Desarrollo de un algoritmo para detección de anomalías con base en estimación de densidad basada en kernels, matrices de densidad y medidas cuánticas

Oscar Alberto Bustos Briñez

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2023

Desarrollo de un algoritmo para detección de anomalías con base en estimación de densidad basada en kernels, matrices de densidad y medidas cuánticas

Oscar Alberto Bustos Briñez

Tesis presentada como requisito parcial para optar al título de:
Magister en Ingeniería de Sistemas y Computación

Director(a):
Fabio Augusto González Osorio, Ph.D.

Codirector(a):
Joseph Alejandro Gallego Mejía, Ph.D.(c)

Línea de Investigación:
Computación Teórica
Grupo de Investigación:
MindLab

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2023

Este trabajo está dedicado a mi familia, que me ha amado y apoyado incondicionalmente a lo largo de todos estos años, y a quienes nunca podré agradecer lo suficiente.

Y también está dedicado a aquel muchacho triste que una vez, a pesar del miedo, decidió dejar atrás la oscuridad y mirar al futuro.

Agradecimientos

Este trabajo no habría sido posible sin el impulso del profesor Fabio González, que fue la primera persona en presentarme el campo del aprendizaje de máquina y todas sus emocionantes posibilidades, y no habría llegado tan lejos sin el decidido apoyo del profesor Alejandro Gallego, cuyos conocimientos, disciplina y experiencia han sido infinitamente valiosos tanto a nivel personal como profesional. También expreso mi gratitud con los miembros del grupo de investigación MindLab, especialmente Diego Useche y Santiago Toledo, por sus relevantes aportes y el ejemplo de su ética de trabajo. Finalmente, un enorme agradecimiento a la Universidad Nacional de Colombia y a la Facultad de Ingeniería, que me han permitido llevar a cabo esta increíble travesía y me han dado más oportunidades de las que nunca imaginé.

Título en Español:
Desarrollo de un algoritmo para detección de anomalías con base en estimación de densidad basada en kernels, matrices de densidad y medidas cuánticas

Resumen

Esta tesis presenta un algoritmo innovador diseñado para realizar detección de anomalías en diversos conjuntos de datos. Este método, denominado *Anomaly Detection through Density Matrices and Fourier Features* (AD-DMKDE), integra estimación de densidad basada en kernels (en inglés *Kernel Density Estimation* o KDE) y aprendizaje de máquina (conocida como *Machine Learning* en inglés) con las matrices de densidad y la medición cuántica, dos prometedores conceptos provenientes del campo de la computación cuántica. Se establecen las bases teóricas y metodológicas que sustentan este método; asimismo, se presentan los detalles de su desarrollo e implementación. Se realiza una comparación sistemática del algoritmo propuesto contra doce métodos variados de detección de anomalías; AD-DMKDE muestra un rendimiento competitivo al ser aplicado sobre una selección de veinticuatro conjuntos de datos. Se establecen las fortalezas y limitaciones del algoritmo propuesto y, a partir del análisis estadístico de su rendimiento, se enuncian una serie de conclusiones y posibles líneas de trabajo futuro.

Palabras clave: detección de anomalías, algoritmos de aprendizaje automático, estimación de densidad, aprendizaje automático cuántico, análisis de datos.

Title in English:
Development of an anomaly detection algorithm based on kernel density estimation, density matrices and quantum measurement

Abstract

This thesis presents a novel algorithm designed to perform anomaly detection on multiple data sets. This method, called *Anomaly Detection through Density Matrices and Fourier Features* (AD-DMKDE), integrates Kernel Density Estimation (KDE) and Machine Learning with density matrices and quantum measurement, two promising concepts from quantum computing. The theoretical and methodological foundations that support this method are established, along with the details of its development and implementation. A systematic comparison of the proposed algorithm with twelve state-of-the-art anomaly detection methods is presented, and AD-DMKDE demonstrates competitive performance when applied on twenty-four benchmark data sets. The strengths and limitations of the proposed algorithm are identified, and based on a statistical analysis of its performance, a series of conclusions and possible lines of future work are stated.

