anomalías en series temporales: se evalúan observaciones que ocurren en un instante para predecir un evento que no sólo ocurre en una instancia sino en el intervalo posterior.

Como era de esperar, los escenarios mencionados anteriormente usan medidas específicas y configuraciones de experimentación para validar sus resultados. Por un lado, la mayoría de estas medidas provienen de escenarios supervisados desbalanceados, como la exhaustividad o *recall* de la clase minoritaria o las nuevas clases, o alguna combinación de la precisión (*precision*) y el *recall*, como la medida-*F*, o medidas derivadas de la curva ROC (*Receiver Operating Characteristic*), como el Área Bajo la Curva (AUC, por *Area Under the Curve*) [GU16]. Por otro lado, para esos escenarios donde el foco principal es la componente temporal, las propuestas incluyen una medida para la prontitud en la detección [LA15] con el objetivo de minimizar la probabilidad de un fallo.

En resumen, existen múltiples escenarios para la detección de comportamiento anormal que son referidos por el nombre de detección de anomalías y múltiples medidas que tienen inconvenientes asociados con respecto a la tarea de mantenimiento predictivo:

- Los beneficios del enfoque como un problema desbalanceado se ven diluido por la omisión de la componente temporal.
- La evaluación de la prontitud en la detección se agrega para una mejor comprensión a través de la inclusión de parámetros, dificultando la tarea de la adaptación a escenarios reales.

Para compensar estos elementos, recogemos los diferentes métodos para la detección de eventos anómalos, unificando los esquemas de evaluación y las implicaciones de la asignación de las clases a las instancias en un nuevo método de evaluación para detección de anomalías no supervisada donde los eventos de interés guardan una relación incierta con los datos. La definición del modelo propuesto comienza con una ventana temporal que precede los eventos de interés. La consideración general es que una instancia positiva es detectada cuando el algoritmo lanza una alarma dentro de dicha ventana. Sin embargo, esto provoca las siguientes preguntas: ¿Cuál es el límite

superior de la ventana? ¿Cuál es la utilidad de ventanas tan grandes que agrupen a todas las instancias? Teóricamente, podríamos considerar la totalidad de la serie temporal como la ventana previa, en cuyo caso sólo tendríamos una instancia positiva relativa al último evento registrado, e instancias negativas posteriores a esta última observación positiva, lo que carece de sentido para la realización de un estudio, pues los detectores aletorios se beneficiarían del método de evaluación. Podemos resumir el método propuesto basándonos en los siguientes componentes:

- En primer lugar, definimos una transformación de las instancias, de observaciones instantáneas a intervalos, con una agregación que produzca una visión más amplia del evento de una anomalía a través de la inclusión de un parámetro para la longitud de la ventana que precede cada evento anotado.
- Entonces, describimos cuáles son las opciones para la agregación y cuáles son las implicacionees, incluyendo una agregación que tenga en cuenta la prontitud en la detección derivada del *Numenta Anomaly Benchmark* (NAB).
- El siguiente componente es la ROC basada en la ventana de precedencia (pw-ROC, por *preceding window ROC*), que generaliza la definición de la ROC clásica para incluir el parámetro de la longitud de la ventana descrita anteriormente. La agregación de estas pw-ROC conforma la superficie ROC, donde el parámetro de la longitud de la ventana representa la tercera dimensión.

Por tanto, el proceso completo genera una figura que proporciona la información sobre la calidad del algoritmo, no solo en los distintos niveles del umbral para asignar la etiqueta de anomalía, sino también a diferentes niveles de distancia hasta el siguiente evento de interés. Se han implementado dos versiones de esta propuesta: una clásica usando `python`, y otra versión usando `pySpark` para ser capaces de manejar problemas de series temporales con grandes volúmenes de datos. Ambas versiones están disponibles en GitHub[1].

Para validar la utilidad de la propuesta, incluimos la evaluación usando este método de tres algoritmos del estado del arte y examinamos sus resultados usando un conjunto de datos proporcionado por ArcelorMittal[2]. También evaluamos los algoritmos utilizando un sistema de puntuación para rangos anómalos y comparamos esta evaluación con nuestro método para analizar los beneficios de la propuesta. La

1 https://github.com/ari-dasci/S-pwROC
2 https://corporate.arcelormittal.com/

popularidad del escenario, donde el uso de algoritmos no supervisados es preferible por la ausencia de seguridad sobre los posibles eventos que surgen, aunque los eventos de interés puedan anotarse a través de observaciones para la aplicación de algoritmos, muestra la fiabilidad del método de evaluación propuesto.

En resumen, las principales contribuciones de este trabajo son:

- Describimos las características distintivas del problema de detección de anomalías para escenarios de series temporales de mantenimiento predictivo.
- Proponemos un método de evaluación para los escenarios descritos con un *software* asociado para series temporales de *Big Data*.
- Incluimos un caso de estudio con una comparación entre la salida del método propuesto con una propuesta para un escenario análogo.

6.2 FRAMEWORK DE EVALUACIÓN PARA DETECCIÓN DE ANOMALÍAS TEMPORALES NO SUPERVISADA.

En esta sección describimos la propuesta de un *framework* de evaluación para detección de anomalías en series temporales de escenarios frecuentes en la literatura, de manera que los investigadores dispongan de una medida relevante que se ajuste a sus datos. Como hemos mencionado en la sección anterior, a menudo las anomalías en series temporales se etiquetan como puntos individuales y los sistemas de puntuación promueven una detección temprana. Sin embargo, a estos sistemas les falta un mecanismo de balanceo que tenga en cuenta la importancia relativa de las instancias positivas. De esta manera, los sistemas actuales sobreestiman la relevancia de la detección temprana en detrimento de una mayor cantidad de falsos positivos o falsos negativos.

Nuestra intención es introducir el mecanismo de la curva ROC, ya que tiene en cuenta la relación entre la TPR y la FPR para los posibles umbrales. Un beneficio adicional de usar la curva ROC es la posibilidad de trabajar con puntuaciones de anomalía en lugar de etiquetas para las instancias predichas. Esta capacidad proporciona más opciones en el diseño de los métodos de combinación.

Esta sección está estructurada de la siguiente manera: En la Subsección 6.2.1 incluimos la definición formal de la transformación propuesta de las instancias temporales en intervalos. El componente de la agregación se describe en la Subsección 6.2.2. La

Subsección 6.2.3 está dedicada a la definición de la propuesta de la pw-ROC y las consideraciones de la superficie ROC resultante.

6.2.1 *Primer componente: Transformación en instancias de intervalos*

Sea $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ una serie temporal y $\{t_1, \ldots, t_N\}$ los instantes temporales para esas instancias. Notamos por $\mathbf{S} = \{s_1, \ldots, s_M\}$ el conjunto de instancias temporales donde ocurren los M eventos de interés. Dependiendo del escenario o caso de estudio, los M incidentes pueden ser marcas de tiempo que representan los eventos o el inicio de intervalos de tiempo anómalos. Sea $W_{max} = \text{mín}_{i=1,\ldots,M-1}\{s_{i+1} - s_i\}$ la longitud máxima de de las ventanas que definen los intervalos positivos.

De aquí en adelante, notamos por w a la longitud escogida para la ventana utilizada para determinar los intervalos de evaluación y las medidas de calidad. Este parámetro está en el intervalo $(0, W_{max}]$. En el caso de hacer $w = 0$ sería preferible utilizar un modelo de evaluación que tenga en cuenta el resultado en cada instancia.

Con respecto al valor práctico de w, podemos definir el inicio del intervalo positivo para un escenario supervisado, de manera general, como un 10 % del período estudiado dividido por el número de anomalías [LA15]. Este valor está sujeto a los intereses de los investigadores y los expertos en el dominio, especialmente para escenarios no supervisados.

Con las definiciones previas, definimos el conjunto de intervalos positivos y negativos que representan las instancias agregadas, que contienen a las observaciones originales $\mathbf{x}_j, j = 1, \ldots, N$:

$$\begin{aligned}\mathcal{I}_w = \big\{\{\mathbf{x}_j : s_{i-1} < t_j; s_i - (k+1)w < t_j \leq s_i - kw\} : \\ i = 1, \ldots, M; k \in \mathbb{N}_0\big\},\end{aligned} \tag{6.1}$$

con $s_0 = \text{mín}\{s_1, t_1\}$, no definido de otra manera para $i = 1$, y $\mathbb{N}_0$ correspondiendo con los números naturales y el cero.

A partir de estas instancias, podemos definir de una manera más simple las instancias positivas como un subconjunto de $\mathcal{I}$, las cuales son las instancias a una distancia menor a w de un evento:

$$\mathcal{P}_w = \big\{\{\mathbf{x}_j : 0 \leq s_i - t_j < w\} : i = 1, \ldots, M\big\}, \tag{6.2}$$

y las instancias negativas como aquellas a una mayor que w (por la derecha) de un incidente:

$$\mathcal{N}_w = \big\{\{\mathbf{x}_j : s_{i-1} < t_j; s_i - (k+1)w < t_j \leq s_i - kw\} : \\ i = 1, \ldots, M; k \in \mathbb{N}\big\}. \tag{6.3}$$

A partir de estas definiciones, es claro que $\mathcal{I}_w = \mathcal{P}_w \cup \mathcal{N}_w, \mathcal{P}_w \cap \mathcal{N}_w = \emptyset$. Por mayor simplicidad, notamos $\mathcal{I}_w = \{X_l, l = 1, \ldots, p\}$, donde cada X_l representa una agregación de las instancias originales. Es importante notar que estas definiciones permiten la aplicación de la evaluación a entornos con instancias no muestreadas de manera uniforme en el tiempo. En Figura 6.1 se muestra la partición $\mathcal{I}_w$ de una serie temporal en las instancias consideradas usando el color de fondo, el cual representa a los elementos de $\mathcal{I}_w$. Las ventanas anómalas (los elementos de $\mathcal{P}_w$) se marcan con una línea roja y los eventos marcados se indican con un punto rojo.