Keywords: anomaly detection, machine learning algorithms, density estimation, quantum machine learning, data analysis.

Esta tesis de maestría se sustentó el 24 de octubre de 2023 a las 9:00 a.m., y fue evaluada por los siguientes jurados:

Germán Jairo Hernández Pérez Ph.D.
Universidad Nacional de Colombia, Facultad de Ingeniería

Jorge Eliécer Camargo Mendoza Ph.D.
Universidad Nacional de Colombia, Facultad de Ingeniería

Contenido

Agradecimientos vii

Resumen ix

1 Introducción 1
1.1 Definición del Problema . . . 2
1.2 Objetivos . . . 3
1.3 Contribuciones y Producción Académica . . . 3
1.4 Estructura del Documento . . . 5

2 Antecedentes y Marco Teórico 6
2.1 Detección de Anomalías . . . 6
2.1.1 ¿Qué es una anomalía? . . . 6
2.1.2 Algoritmos de Detección de Anomalías . . . 8
2.1.3 Métodos de línea base . . . 14
2.2 Computación Cuántica . . . 16
2.2.1 Conceptos Básicos . . . 16
2.2.2 Medición Cuántica y Regla de Born . . . 18
2.2.3 Matrices de Densidad . . . 19

3 Anomaly Detection through Density Matrices and Fourier Features 21
3.1 Introducción a AD-DMKDE . . . 21
3.2 Random Fourier Features y Adaptive Fourier Features . . . 22
3.3 Construcción de la Matriz de Densidad . . . 23
3.4 Estimación de Densidad mediante Medición Cuántica . . . 24
3.5 Cálculo del umbral y Predicción . . . 25
3.6 Análisis de Complejidad y Comparación con KDE . . . 26

4 Evaluación Experimental 28
4.1 Conjuntos de Datos . . . 28
4.2 Configuración Experimental . . . 32

5 Resultados 34
5.1 Discusión de Resultados . . . 34
5.2 Análisis estadístico . . . 39

6 Conclusiones y Trabajo Futuro **42**
6.1 Conclusiones 42
6.2 Trabajo Futuro 43

Bibliografía **45**

1 Introducción

En el campo del análisis de datos, se conoce como *anomalías* a los datos, observaciones o patrones que no siguen el comportamiento normal del conjunto de datos del que provienen [18], o cuyas diferencias respecto a otros datos son tan notables como para suponer que sus procesos generativos no son iguales [1]. Aunque se suelen confundir con datos afectados por ruido (también llamados *outliers*), las anomalías a menudo contienen información útil sobre elementos extraños o comportamientos inesperados que influyen en la generación o medición de los datos [78]. El reconocimiento de los elementos anómalos, conocido como Detección de Anomalías (o AD, por sus siglas en inglés), proporciona información útil acerca de los mecanismos subyacentes a la obtención de los datos, por lo que es enormemente útil en diversas aplicaciones [62], incluyendo la detección de alteraciones en sensores, el diagnóstico médico, la detección de intrusos en redes informáticas, la revisión de transacciones en busca de fraudes o el análisis de imágenes, entre otras.

Muchos de los métodos comunes en Detección de Anomalías provienen de variaciones de algoritmos clásicos utilizados en Minería de Datos [2] o para diversas tareas en Aprendizaje de Máquina [17]. Gracias a los avances que se han presentado en años recientes con respecto al aprendizaje automático mediante redes neuronales (impulsados por el desarrollo de arquitecturas profundas, la disponibilidad de grandes bases de datos y el incremento del poder computacional del hardware), ha surgido una amplia variedad de nuevas propuestas en Detección de Anomalías basadas en el uso de redes profundas, que han mostrado resultados importantes en múltiples escenarios [4]. En cualquier caso, la identificación de datos anómalos es una tarea compleja y demandante en recursos computacionales, debido entre otros a la naturaleza heterogénea y cambiante de las anomalías, a la ausencia de información previa sobre su naturaleza, y a que se presentan con relativamente poca frecuencia, de modo que conforman una proporción muy pequeña dentro de los conjuntos de datos [66].