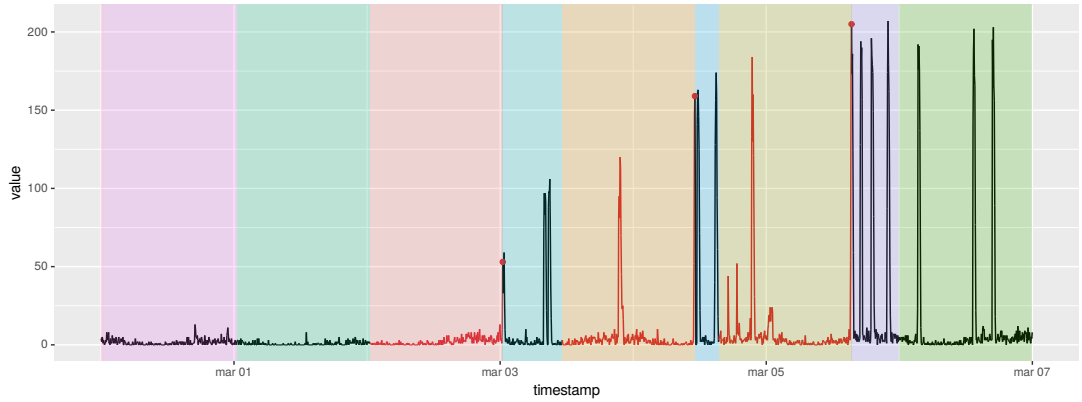


Figura 6.1: Partición por intervalos de la serie temporal basada en las anomalías.

6.2.2 *Segundo componente: Funciones de agregación y puntuación de prontitud*

Una vez que las instancias temporales se han agregado en intervalos, proponemos el uso de una función f para agregar la puntuación de anomalía proporcionada por el algoritmo A para esos intervalos:

$$f(X_l) = f\left(\{A(\mathbf{x}) : \mathbf{x} \in X_l\}\right). \tag{6.4}$$

Hay algunas opciones para la definición de f que depende en el escenario de aplicación y los algoritmos utilizados. Aquí incluimos las principales opciones:

MEDIA Para aquellos algoritmos que proporcionan una puntuación de anomalía, la media es la función de agregación básica. Los resultados se esperan que sean más representativos para aquellos escenarios con más instancias en cada intervalo.

CCDF El complemento de la función de distribución acumulada (CCDF por sus siglas en inglés), con un umbral, para calcular el porcentaje de instancias con una puntuación de anomalía mayor que cierto umbral. Esta función de agregación puede ser apropiada para escenarios con menos instancias o para tratar con algoritmos que sólo proporcionan una etiqueta en lugar de una puntuación de anomalía. Este agregación con un umbral de 0.5 es equivalente a usar la mediana.

NAB El sistema de ponderación del *benchmark* de Numenta se reformula como una función de agregación, dando menos relevancia a aquellas observaciones demasiado cercanas al evento puesto que es inminente y la predicción de anomalía carece de uso.

$$f_{NAB,w}(X_l) = \sum_{\mathbf{x}_j \in X_l} \sigma_w(\text{time\_until\_next\_alarm}(t_j)) \cdot \text{algorithm}(\mathbf{x_j}),$$

donde $\sigma_w(t) = \frac{2}{1+e^{15t/w}} - 1$. Esto da un peso cercano a uno a aquellas instancias a más de w unidades de tiempo del evento, por lo que podemos ignorar el peso en los intervalos negativos. El coeficiente 15 se deriva del NAB, donde usan un coeficiente de 5 para una ventana de 3 horas.

CONSIDERACIONES PARA EL FILTRADO Para esas instancias que proporcionan una etiqueta de anomalía para cada instancia, especialmente para esas situaciones donde proporcionan una tasa elevada de etiquetas positivas, las funciones de agregación mencionadas previamente pueden conducir a una elevada tasa de falsos positivos. Como hemos descrito en la sección anterior, hay algunas opciones para filtrar las instancias positivas:

VENTANA SIN ALARMA Se define una segunda ventana w_2 definida después de una instancia definida como positiva por el algoritmo, dentro de la cual ninguna otra instancia es declarada como positiva en la agregación [ICC+20].

CONTADOR Podemos considerar una ventana deslizante, donde sólo la última alarma es considerada como anomalía si hay más de un cierto umbral K de alarmas en dicho período.

El uso de estas funciones es independiente de la función de agregación y puede ser considerada como parte de ella a través de la composición $f' = f \circ g$, con g una función filtro y f, f' funciones de agregación.

6.2.3 *Tercer componente: Evaluación basada en la ROC basada en la ventana de precedencia*

La curva ROC puede definirse como una gráfica de la sensitividad frente a $1 -$ especificidad para todos los posibles umbrales c:

$$\text{ROC} = \{(P(y > c|\hat{y} = 0), P(y > c|\hat{y} = 1)) : c \in (\infty, \infty)\}, \tag{6.5}$$

donde y representa la puntuación y $\hat{y}$ la clase real. En nuestra propuesta, esta definición está sujeta a w, puesto que determina la puntuación y a través de la agregación hecha por f. La inclusión del tiempo como un parámetro para la definción de la ROC fue también sugerida por Heagerty *et al*. [HLPoo] para un escenario diferente. Su propuesta consiste en una estimación para la ROC en un instante temporal posterior en un tratamiento médico, y la clase representa la supervivencia del paciente. Esto significa que sólo el cálculo de la ROC se cambia con el parámetro de la longitud de la ventana y no afecta a la etiqueta en sí. Además, la ventana de estudio es posterior al evento de interés y la clase de las instancias no vuelve a cambiar, a diferencia de nuestro caso, donde la ventana que precede el evento es una instancia positiva y la clase tras el evento es negativa. Para nuestro *framework* de evaluación, proponemos la siguiente definición de la pw-ROC:

DEFINICIÓN ROC basada en la ventana de precedencia para la ventana de longitud w:

$$\begin{aligned}\text{pw-ROC}_w = \{&(P(f(X) > c|X \in \mathcal{N}_w), P(f(X) > c|X \in \mathcal{I}_w)) : \\ &c \in (-\infty, \infty)\}\end{aligned} \tag{6.6}$$

Es importante notar la influencia de w, puesto que afecta a la definición de $\mathcal{N}_w$ y $\mathcal{I}_w$. Una vez que hemos obtenido las diferentes curvas pwROC para las longitudes de ventana deseadas, podemos generar una superficie ROC para observar las

diferencias en el balance entre precisión y especificidad con respecto a la longitud de la ventana. El estudio de este parámetro permite la caracterización del problema si los investigadores no tienen información a priori sobre la posible longitud de la ventana donde los sensores permiten detectar las anomalías. Se espera que el AUC mejore conforme aumenta la longitud de la ventana, puesto que hay menos instancias negativas. Por tanto, la longitud de la ventana cuando ocurre el incremento en la medida es otro elemento para tener en cuenta al comparar algoritmos. Si hay algoritmos que tienen un mejor AUC para ventanas más cortas, esto implica que los eventos anómalos son detectables por estos algoritmos. Otra opción es que el incremento puede producirse en el límite superior del rango de las longitudes de la ventana porque no haya instancias negativas, por lo que el conocimiento del experto es crucial para evitar esta situación.

El proceso completo está resumido en las Figuras 6.2 a la 6.6: De las puntuaciones de anomalía etiquetas proporcionadas por el algoritmo, como se muestra en la Figura 6.2, podemos filtrar y replicar el particionamiento por ventanas según los diferentes valores del parámetro de longitud de ventana, obteniendo distintas particiones de la serie temporal, ilustrado en las Figuras 6.3 a la 6.5. Entonces, la agregación se lleva a cabo según el objetivo de la experimentación y se obtiene la superficie ROC (Figura 6.6).

Las Figuras 6.3 a 6.5 también contribuyen a explicar la influencia de la longitud de la ventana en la definición de la anomalía. Consideremos, por simplicidad, que el algoritmo sólo proporciona etiquetas $\{0, 1\}$ y que la función de agregación es el máximo. Entonces, un intervalo es considerado positivo si hay una anomalía predicha en dicho intervalo. Como hemos comentado previamente, la puntuación se espera que se incremente conforme crece la longitud de la ventana, puesto que aquellos intervalos no detectados (mostrados en rojo y culminados con un punto representando un falso negativo) podrían incluir algunas instancias positivas que fuesen consideradas falsos positivos con una ventana más corta, como vemos en los dos primeros puntos marcados en la Figura 6.3 y la Figura 6.4. De manera similar, la mayoría de falsos positivos serán incluidas por intervalos verdaderos positivos, incrementando la precisión, como se ilustra en los cambios entre la Figura 6.4 y la Figura 6.5.

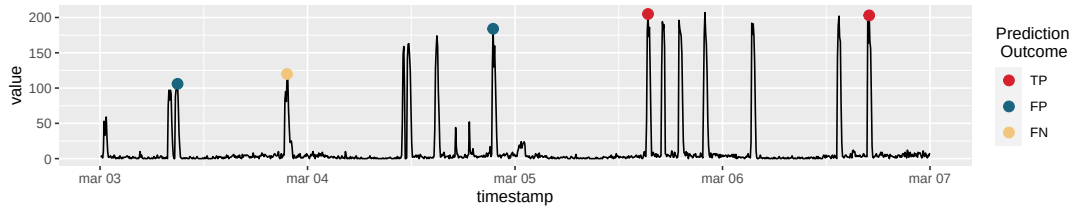


Figura 6.2: Puntuaciones de anomalía y predicción de algoritmo

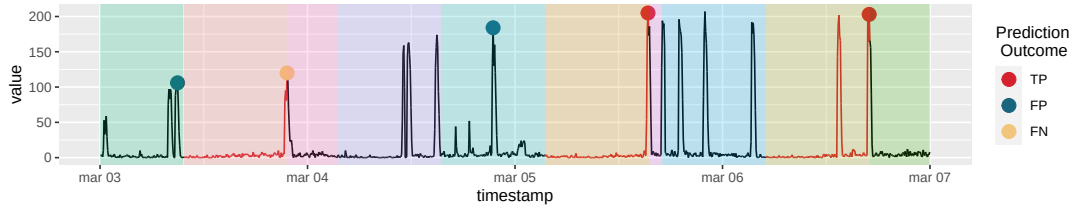


Figura 6.3: Ventana de 12 horas

6.2.4 *Software*

El paquete `pwROC` implementa el método de evaluación de algoritmos descrito. Como hemos comentado previamente, este paquete incluye dos versiones del método: una versión no distribuiida, que puede instalarse sin las dependencias de Spark y `pySpark`, y una versión distribuida (el submódulo `pwROCBD`) para series temporales en entornos *Big Data*, lo que requiere una instalación funcional de Spark y tiene el paquete `pySpark` como dependencia.