Una de las estrategias más comunes dentro de la detección de anomalías se conoce como Estimación de Densidad, cuyo objetivo principal es el análisis de la distribución de los datos mediante la construcción de una función de densidad que modela dicha distribución, bajo el supuesto de que los datos provienen de un proceso estocástico no conocido [61]. A partir de esta aproximación, es posible determinar las áreas en las que esta función presenta valores bajos, de tal forma que los datos situados en estas regiones de baja densidad se consideran como anomalías [54]. Los métodos habituales utilizados en Estimación de Densidad suelen

ser de dos tipos: paramétricos (cuando se supone que los datos siguen una distribución estadística conocida cuyos parámetros se busca estimar) o no paramétricos (en donde los datos determinan *a posteriori* la función de densidad, que generalmente no posee forma analítica).

La aproximación usual para realizar estimación de densidad no paramétrica se conoce como *Kernel Density Estimation* (KDE). Este método reconstruye la distribución de los datos mediante el uso de funciones auxiliares denominadas *kernels*, que definen transformaciones de los datos originales hacia nuevos espacios donde las relaciones entre diferentes muestras se pueden determinar con mayor facilidad [6]. KDE permite modelar funciones de densidad prescindiendo de las limitaciones de las distribuciones conocidas, por lo que se utiliza mayormente en escenarios en los que no se tiene información previa sobre la distribución de los datos o éstos no son fáciles de tratar mediante métodos estadísticos clásicos [40].

1.1. Definición del Problema

Como método para Estimación de Densidad y Detección de Anomalías, KDE presenta varias ventajas, incluyendo una sólida fundamentación matemática y una interpretación relativamente sencilla de sus resultados. Sin embargo, KDE es un método computacionalmente costoso, altamente sensible al ruido y que trabaja mejor con datos de baja dimensionalidad (es decir, datos que se describen con relativamente pocas características), lo que implica que puede ser muy ineficiente ante conjuntos de datos de altas dimensiones, como imágenes o video [19]. En los últimos años, múltiples propuestas han sido planteadas en aras de superar estas limitaciones, mayormente enfocadas en la integración de KDE con redes neuronales profundas que realizan extracción de características, como los autoencoders variacionales o VAE [86], o las redes recurrentes LSTM [53].

Por otra parte, una alternativa menos explorada gira en torno a la computación cuántica, un paradigma de procesamiento de información que utiliza fenómenos de la física cuántica para resolver problemas computacionales, y cuyos algoritmos han mostrado notables mejoras de desempeño respecto de sus contrapartes clásicas en un amplio rango de problemas [64]. En particular, las matrices de densidad, un mecanismo matemático que combina probabilidad y álgebra lineal para describir estados cuánticos de forma compacta, en combinación con un proceso de carácter cuántico denominado "medición cuántica" (*quantum measurement* en inglés), que permite obtener las probabilidades de diversos escenarios a partir de una matriz de densidad dada, han mostrado un alto potencial para representar eficientemente procesos estocásticos. Este esquema ha sido planteado y desarrollado en detalle en los trabajos seminales de González *et al.* [30, 31].

En este sentido, la interacción entre algoritmos de estimación de densidad basados en KDE y estos conceptos provenientes de la computación cuántica (las matrices de densidad y la

medición cuántica) puede ser un sólido punto de partida para nuevos desarrollos en Detección de Anomalías, de forma que los métodos surgidos de dicha interacción puedan aplicarse en múltiples escenarios (es decir, ante conjuntos de datos con diferentes tamaños, dimensiones y estructuras internas) con un desempeño competitivo y consistente, conservando la mayor parte de las ventajas de KDE mientras abordan algunas de sus limitaciones.