El paquete contiene la funcionalidad para preprocesar el conjunto de datos, filtrando las instancias de manera acorde a las paradas de mantenimiento, calculando una curva ROC específica para una longitud de ventana o calculando la superficie ROC para las longitudes de ventana deseadas. Las funciones de agregación disponibles son la media, mediana, el complemento de la función de distribución acumulada y el esquema de ponderación de NAB. El submódulo `pwROCBD` tiene la misma funcionalidad que la implementación clásica.

Mediante el uso de `pandas.DataFrame` y `pyspark.sql.DataFrame`, `pwROC` permite al ususario la integración del sistema de puntuación en sus análisis. La entrada esperada consiste en el `DataFrame` con el *time stamp* y la puntuación de anomalía, y un `numpy.array` con la marca de tiempo de los inicios de los eventos de interés.

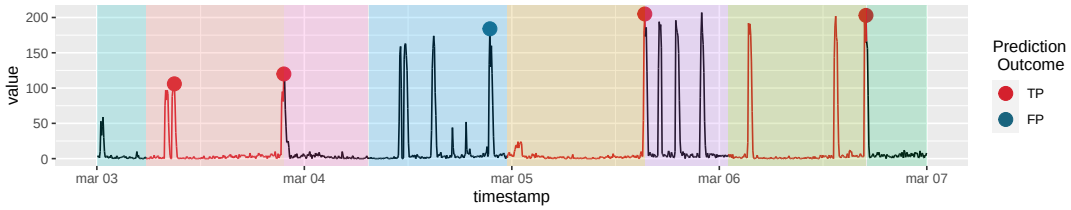


Figura 6.4: Ventana de 16 horas

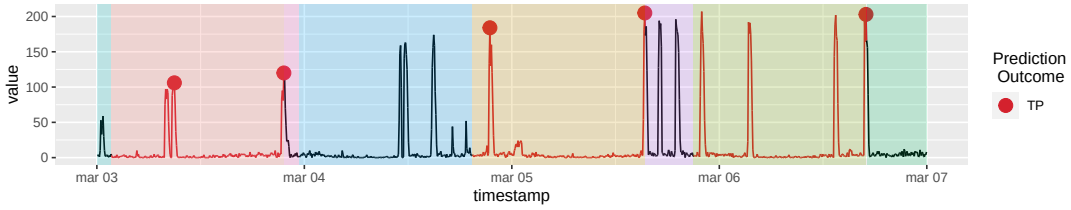


Figura 6.5: Ventana de 20 horas

6.3 CASO DE ESTUDIO

En esta sección incluimos los detalles del análisis llevado a cabo con un caso de estudio real para mostrar el correcto funcionamiento de la medida propuesta y la comparación entre diferentes medidas. Este análisis incluye una comparaación con el método de evaluación para el escenario más similar existente en la literatura.

6.3.1 *Descripción de los datos de ArcelorMittal*

Los datos utilizados han sido proporcionados por ArcelorMittal. Han sido generados por un activo que requiere atención permanente puesto que los fallos ocurren frecuentemente. Según la importancia del fallo, la máquina puede ser detenida para una reparación rápida o que se necesiten varios días. La intención es prevenir las paradas más largas mediante la pronta detección de los problemas de la máquina.

Los datos incluyen las series temporales de la sensórica y otra información relacionada (que puede indicar algún problema pero no implica un evento de interés), por ejemplo un registro de fallos o información contextual. Este conjunto de datos consta de más de 38 millones de observaciones y 112 variables, que incluyen información del contexto operacional y ambientales.

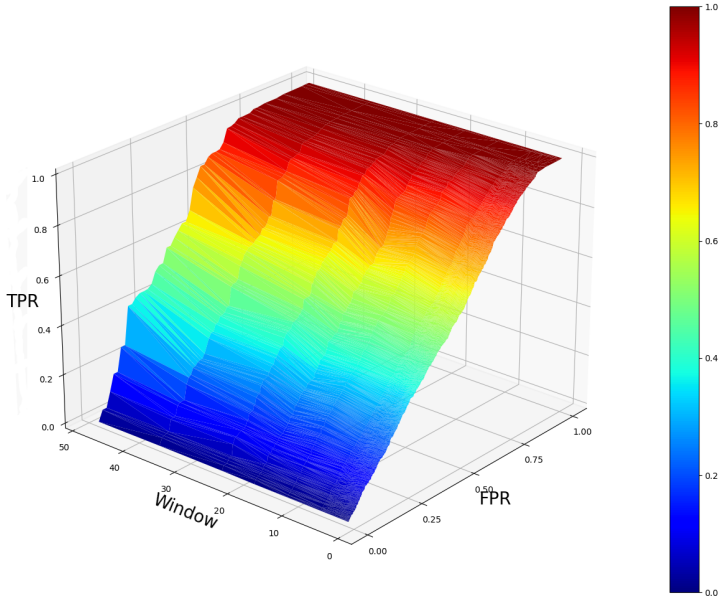


Figura 6.6: Superficie ROC

Hemos preprocesado estos datos escalándolos al rango $[0, 1]$ para evitar la influencia de la escala en el comportamiento de los algoritmos y descartado seis características del total de 112 por su valor constante.

6.3.2 *Algoritmos incluidos en la experimentación*

La experimentación incluye los resultados obtenidos por tres algoritmos de detección de anomalías no supervisados. Estos algoritmos son rediseños para *Big Data* de algoritmos clásicos de detección de anomalías y están disponibles en el paquete de Spark Package AnomalyDSD[3]. El uso de algoritmos de *Big Data* está motivado por el volumen del conjunto de datos usado.

- HBOS_BD: El algoritmo HBOS (*Histogram-based Outlier Score* [GD12] realiza un histograma para cada característica de los datos para asignar una puntuación

3 https://spark-packages.org/package/ari-dasci/S-AnomalyDSD

de anomalía según el número de instancias presentes en cada intervalo. Se proponen dos alternativas para los atributos numéricos: una estática con intervalos de igual anchura, y una dinámica, donde los valores se dividen en intervalos con un igual número de observaciones.

- LODA_BD: El algoritmo LODA (*Lightweight Online Detector of Anomalies*) es un método basado en ensamblado que usa la combinación de histogramas unidimensionales [Pev16]. La selección de variables aleatorias para realizar el histograma introduce un grado de variabilidad, una característica valiosa en métodos de ensamblado.
- XGBOD_BD: El algoritmo XGBOD (*Extreme Gradient Boosting Outlier Detection*) es una adaptación de XGBoost para escenarios semisupervisados [ZH18]. Este algoritmo usa algoritmos de detección no supervisados para obtener una representación válida para la fase supervisada del algoritmo.

En la Tabla 6.1 incluimos los parámetros por defecto para cada algoritmo incluido en la comparación. Los mejores parámetros para los algoritmos se han determinado mediante el *framework* de optimización Optuna [ASY+19]. La medida utilizada para la optimización ha sido el AUC propuesto para una ventana de seis horas.

Algoritmo	Parámetros
HBOS_BD	n_bins = 100, strategy = “static"
LODA_BD	n_bins = 100, k = 100
XGBOD_BD	detector = “LODA_BD", n_TOS = 10, n_selected_TOS = 5, TOS_strategy = “acc", threshold = 0.1

Tabla 6.1: Parámetros por defecto para los detectores de anomalías

Estos algoritmos devuelven una puntuación de anomalía para cada instancia, así que no se han utilizado las funciones descritas en Ecuación 6.2.2, aunque se incluyen en este capítulo para proporcionar los mecanismos para adaptar el sistema de puntuación a los intereses del investigador.

6.3.3 *Resultados y análisis*

En la Tabla 6.2 incluimos el valor AUC de los algoritmos para varios valores de la ventana usando las diferentes funciones de agregación. El mejor resultado de AUC está resaltado en negrita. Para este método de evaluación las variantes de HBOS son en general las más efectivas, indicando la anormalidad de los intervalos de tiempo previos a las alarmas. El valor creciente de AUC con respecto al período considerado es general para todas las agregaciones y esquemas de ponderación. Como se ha descrito en la Sección 6.2, este es el comportamiento esperado puesto que hay menos ventanas a ser consideradas y aquellas previas a un incedente se considerarán anómalas. Sin embargo, este comportamiento no está garantizado para algoritmos con una salida aleatoria, y dichos valores aleatorios de AUC se obtienen en esta experimentación para ciertos algoritmos y ventanas pequeñas. En este caso de estudio particular, valores pequeños de AUC no deben atribuirse a un rendimiento aleatorio sino a la relación entre el comportamiento anómalo de la máquina estudiada y la marca de tiempo donde el evento de interés comienza.

Horas previas	Función de agregación	HBOS		LODA	XGBOD
		Estática	Dinámica		
1	Media	0.5623	**0.5760**	0.5592	0.4963
	ccdf	0.4206	0.4410	0.4252	**0.4510**
	NAB	**0.5159**	0.5155	0.5155	0.4314
6	Media	0.5692	**0.5886**	0.5584	0.4924
	ccdf	0.4619	0.4734	0.4827	**0.5092**
	NAB	0.6569	**0.6571**	0.6557	0.6258
48	Media	**0.7577**	0.7334	0.6781	0.5651
	ccdf	0.5792	0.5620	**0.6418**	0.6094
	NAB	0.8373	0.8322	0.8332	**0.8537**

Tabla 6.2: Valor AUC para cada algoritmo y ventana para diferentes agregaciones.

Con respecto a la agregación mediante la CCDF, los valores muestran cómo una agregación no lineal beneficia el algoritmo XGBOD. Por tanto, dichos algoritmos

son más apropiados para escenarios donde puntos individuales puedan indicar un comportamiento anómalo en lugar de aquellos donde las anomalías provengan de una degradación del sistema reflejada en la serie temporal.

Los valores usando el sistema de ponderación del *benchmark* de Numenta son similares a los obtenidos usando la media, algo esperable teniendo en cuenta que es también una combinación lineal de las puntuaciones. Sin embargo, conforme se incrementa la longitud de las ventanas los valores más alejados a los eventos tienen un mayor peso relativo y vemos cómo los algoritmos no marcan como anómalos los puntos cercanos al mismo. Es importante notar que XGBOD obtiene el mejor resultado para la ventana de 48 horas con este esquema de evaluación, por lo que este algoritmo también detecta un comportamiento anómalo previo al evento, aunque sólo al comienzo de la ventana, el cual es el comportamiento recompensado por el sistema de puntuación del NAB.

En la Figura 6.7 se muestra la curva ROC para el algoritmo LODA con la media como agregación para el intervalo de 36 horas. El rendimiento es sólo ligeramente mejor que una predicción aleatoria, aunque para esta longitud de ventana podría ser útil en la detección de anomalías. En la Figura 6.8 mostramos la superficie ROC del algoritmo LODA para ventanas de hasta 48 horas previas a un incidente también con la media como función de agregación.

En esta imagen la superficie ROC muestra que el rendimiento es cercano a aleatorio para ventanas menores de 20 horas. Podemos observar un incremento en el rendimiento cuando la longitud de la ventana es cercana al límite superior, particularmente en la ventana más larga, donde hay un incremento en la tasa de VP mientras la de FP permanece baja.

Hemos realizado un análisis estadístico con los resultados de las AUCs para la agregación media para comparar los algoritmos. En este capítulo no se presenta un algoritmo de detección sino un *framework* de evaluación, por lo que el análisis no está centrado en uno de los algoritmos. Para mostrar que hay diferencias entre los resultados hemos usado el test de Friedman, puesto que los resultados no se espera que sigan una distribución normal por la diferencia entre la longitud de las ventanas [CGR+20]. El p-valor obtenido es $1.41 \cdot 10^{-6}$, por lo que rechazamos la hipótesis nula que representa la equivalencia de los métodos.

Los resultados del test de Friedman con el ajuste post-hoc de Holland, usado para mostrar las diferencias identificadas por el test de Friedman se muestran gráficamente en la Figura 6.9. Aquí mostramos que aunque las variantes del algoritmo HBOS son los algoritmos con mejores resultados no podemos descartar la posibilidad de que

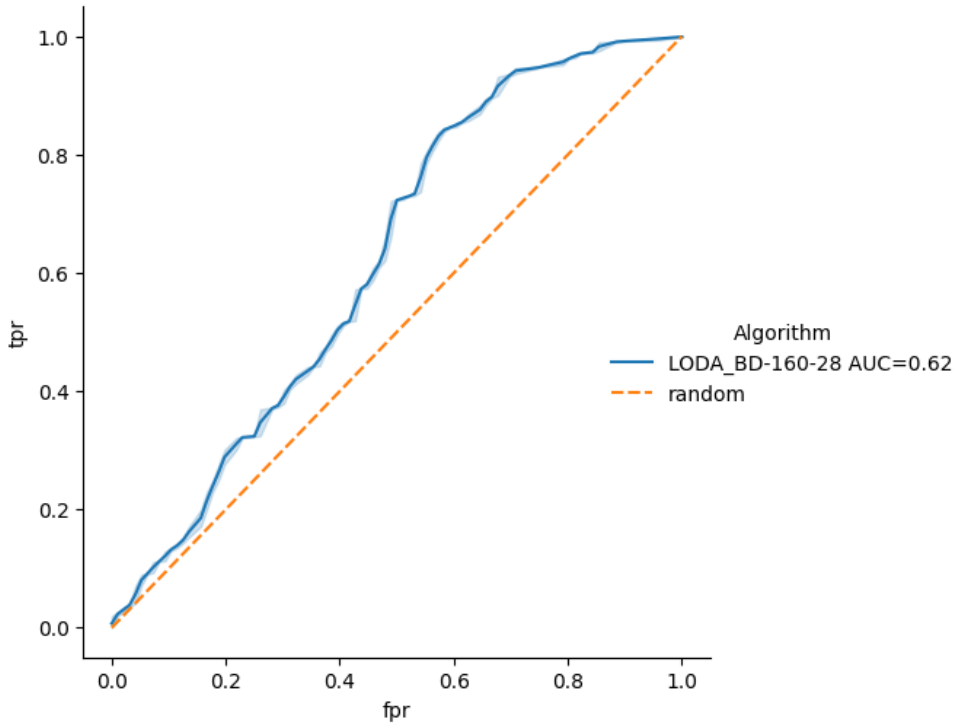


Figura 6.7: Ejemplo de curva ROC para ventana de 36 horas.

estas diferencias se produzcan al azar. La principal equivalencia entre los algoritmos que puede ser descartada es entre la versión dinámica de HBOS y los algoritmos LODA y XGBOD y entre XGBOD y las variantes de HBOS.

6.3.4 *Comparación con la exhaustividad y precisión basadas en intervalos*

En esta sección comparamos nuestro método con la propuesta de puntuación basada en intervalos [TLZ+19]. El objetivo de la comparación es mostrar que el concepto de la ROC basada en la ventana previa incluye la información calculada por la puntuación basada en intervalos, con el beneficio adicional de la curva ROC para escenarios no supervisados. Además, incluimos ciertas consideraciones:

- La propuesta basada en intervalos está orientada a la evaluación de la detección de eventos anómalos, por lo que hemos adaptado el conjunto de datos, etique-

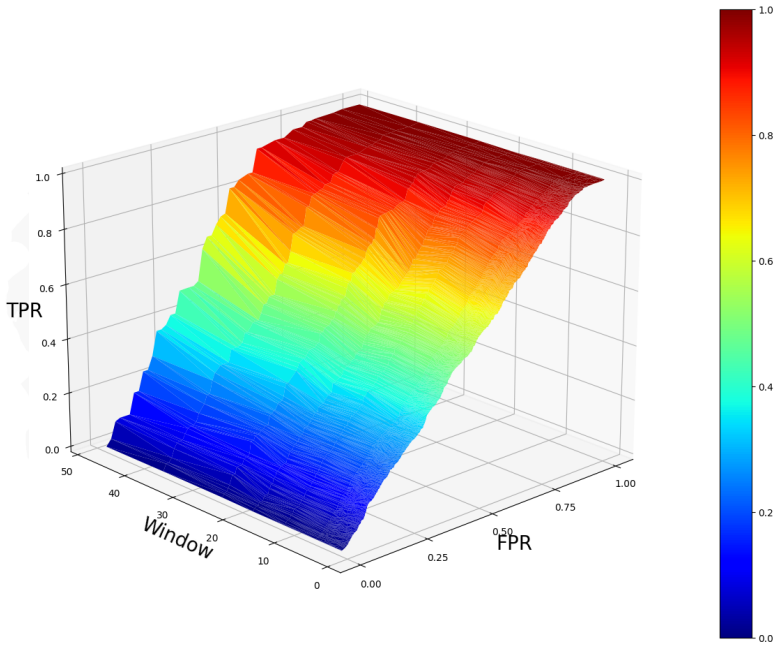


Figura 6.8: Superficie ROC para algoritmo LODA

tando como anómalas las instancias previas a los eventos usando diferentes longitudes de ventana, de manera similar al método propuesto.

- Este método asume que la salida del algoritmo son etiquetas binarias, mientras nuestro método trabaja con la puntuación de anomalía. Entonces, para los algoritmos incluidos en la comparación, se han usado ciertos niveles de anomalía para transformar la puntuación de anomalía en una etiqueta binaria. Para realizar una comparación más directa hemos calculado la precisión, exhaustividad y puntuación F_1 a los umbrales equivalentes para la curva ROC. Es importante notar que los umbrales en la evaluación mediante pw-ROC se refiere a la puntuación agregada, aunque no hay una relación unívoca entre instancias.

- La comparación se ha realizado usando un subconjunto por el coste computacional de las medidas basadas en intervalos.

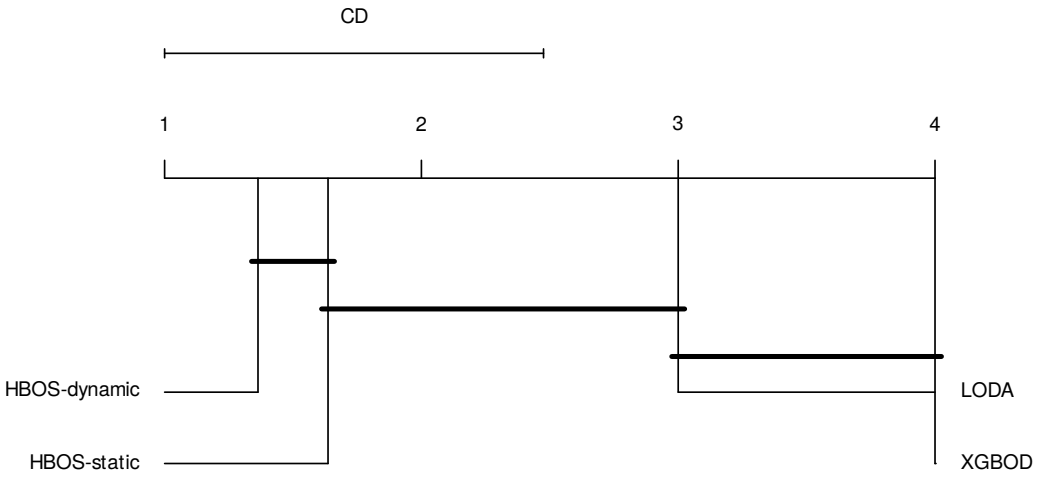


Figura 6.9: Diagrama de *Critical Difference*

- Se ha usado la media como función de agregación para las medidas pw-ROC.
- La ponderación basada en tiempo descrita en la propuesta basada en intervalos tiene un objetivo similar a los pesos del NAB. Aquí se ha usado una ponderación plana en ambos sistemas para centrar la comparación en la evaluación de la detección de anomalía.