1.2. Objetivos

El objetivo principal de esta tesis es el desarrollo de un algoritmo integrador de detección de anomalías, cuya base es DMKDE [30], un método para realizar estimación de densidad de forma eficiente y robusta a través del uso de matrices de densidad como mecanismo de almacenamiento y de la medición cuántica como mecanismo de lectura. Para lograrlo, se plantean los siguientes objetivos específicos:

- Proponer y/o adaptar estrategias de detección de anomalías basadas en KDE (que pueden incluir o no el uso de redes neuronales como parte de su arquitectura), definiendo sus características teóricas, posibles escenarios de aplicación, parámetros de interés y eventuales limitaciones.
- Integrar las diferentes estrategias propuestas con DMKDE o posibles variantes, de tal forma que el proceso de medición cuántica sirva como una aproximación a la estimación de densidad desde una perspectiva de computación cuántica.
- Evaluar la efectividad del algoritmo propuesto a través de su aplicación en diferentes conjuntos de datos, utilizando diversas métricas de desempeño, y comparándolo con los algoritmos más utilizados en el área de detección de anomalías.

1.3. Contribuciones y Producción Académica

La principal contribución académica de esta tesis corresponde al artículo resumen ***Bustos-Brinez, O.**, Gallego-Mejia, J., & González, F. A. (2022). Anomaly Detection through Density Matrices and Kernel Density Estimation (AD-DMKDE). Neural Information Processing Systems Conference: LatinX in AI (LXAI) Research Workshop 2022, New Orleans, USA* [15] y al artículo de conferencia ***Bustos-Brinez, O.**, Gallego-Mejia, J., & González, F. (2023). AD-DMKDE: Anomaly Detection Through Density Matrices and Fourier Features. En Information Technology and Systems: Proceedings of ICITS 2023. Springer International Publishing, 2023* [16] (en proceso de publicación). Estos dos artículos presentan los principales elementos del algoritmo propuesto en este documento, incluyendo su estructura interna, su formulación algorítmica y su aplicación en un esquema experimental más limitado que el que se presenta aquí. Además de estos artículos, se creó un póster que resume los elementos

principales del método y muestra su desempeño en la evaluación experimental; el póster fue presentado en la conferencia *LatinX in AI (LXAI) Research Workshop 2022*, y puede encontrarse en la dirección web https://www.researchgate.net/publication/367453145_Anomaly_Detection_through_Density_Matrices_and_Kernel_Density_Estimation_AD-DMKDE.

Por otra parte, el resultado práctico de esta tesis comprende la implementación en código (lenguaje Python) del algoritmo presentado y analizado en este documento. El repositorio en el que se encuentra este código incluye también archivos con los conjuntos de datos elegidos para la experimentación, y referencias a las implementaciones de DMKDE y de los algoritmos de detección de anomalías utilizados como línea base. Se puede acceder al repositorio a través de la dirección web https://github.com/oabustosb/AD-DMKDE.

Además del diseño, análisis e implementación del algoritmo presentado en este documento, se realizó una serie de contribuciones adicionales como parte de la producción de diversos artículos de investigación. A continuación se detalla una lista de estos artículos y los aportes realizados en cada uno.

- *Useche, D. H.,* ***Bustos-Brinez, O. A.****, Gallego, J. A., & González, F. A. (2022). Computing expectation values of adaptive Fourier density matrices for quantum anomaly detection in NISQ devices. En arXiv: 2201.10006* [88]. Este artículo presenta un modelo para estimación de densidad similar a DMKDE, en el que un circuito cuántico se encarga de construir la matriz de densidad requerida (se plantean dos formas posibles para esto) y de ejecutar el proceso de medición cuántica, siguiendo para ello las reglas propias de la computación cuántica. Este mecanismo fue complementado para realizar detección de anomalías, y probado en un conjunto de datos específico. El proyecto fue la culminación del proceso desarrollado para la Qiskit Hackathon Global 2021, donde el equipo obtuvo el cuarto lugar en la competencia y una mención honorífica.
- *Gallego-Mejia, J.,* ***Bustos-Brinez, O.****, & González, F. A. (2022). LEAN-DMKDE: Quantum Latent Density Estimation for Anomaly Detection. En arXiv: 2211.08525* [27]. Este artículo define un esquema para detección de anomalías basado en DMKDE, con un autoencoder (red neuronal profunda) como primera etapa en el procesamiento de los datos, y la inclusión del error de reconstrucción como parte del análisis de la normalidad de los datos. Para este artículo, la contribución realizada se centró en el planteamiento y desarrollo de la evaluación experimental, el análisis de resultados y el *ablation study*. Además, se construyó un póster que resume los desarrollos presentados en el artículo, el cual fue presentado en la conferencia *The 37th AAAI Conference on Artificial Intelligence*, realizada en Washington, USA, en febrero de 2023.
- *Gallego-Mejia, J.,* ***Bustos-Brinez, O.****, & Gonzalez, F. (2022). InQMAD: Incremental Quantum Measurement Anomaly Detection. En "2022 IEEE International Conference on Data Mining Workshops (ICDMW)" (pp. 787-796)* [28]. Este artículo presenta