Para cada algoritmo, sean $q_{0.05}$ y $q_{0.95}$ los cuantiles 0.05 y 0.95 de las puntuaciones, respectivamente. Entonces, las etiquetas han sido asignando una transformación lineal para los umbrales:

$$\text{Label}(x, \alpha) = \begin{cases} 1 & \text{if } x \geq q_{0.05} + \alpha(q_{0.95} - q_{0.05}) \\ 0 & \text{otherwise} \end{cases}, \tag{6.7}$$

donde α toma los valores $0.2, 0.5$ y 0.8. Hemos seleccionado los cuantiles 0.05 y 0.95 para filtrar las puntuaciones más extremas. Entonces, hemos usado los tres umbrales para imitar un escenario donde sólo se dispone de una estimación de la distribución de las puntuaciones. En la Tabla 6.3 mostramos la puntuación F_1 basado en intervalos usando diferentes longitudes de ventana y umbrales. Con la excepción de la combinación de la ventana más corta y el umbral más bajo, donde las versiones estática y dinámica para las ventanas de una y seis horas, la mejor puntuación F_1 se

Previous hours	Threshold	HBOS		LODA	XGBOD
		Static	Dynamic		
1	0.2	0.121	**0.129**	0.116	0.122
	0.5	0.090	0.114	0.103	**0.122**
	0.8	0.089	0.071	0.075	**0.122**
6	0.2	**0.516**	0.473	0.415	0.490
	0.5	0.339	0.407	0.332	**0.490**
	0.8	0.188	0.168	0.175	**0.490**
48	0.2	0.795	0.915	0.857	**0.998**
	0.5	0.618	0.691	0.512	**0.998**
	0.8	0.245	0.274	0.289	**0.998**

Tabla 6.3: Medida F_1 basada en intervalos

obtiene para otro escenario por el algoritmo XGBOD. Las grandes diferencias entre las puntuaciones con los diferentes umbrales de anomalías implican que este sistema de puntuación recompensa muy positivamente la detección de instancias anómalas sin penalizar los falsos positivos, posiblemente debido al bajo umbral. Las diferencias con respecto a la evaluación mediante pw-ROC, donde XGBOD obtuvo los peores resultados, puede explicarse por el uso de un umbral, lo cual es determinante para la detección de anomalías. Esta circunstancia viene explicada por las Figuras 6.10 a 6.12, las cuales muestran la comparación para la precisión, exhaustividad y medida F_1 entre la medida basada en intervalos y en la basada en la ventana de precedencia para cada umbral y longitud de ventana para ambos métodos de evaluación (representado en colores).

La Figura 6.10 muestra la precisión de los algoritmos para los distintos umbrales y longitud de ventanas. Los resultados para ambos métodos de puntuación son más similares para la medida de precisión. Para la ventana de 48 horas todos los intervalos anómalos predichos caen en intervalos anómalos, suponiendo una precisión basada en intervalos de 1. Sin embargo, este no es el caso para el cálculo usando la precisión mediante pw-ROC puesto que algunas puntuaciones medias de intervalos positivos son inferiores a la media de instancias fuera de estas ventanas. Por tanto, la situación

aquí podría resultar de que intervalos positivos y ventanas contienen instancias anómalas, aunque la mayoría de puntuaciones sean muy inferiores.

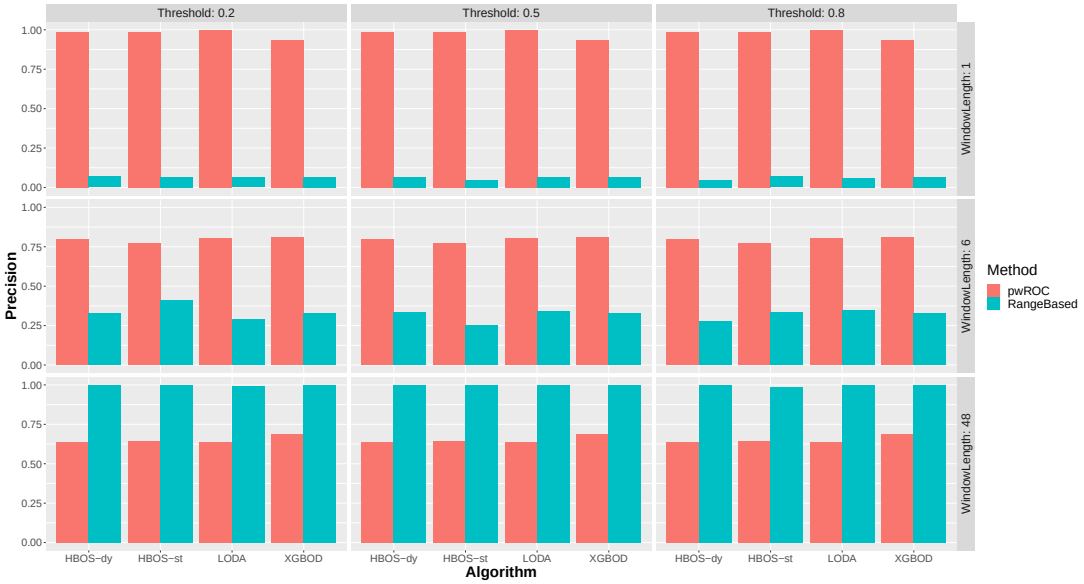


Figura 6.10: Precisión basada en ventanas precedentes frente a basada en intervalos

La mencionada hipótesis se ve apoyada por la comparación entre la exhaustividad basada en intervalos y la basada en la ventana de precedencia mostrada en la Figura 6.11. Es importante notar que la exhaustividad basada en intervalos del algoritmo XGBOD es 1 en todos los escenarios, lo que indica que todas las instancias positivas son etiquetadas como tal para todos los umbrales. Por tanto, la determinación del umbral de anomalía es crucial para la evaluación de este algoritmo, mientras que el método basado en pw-ROC proporciona una evaluación más parecida a los otros algoritmos. Es decir, para este enfoque deberíamos conocer la distribución de las puntuaciones y ajustar el umbral según esta distribución, puesto que la precisión es muy baja para las ventanas más cortas. Estos resultados refuerzan la hipótesis de que los métodos de evaluación basada en ROC son más justos, ya que muestran la efectividad para todos los umbrales. El hecho de que la presencia o ausencia de instancias etiquetadas como anomalías afecte a la evaluacción del intervalo completo amplifica la polarización de la salida de la evaluación basada en intervalos, mientras

que la pw-ROC agrega la información de las instancias en los periodos, un proceso que evita la dicotomización de la salida.

Figura 6.11: Exhaustividad basada en ventanas precedentes frente a basada en intervalos

La combinación de las puntuaciones previas se muestra en la Figura 6.12, que ilusta la comparación entre la F_1 basada en intervalos para cada umbral y longitud de ventana. La diferencia más profunda recae de nuevo en el algoritmo XGBOD que tiene un buen resultado en término de pw-ROC para los umbrales bajos pero obtiene una F_1 basada en intervalos cercano a 1 para cada umbral para la ventana de 48 horas.

Como hemos mencionado previamente, este es un caso donde la buena definición del umbral es crucial, mientras que el cálculo de la puntuación basada en pw-ROC no se ve afectada por ello, no solo innecesaria para la definición de la curva sino también para la agregación de la ventana.

ANÁLISIS DE COSTE El supuesto coste computacional de la medida basada en intervalos es $O(N_r \times N_p)$, donde N_r es el número de intervalos positivos y N_p es el número de intervalos positivos predichos. Sin embargo, este coste omite el cómputo del tamaño de la intersección de los intervalos reales y predichos y la ponderación

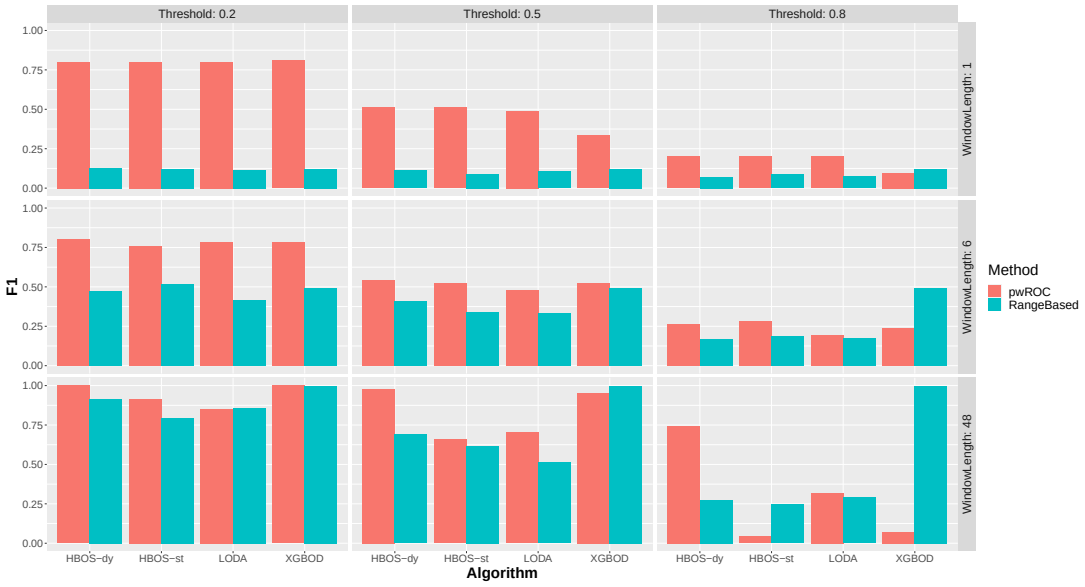


Figura 6.12: Medida F_1 basada en ventanas precedentes frente a basada en intervalos

de la instancia en ella, que podría llevar a un coste $O(n^2)$, donde n es el número de instancias. La complejidad de nuestra propuesta es $O(n)$, que podría ser un número mucho mayor que $N_r \cdot N_p$, aunque es mucho más eficiente en la práctica.