una propuesta para abordar el problema de detección de anomalías en streams de datos a partir de una variación de DMKDE, con enfoque en la rapidez y adaptabilidad del método. Similar al caso previo, la contribución realizada incluyó el planteamiento y desarrollo de la evaluación experimental.

1.4. Estructura del Documento

Este trabajo está dividido en seis capítulos. El Capítulo 1 presenta una introducción de los conceptos básicos de la investigación, junto con la definición del problema, los objetivos y las contribuciones académicas realizadas. El Capítulo 2 presenta en detalle las bases teóricas y metodológicas sobre las que el proyecto ha sido desarrollado, incluyendo un panorama de los algoritmos en detección de anomalías, los métodos de linea base seleccionados, y una introducción a la computación cuántica con énfasis en matrices de densidad y medición cuántica. El Capítulo 3 presenta la descripción del diseño y arquitectura del algoritmo AD-DMKDE, incluyendo su formulación matemática, las etapas que lo componen y el análisis de su complejidad computacional. Los Capítulos 4 y 5 presentan, en su orden, la preparación de la evaluación experimental del método incluyendo los conjuntos de datos utilizados, y los resultados obtenidos en dicha evaluación junto con un análisis estadístico de los mismos. Finalmente, el Capítulo 6 presenta las conclusiones del trabajo realizado y plantea posibles líneas de trabajo futuro.

2 Antecedentes y Marco Teórico

Este Capítulo presenta las bases conceptuales que soportan la metodología propuesta, con un particular énfasis en la detección de anomalías y la computación cuántica. Para cada uno de estos conceptos, una descripción comprensiva es presentada, incluyendo algunos de sus escenarios de aplicación y sus relaciones con otras áreas del conocimiento. Se profundiza en los métodos más comunes para detección de anomalías, con una breve descripción de cada método, y en los conceptos cuánticos necesarios para el algoritmo propuesto, esto es, las matrices de densidad y la medición cuántica.

2.1. Detección de Anomalías

2.1.1. ¿Qué es una anomalía?

Una definición formal de anomalía puede darse desde una perspectiva probabilística [78]. Si se define un espacio de datos $\mathcal{X} \subseteq \mathbb{R}^n$, la normalidad de los datos estará dada por una distribución $\mathbf{P}$ sobre $\mathcal{X}$ que define el comportamiento "normal" en el espacio de datos, de tal forma que los puntos de datos normales se consideran como muestras de dicha distribución. A partir de $\mathbf{P}$ se puede señalar las regiones de $\mathcal{X}$ donde los datos normales tienden a concentrarse, así como regiones donde es poco probable que aparezcan; de este modo, las anomalías se definen como los puntos de datos que se encuentran en regiones de baja probabilidad bajo la distribución $\mathbf{P}$. Más formalmente, un punto $\mathbf{x}_0 \in \mathcal{X}$ es anómalo si su probabilidad $\mathbf{P}(\mathbf{x}_0)$ es menor a un valor de umbral τ, el cual define la frontera a partir de la cual la probabilidad del dato es lo suficientemente pequeña para considerarlo como significativamente diferente. En general, los procesos generativos de los datos son altamente complejos y no se conocen de antemano, por lo que los métodos de detección de anomalías suelen construir, en mayor o menor grado, aproximaciones a la distribución $\mathbf{P}$ con base en el conjunto de datos dado.