La Tabla 6.4 muestra el coste computacional de cada método de evaluación para cada umbral y longitud de ventana. Los resultados incluidos en esta tabla son la media del tiempo de ejecución de todos los algoritmos medido en segundos. El incremento en el coste computacional de la medida basada en intervalos para ventanas más largas es causado por la necesidad de calcular la intersección entre los intervalos anómalos más largos. Una circunstancia similar ocurre con el umbral de 0.5, que implica un mayor número de cambios entre predicciones normales y anómalos y por tanto un mayor número de intervalos anómalos predichos. Por contra, el coste del método propuesto se ve sólo afectado por el número de instancias, y el uso de ventanas más largas significa menos agregaciones de más instancias.

Longitud de ventana	Umbral	pwROC	Basado en intervalos
1	0.2	0.14	146.12
	0.5	0.11	214.73
	0.8	0.09	101.24
6	0.2	0.46	233.41
	0.5	0.41	360.14
	0.8	0.29	167.15
48	0.2	0.94	17533.13
	0.5	0.88	29105.63
	0.8	0.38	12166.45

Tabla 6.4: Comparación de coste computacional (s) de los métodos de evaluación

6.4 CONCLUSIONES FINALES

En este capítulo se propone un *framework* de evaluación para detección de anomalías para escenarios de series temporales. Este método de evaluación adapta el uso del AUC, una medida extendida en la evaluación de algoritmos de detección de anomalías y de clasificación desbalanceada, para escenarios de detección de eventos contemplando la falta de certeza sobre el efecto del evento en las instancias previas. La descripción de los componentes aplicados a la serie temporal original permite la propuesta de nuevas funciones de agregación diseñadas para cada caso particular dentro del *framework* propuesto, facilitando conclusiones más ajustadas al caso de estudio. También hemos adaptado el sistema de puntuación del *benchmark* de Numenta para detección de anomalías temprana, aunque nuestra definición evita la proliferación de parámetros que dificultan el entendimiento de la medida.

El caso de estudio incluido con la experimentación de tres algoritmos distribuidos sobre un conjunto de datos real con las propiedades mencionadas sobre la anotación de las anomalías, y la comparación con un sistema de evaluación basados en intervalos, han validado la robustez de la propuesta de las métricas a través de la descripción de una medida basada en la ROC en lugar de una dependiente del umbral de anomalías, la cual proporciona una situación más dicotómica. Por tanto, la medida de evaluación hace viable el estudio de un problema de detección de

anomalía, gracias a sus implementaciones tanto para un escenario clásico como para escenarios de *Big Data*, que de otra manera tendrían que usar medidas inapropiadas o de un cálculo poco eficiente para valorar la calidad de los algoritmos.

7

CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

En esta tesis hemos abordado varios problemas dentro del marco de la detección de anomalías en entornos no supervisados. El primer objetivo ha sido la la definición del problema abordado y la especificación de la información disponible tanto para la ejecución del entrenamiento como en el momento de la evaluación.

Para la propuesta algorítmica se ha partido de un modelo clásico de detección de anomalías como es HBOS para la búsqueda de elementos anómalos en subespacios de una mayor dimensionalidad. La elección de este algoritmo viene justificada por la inclusión de un mecanismo de explicabilidad que hace uso de la representación generada por los histogramas para la formación de reglas representativas que permitan justificar la determinación de la clase de una instancia y proporcionar ejemplos de modificaciones que puedan cambiar la instancia de clase. Además, la implementación de este modelo se ha realizado para entornos distribuidos para poder abordar problemas en el ámbito *Big Data*. La experimentación llevada a cabo, tanto en escenarios de datos estáticos como de series temporales, ha mostrado que la simplicidad del modelo que permite fácilmente la reconstrucción de las proyecciones en varias dimensiones y de la obtención de reglas que explican la anomalía va en detrimento de la precisión del modelo. Sin embargo, en función de la información disponible sobre el conjunto de datos puede ser relevante la obtención de explicaciones sobre elementos clave en contraprestación a cierta fiabilidad en los resultados.

El último aspecto abordado en esta tesis ha sido el de la evaluación de algoritmos de detección de anomalías en entornos no supervisados para series temporales cuando se dispone de información puntual sobre ciertos eventos de interés. Este problema es habitual en la investigación sobre mantenimiento predictivo, donde se pretende minimizar el riesgo de componentes de sistemas industriales mediante la identificación de posibles fallos a lo largo de una serie temporal, aunque se usan algoritmos de detección de anomalías que proporcionan esta información punto a punto. Esta disociación entre los sistemas de evaluación comunes en la literatura y la realidad temporal ha motivado la propuesta de un *framework* de

evaluación para algoritmos de detección de anomalías en series temporales, donde no hay instancias marcadas como anómalas, sino eventos que queremos predecir en base al grado de anomalía que indican los algoritmos en períodos previos. Por la naturaleza de los datos, proporcionamos además de una implementación clásica, una implementación distribuida del método de evaluación y realizamos una comparación con una propuesta de evaluación basada en intervalos anómalos para mostrar la utilidad de la propuesta y la mejora en eficiencia de su cómputo con respecto a la evaluación basada en intervalos.

7.2 PUBLICACIÓN ASOCIADA A LA TESIS

- Carrasco, J., López, David, Aguilera-Martos, I., García-Gil, D., Markova, I., García-Barzana, M., Arias-Rodil, M., Luengo, J., & Herrera, F. (2021). Anomaly detection in predictive maintenance: A new evaluation framework for temporal unsupervised anomaly detection algorithms. *Neurocomputing*, 462(), 440–452. DOI: http://dx.doi.org/10.1016/j.neucom.2021.07.095
  - Estado: Publicado
  - Factor de impacto (JCR 2021): 5.719
  - Categoría: Artificial Intelligence (Q1)
  - Categoría: Cognitive Neuroscience (Q1)
  - Categoría: Computer Science Applications (Q1)

7.3 TRABAJOS FUTUROS

Como hemos visto, el diseño del modelo basado en la combinación de modelos explicables y cuyo cómputo se puede reutilizar en subespacios de dimensión mayor, como son los histogramas no ha proporcionado los resultados esperados en términos de precisión. Por tanto, la labor principal será, partiendo de este modelo explicable, realizar una búsqueda de métodos de agregación que permita la identificación de anomalías en subespacios.

Actualmente se está trabajando bajo el paraguas de un proyecto de detección de anomalías con datos reales en un escenario de mantenimiento predictivo, por lo que continuaremos el trabajo de detección de anomalías para entornos *Big Data* con el

objetivo adicional de la explicabilidad, un aspecto fundamental para la extracción de conocimiento en problemas no supervisados con grandes volúmenes de datos.

BIBLIOGRAFÍA

[Agg17] C. C. Aggarwal, *Outlier Analysis*. New York: Springer-Verlag, 2017.

[AGL+22] I. Aguilera-Martos, Á. M. García-Vico, J. Luengo, S. Damas, F. J. Melero, J. J. Valle-Alonso y F. Herrera, *TSFEDL: A Python Library for Time Series Spatio-Temporal Feature Extraction and Prediction Using Deep Learning (with Appendices on Detailed Network Architectures and Experimental Cases of Study)*, jun. de 2022. arXiv: 2206.03179 [cs].

[AAV+22] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja y U. Genc, *Anomalib: A Deep Learning Library for Anomaly Detection*, feb. de 2022. arXiv: 2202.08341 [cs].

[AAB18] S. Akcay, A. Atapour-Abarghouei y T. P. Breckon, «Ganomaly: Semi-supervised Anomaly Detection via Adversarial Training», en *Asian Conference on Computer Vision*, Springer, 2018, págs. 622-637.

[ASY+19] T. Akiba, S. Sano, T. Yanase, T. Ohta y M. Koyama, «Optuna: A Next-generation Hyperparameter Optimization Framework», en *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ép. KDD '19, New York, NY, USA: Association for Computing Machinery, jul. de 2019, págs. 2623-2631.

[ABC20] Y. Almardeny, N. Boujnah y F. Cleary, «A Novel Outlier Detection Method for Multivariate Data», *IEEE Transactions on Knowledge and Data Engineering*, págs. 1-1, 2020.

[AFP13] F. Angiulli, F. Fassetti y L. Palopoli, «Discovering Characterizations of the Behavior of Anomalous Subpopulations», *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, n.º 6, págs. 1280-1292, jun. de 2013.

[AP02] F. Angiulli y C. Pizzuti, «Fast Outlier Detection in High Dimensional Spaces», en *Principles of Data Mining and Knowledge Discovery*, T. Elomaa, H. Mannila y H. Toivonen, eds., ép. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2002, págs. 15-27.

[BBM+15] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller y W. Samek, «On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation», *PLOS ONE*, vol. 10, n.º 7, e0130140, jul. de 2015.

[BDD+20] A. Barredo Arrieta y col., «Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI», *Information Fusion*, vol. 58, págs. 82-115, jun. de 2020.

[BKL+21] A. Bhatnagar y col., «Merlion: A Machine Learning Library for Time Series», *arXiv:2109.09265 [cs, stat]*, sep. de 2021. arXiv: 2109.09265 [cs, stat].

[BGM+01] A. M. Bianco, M. Garcia Ben, E. Martinez y V. J. Yohai, «Outlier Detection in Regression Models with Arima Errors Using Robust Estimates», *Journal of Forecasting*, vol. 20, n.º 8, págs. 565-579, 2001.

[BKN+00] M. M. Breunig, H.-P. Kriegel, R. T. Ng y J. Sander, «LOF: Identifying Density-Based Local Outliers», en *ACM Sigmod Record*, vol. 29, ACM, 2000, págs. 93-104.

[BZA17] L. Bu, D. Zhao y C. Alippi, «An Incremental Change Detection Test Based on Density Difference Estimation», *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, n.º 10, págs. 2714-2726, oct. de 2017.

[CGR+20] J. Carrasco, S. García, M. M. Rueda, S. Das y F. Herrera, «Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review», *Swarm and Evolutionary Computation*, vol. 54, pág. 100 665, mayo de 2020.

[CIL19] A. Carreño, I. Inza y J. A. Lozano, «Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework», *Artificial Intelligence Review*, oct. de 2019.

[CS12] K. M. Carter y W. W. Streilein, «Probabilistic Reasoning for Streaming Anomaly Detection», *2012 IEEE Statistical Signal Processing Workshop (SSP)*, ago. de 2012.

[CBK09] V. Chandola, A. Banerjee y V. Kumar, «Anomaly Detection: A Survey», *ACM computing surveys (CSUR)*, vol. 41, n.º 3, pág. 15, 2009.

[CHZ+20] C. Chen, Z. Hua, R. Zhang, G. Liu y W. Wen, «Automated arrhythmia classification based on a combination network of CNN and LSTM», *Biomedical Signal Processing and Control*, vol. 57, pág. 101 819, mar. de 2020.

[Cra09] N. Craswell, «Precision at n.», en *Encyclopedia of Database Systems*, Berlin, Germany: Springer, 2009, págs. 2127-2128.

[DAN+14] X. H. Dang, I. Assent, R. T. Ng, A. Zimek y E. Schubert, «Discriminative Features for Identifying and Interpreting Outliers», en *2014 IEEE 30th International Conference on Data Engineering*, mar. de 2014, págs. 88-99.

[DG04] J. Dean y S. Ghemawat, «MapReduce: Simplified Data Processing on Large Clusters», 2004.

[DSL+21] T. Defard, A. Setkov, A. Loesch y R. Audigier, «PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization», en *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante y R. Vezzani, eds., ép. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, págs. 475-489.

[FGG+18] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk y F. Herrera, *Learning from Imbalanced Data Sets*. Springer, 2018.

[FFA20] D. Fernández-Francos, Ó. Fontenla-Romero y A. Alonso-Betanzos, «One-Class Convex Hull-Based Algorithm for Classification in Distributed Environments», *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, n.º 2, págs. 386-396, feb. de 2020.

[GT06] J. Gao y P.-N. Tan, «Converting Output Scores from Outlier Detection Algorithms into Probability Estimates», en *Sixth International Conference on Data Mining (ICDM'06)*, IEEE, 2006, págs. 212-221.

[GWL21] Y. Gao, H. Wang y Z. Liu, «An end-to-end atrial fibrillation detection by a novel residual-based temporal attention convolutional neural network with exponential nonlinearity loss», *Knowledge-Based Systems*, vol. 212, pág. 106 589, ene. de 2021.

[GD12] M. Goldstein y A. Dengel, «Histogram-Based Outlier Score (Hbos): A Fast Unsupervised Anomaly Detection Algorithm», *KI-2012: Poster and Demo Track*, págs. 59-63, 2012.

[GU16] M. Goldstein y S. Uchida, «A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data», *PLOS ONE*, vol. 11, n.º 4, e0152173, abr. de 2016.

[GIK22] D. Gudovskiy, S. Ishizaka y K. Kozuka, «CFLOW-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows», en *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, págs. 98-107.

[GMR+16] S. Guha, N. Mishra, G. Roy y O. Schrijvers, «Robust Random Cut Forest Based Anomaly Detection on Streams», en *International Conference on Machine Learning*, 2016, págs. 2712-2721.

[GMR+18] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini y F. Giannotti, «Local Rule-Based Explanations of Black Box Decision Systems», *arXiv:1805.10820 [cs]*, mayo de 2018. arXiv: 1805.10820 [cs].

[GSY+19] B. Gunay, Z. Shi, C. Yang, W. Shen y D. Darwazeh, «An Inquiry into the Predictability of Failure Events in Chillers and Boilers», en *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, ago. de 2019, págs. 222-227.

[GES+19] N. Gupta, D. Eswaran, N. Shah, L. Akoglu y C. Faloutsos, «Beyond Outlier Detection: LookOut for Pictorial Explanation», en *Machine Learning and Knowledge Discovery in Databases*, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley y G. Ifrim, eds., ép. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, págs. 122-138.

[HPK11] J. Han, J. Pei y M. Kamber, *Data Mining: Concepts and Techniques*. Elsevier, 2011.

[HHH+22] S. Han, X. Hu, H. Huang, M. Jiang e Y. Zhao, *ADBench: Anomaly Detection Benchmark*, sep. de 2022. arXiv: 2206.09426 [cs].

[HM82] J. A. Hanley y B. J. McNeil, «The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve.», *Radiology*, vol. 143, n.º 1, págs. 29-36, abr. de 1982.

[Haw80] D. M. Hawkins, *Identification of Outliers*. Springer, 1980, vol. 11.

[HXD03] Z. He, X. Xu y S. Deng, «Discovering Cluster-Based Local Outliers», *Pattern Recognition Letters*, vol. 24, n.º 9, págs. 1641-1650, 2003.

[HLP00] P. J. Heagerty, T. Lumley y M. S. Pepe, «Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker», *Biometrics*, vol. 56, n.º 2, págs. 337-344, 2000.

[HAK00] A. Hinneburg, C. C. Aggarwal y D. A. Keim, «What is the nearest neighbor in high dimensional spaces?», en *26th Internat. Conference on Very Large Databases*, 2000, págs. 506-515.

[Ho95] T. K. Ho, «Random Decision Forests», en *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, ago. de 1995, 278-282 vol.1.

[HW20] M.-L. Huang e Y.-S. Wu, «Classification of atrial fibrillation and normal sinus rhythm based on convolutional neural network», *Biomedical Engineering Letters*, ene. de 2020.

[INB+17] V. Ishimtsev, I. Nazarov, A. Bernstein y E. Burnaev, «Conformal K-NN Anomaly Detector for Univariate Data Streams», *arXiv preprint arXiv:1706.03412*, 2017. arXiv: 1706.03412.

[ICC+20] A. Iturria, J. Carrasco, S. Charramendieta, A. Conde y F. Herrera, «Otsad: A package for online time-series anomaly detectors», *Neurocomputing*, vol. 374, págs. 49-53, ene. de 2020.

[KMF+17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye y T.-Y. Liu, «LightGBM: A Highly Efficient Gradient Boosting Decision Tree», en *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[KW14] D. P. Kingma y M. Welling, *Auto-Encoding Variational Bayes*, mayo de 2014. arXiv: 1312.6114 [cs, stat].

[KN99] E. M. Knorr y R. T. Ng, «Finding Intensional Knowledge of Distance-Based Outliers», en *Proceedings of the 25th International Conference on Very Large Data Bases*, ép. VLDB '99, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, págs. 211-222.

[KZ+08] H.-P. Kriegel, A. Zimek y col., «Angle-Based Outlier Detection in High-Dimensional Data», en *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2008, págs. 444-452.

[LA15] A. Lavin y S. Ahmad, «Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark», *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, dic. de 2015.

[LGT+18] T. J. Lee, J. Gottschlich, N. Tatbul, E. Metcalf y S. Zdonik, «Precision and Recall for Range-Based Anomaly Detection», *arXiv:1801.03175 [cs]*, feb. de 2018. arXiv: 1801.03175 [cs].

[LZB+20] Z. Li, Y. Zhao, N. Botta, C. Ionescu y X. Hu, «COPOD: Copula-Based Outlier Detection», en *2020 IEEE International Conference on Data Mining (ICDM)*, nov. de 2020, págs. 1118-1123.

[LZH+22] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu y G. H. Chen, «ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions», *arXiv:2201.00382 [cs, stat]*, ene. de 2022. arXiv: 2201.00382 [cs, stat].

[LTZ08] F. T. Liu, K. M. Ting y Z.-H. Zhou, «Isolation Forest», en *2008 Eighth IEEE International Conference on Data Mining*, IEEE, 2008, págs. 413-422.

[LTZ12] F. T. Liu, K. M. Ting y Z.-H. Zhou, «Isolation-Based Anomaly Detection», *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, n.º 1, pág. 3, 2012.

[LGR+20] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García y F. Herrera, *Big Data Preprocessing: Enabling Smart Data*. Cham: Springer International Publishing, 2020.

[LL17] S. M. Lundberg y S.-I. Lee, «A Unified Approach to Interpreting Model Predictions», en *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan y R. Garnett, eds., Curran Associates, Inc., 2017, págs. 4765-4774.

[Lüt05] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005.

[MSA18] S. Makridakis, E. Spiliotis y V. Assimakopoulos, «The M4 Competition: Results, findings, conclusion and way forward», *International Journal of Forecasting*, vol. 34, n.º 4, págs. 802-808, oct. de 2018.

[MMA14] B. Micenková, B. McWilliams e I. Assent, «Learning Outlier Ensembles: The Best of Both Worlds–Supervised and Unsupervised», en *Proceedings of the ACM SIGKDD 2014 Workshop on Outlier Detection and Description under Data Diversity (ODD2). New York, NY, USA*, 2014, págs. 51-54.

[MSS11] E. Müller, M. Schiffer y T. Seidl, «Statistical Selection of Relevant Subspace Projections for Outlier Ranking», en *Data Engineering (ICDE), 2011 IEEE 27th International Conference On*, IEEE, 2011, págs. 434-445.

[NHH18] R. Noorossana, S. S. Hosseini y A. Heydarzade, «An overview of dynamic anomaly detection in social networks via control charts», *Quality and Reliability Engineering International*, vol. 34, n.º 4, págs. 641-648, 2018.

[ONT+18] S. L. Oh, E. Y. K. Ng, R. S. Tan y U. R. Acharya, «Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats», *Computers in Biology and Medicine*, vol. 102, págs. 278-287, nov. de 2018.

[OCF08] A. L. I. Oliveira, F. R. G. Costa y C. O. S. Filho, «Novelty detection with constructive probabilistic neural networks», *Neurocomputing*, Neural Networks: Algorithms and Applications, vol. 71, n.º 4, págs. 1046-1053, ene. de 2008.

[Pev16] T. Pevnỳ, «Loda: Lightweight on-Line Detector of Anomalies», *Machine Learning*, vol. 102, n.º 2, págs. 275-304, 2016.

[RRS00] S. Ramaswamy, R. Rastogi y K. Shim, «Efficient Algorithms for Mining Outliers from Large Data Sets», en *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ép. SIGMOD '00, New York, NY, USA: ACM, 2000, págs. 427-438.

[RYS+21] A. Ravi, X. Yu, I. Santelices, F. Karray y B. Fidan, «General Frameworks for Anomaly Detection Explainability: Comparative Study», en *2021 IEEE International Conference on Autonomous Systems (ICAS)*, ago. de 2021, págs. 1-5.

[Ray16] S. Rayana, «ODDS Library», *Stony Brook University, Department of Computer Sciences*, 2016.

[RPL15] H. Raza, G. Prasad e Y. Li, «EWMA Model Based Shift-Detection Methods for Detecting Covariate Shifts in Non-Stationary Environments», *Pattern Recognition*, vol. 48, n.º 3, págs. 659-669, mar. de 2015.

[RXW+19] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong y Q. Zhang, «Time-Series Anomaly Detection Service at Microsoft», en *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ép. KDD '19, New York, NY, USA: Association for Computing Machinery, jul. de 2019, págs. 3009-3017.

[RLF+19] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon y B. Lakshminarayanan, «Likelihood Ratios for Out-of-Distribution Detection», en *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[RSG16] M. T. Ribeiro, S. Singh y C. Guestrin, «"Why Should I Trust You?": Explaining the Predictions of Any Classifier», en *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ép. KDD '16, New York, NY, USA: Association for Computing Machinery, ago. de 2016, págs. 1135-1144.

[RPG16] R. P. Ribeiro, P. Pereira y J. Gama, «Sequential anomalies: A study in the Railway Industry», *Machine Learning*, vol. 105, n.º 1, págs. 127-153, oct. de 2016.

[Rob59] S. W. Roberts, «Control Chart Tests Based on Geometric Moving Averages», *Technometrics*, vol. 1, n.º 3, págs. 239-250, 1959.

[RPZ+22] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox y P. Gehler, «Towards Total Recall in Industrial Anomaly Detection», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, págs. 14 318-14 328.

[RVG+18] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller y M. Kloft, «Deep One-Class Classification», en *International Conference on Machine Learning*, jul. de 2018, págs. 4393-4402.

[RVG+19] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller y M. Kloft, «Deep Semi-Supervised Anomaly Detection», *arXiv:1906.02694 [cs, stat]*, jun. de 2019. arXiv: 1906.02694 [cs, stat].

[SY14] M. Sakurada y T. Yairi, «Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction», en *Proceedings of the MLSDA 2014 2Nd Workshop on Machine Learning for Sensory Data Analysis*, ép. MLSDA'14, New York, NY, USA: ACM, 2014, 4:4-4:11.

[SPS+01] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola y R. C. Williamson, «Estimating the Support of a High-Dimensional Distribution», *Neural Comput.*, vol. 13, n.º 7, págs. 1443-1471, jul. de 2001.

[SSL19] S.-Y. Shih, F.-K. Sun y H.-y. Lee, «Temporal pattern attention for multivariate time series forecasting», *Machine Learning*, vol. 108, n.º 8, págs. 1421-1441, sep. de 2019.

[SO17] N. Singh y C. Olinsky, «Demystifying Numenta Anomaly Benchmark», en *Neural Networks (IJCNN), 2017 International Joint Conference On*, IEEE, 2017, págs. 1570-1577.

[SVL14] I. Sutskever, O. Vinyals y Q. V. Le, «Sequence to Sequence Learning with Neural Networks», en *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014.

[TCF+02] J. Tang, Z. Chen, A. W.-c. Fu y D. W. Cheung, «Enhancing Effectiveness of Outlier Detections for Low Density Patterns», en *Lecture Notes in Computer Science*, 2002, págs. 535-548.

[TLZ+19] N. Tatbul, T. J. Lee, S. Zdonik, M. Alam y J. Gottschlich, «Precision and Recall for Time Series», *arXiv:1803.03639 [cs]*, ene. de 2019. arXiv: 1803.03639 [cs].

[TD01] D. M. Tax y R. P. Duin, «Combining One-Class Classifiers», en *International Workshop on Multiple Classifier Systems*, Springer, 2001, págs. 299-308.

[TL18] S. J. Taylor y B. Letham, «Forecasting at Scale», *The American Statistician*, vol. 72, n.º 1, págs. 37-45, ene. de 2018.

[TYK+16] A. Theofilatos, G. Yannis, P. Kopelias y F. Papadimitriou, «Predicting Road Accidents: A Rare-events Modeling Approach», *Transportation Research Procedia*, Transport Research Arena TRA2016, vol. 14, págs. 3399-3405, ene. de 2016.

[VVK+19] A. Van Looveren, G. Vacanti, J. Klaise, A. Coca y O. Cobb, *Alibi Detect: Algorithms for Outlier Adversarial and Drift Detection*, 2019.

[WMR17] S. Wachter, B. Mittelstadt y C. Russell, «Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR», *Harv. JL & Tech.*, vol. 31, pág. 841, 2017.

[WM19] B. Wang y Z. Mao, «Outlier detection based on a dynamic ensemble model: Applied to process monitoring», *Information Fusion*, vol. 51, págs. 244-258, nov. de 2019.

[WZZ+19] X. Wei, L. Zhou, Z. Zhang, Z. Chen e Y. Zhou, «Early prediction of epileptic seizures using a long-term recurrent convolutional network», *Journal of Neuroscience Methods*, vol. 327, pág. 108 395, nov. de 2019.

[WSP22] P. Wenig, S. Schmidl y T. Papenbrock, «TimeEval: A Benchmarking Toolkit for Time Series Anomaly Detection Algorithms», *Proceedings of the VLDB Endowment*, vol. 15, n.º 12, págs. 3678-3681, ago. de 2022.

[Whi12] T. White, *Hadoop: The Definitive Guide*. .ºReilly Media, Inc.", 2012.

[XXM+19] K. Xu, M. Xia, X. Mu, Y. Wang y N. Cao, «EnsembleLens: Ensemble-based Visual Exploration of Anomaly Detection Algorithms with Multidimensional Data», *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, n.º 1, págs. 109-119, ene. de 2019.

[XLY19] X. Xu, H. Liu y M. Yao, «Recent Progress of Anomaly Detection», *Complexity*, vol. 2019, e2686378, ene. de 2019.

[ZCF+10] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker e I. Stoica, «Spark: Cluster Computing with Working Sets.», *HotCloud*, vol. 10, n.º 10-10, pág. 95, 2010.

[ZWL+13] N. Zeng, Z. Wang, Y. Li, M. Du, J. Cao y X. Liu, «Time Series Modeling of Nano-Gold Immunochromatographic Assay via Expectation Maximization Algorithm», *IEEE Transactions on Biomedical Engineering*, vol. 60, n.º 12, págs. 3418-3424, 2013.

[ZWZ16] N. Zeng, Z. Wang y H. Zhang, «Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter», *Science China Information Sciences*, vol. 59, n.º 11, pág. 112 204, oct. de 2016.

[ZLN+19] J. Zhang, Z. Li, K. Nai, Y. Gu y A. Sallam, «DELR: A double-level ensemble learning method for unsupervised anomaly detection», *Knowledge-Based Systems*, vol. 181, pág. 104 783, oct. de 2019.

[ZBR+17] S. Zhang, S. Bahrampour, N. Ramakrishnan, L. Schott y M. Shah, «Deep Learning on Symbolic Representations for Large-Scale Heterogeneous Time-Series Event Prediction», en *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, mar. de 2017, págs. 5970-5974.

[ZH18] Y. Zhao y M. K. Hryniewicki, «XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning», en *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, págs. 1-8.

[ZHC+21] Y. Zhao, X. Hu, C. Cheng, C. Wang, C. Wan, W. Wang, J. Yang, H. Bai, Z. Li y C. Xiao, «SUOD: Accelerating Large-Scale Unsupervised Heterogeneous Outlier Detection», *Proceedings of Machine Learning and Systems*, vol. 3, págs. 463-478, 2021.

[ZNL19] Y. Zhao, Z. Nasrullah y Z. Li, «PyOD: A Python Toolbox for Scalable Outlier Detection», *arXiv:1901.01588 [cs, stat]*, ene. de 2019. arXiv: 1901.01588 [cs, stat].

[ZG20] Z. Zohrevand y U. Glässer, «Dynamic Attack Scoring Using Distributed Local Detectors», en *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, mayo de 2020, págs. 2892-2896.

[ZSM+18] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho y H. Chen, «Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection», en *International Conference on Learning Representations*, feb. de 2018.