Esta definición basada en distribuciones de probabilidad puede expandirse para considerar múltiples tipos de anomalías con diversas características. En la literatura, se suele distinguir tres categorías generales de anomalías [18, 33]: anomalías *puntuales*, en las que un dato individual se distingue de los demás (por ejemplo una transacción bancaria fraudulenta o un producto dañado); anomalías *contextuales*, donde los datos son anómalos dependiendo de su situación espacial o temporal (por ejemplo, que una temperatura particular sea extrema o no depende de un contexto geográfico y temporal particular); y anomalías *colectivas*, en las

cuales un grupo de datos es anómalo con respecto a todos los demás, pero cuyos puntos son normales dentro de dicho grupo (por ejemplo, una secuencia de ataques a una red provenientes de la misma fuente). Para las anomalías contextuales, la función de probabilidad asociada a la distribución $\mathbf{P}$ se transforma en una distribución condicional $\mathbf{P}(\mathbf{x}|\mathbf{t})$ sobre alguna variable aleatoria T que representa el entorno espacio-temporal del dato; para las anomalías colectivas, se transforma en una distribución conjunta condicional $\mathbf{P}(\mathbf{x}_i, \mathbf{x}_{i+1}, ..., \mathbf{x}_{i+n}|\mathbf{t})$ que considera secuencias de datos. A estas tres categorías generales se les suele agregar algunas nuevas categorías orientadas a ambientes específicos, como en el caso de las señales provenientes de sensores [23], donde se separan anomalías puntuales (como picos súbitos de magnitud) de secuencias anómalas (como periodos constantes sin cambios), o en datos multimodales [96], donde se distinguen las anomalías de bajo nivel (como pixeles inusuales en imágenes) de las anomalías semánticas o de alto nivel [3] (como determinar si un texto es spam o no).

Las aproximaciones a la detección de anomalías dependen notablemente del tipo de datos sobre el que realiza el análisis [41]. Si los datos pueden representarse como una serie de valores numéricos no dependientes del tiempo, y puede definirse un concepto de similaridad entre diferentes datos, entonces las aproximaciones en detección de anomalías pueden basarse en el cálculo de las distancias entre los puntos o las correlaciones entre ellos [48]. Por otra parte, si los datos se presentan en secuencia y su orden es relevante, la detección puede realizarse a partir del análisis de datos puntuales, de secuencias enteras (conocidas como *ventanas de tiempo*) [81] o mediante el análisis *on-line*, útil para tratar grandes cantidades de datos que se generan constantemente y que no pueden almacenarse [75]. Finalmente, otros tipos de datos más complejos requieren análisis específicos, como el caso de las imágenes (donde las anomalías dependen de grupos de pixeles representando distintas jerarquías de conceptos como líneas y formas) [92], del texto (con análisis de caracteres, de palabras o de significados más complejos) [67], o de los grafos (donde se pueden analizar nodos individuales o subgrafos como estructuras anómalas dentro de un esquema mayor) [57].

Toda propuesta en el área debe tener en cuenta múltiples factores del conjunto de datos a utilizar [78], incluyendo las siguientes:

- el tamaño del conjunto de datos y sus posibilidades en términos de escalabilidad,
- si la dimensionalidad de los datos es alta o baja, y si puede reducirse sin perder mucha información mediante mecanismos de reducción dimensional,
- si los datos son categóricos (nominales u ordinales) o numéricos (continuos o discretos),
- si se buscan anomalías de alto nivel (diferencias semánticas, subgrafos) o de bajo nivel (píxeles, caracteres, lexemas o nodos),
- si la distribución de los datos varía en el tiempo o es estacionaria,

- si la distinción de un punto dado como anómalo o normal depende de la totalidad de los datos o sólo de un grupo pequeño de datos relativamente similares a éste,
- hasta qué grado la presencia de ruido tiene un efecto sobre la separabilidad de los datos.

Además, la detección de anomalías es una tarea que posee múltiples grados de libertad, especialmente en cuanto a la elección del umbral de separación τ (o parámetros de los que éste puede depender) y el mecanismo de reconstrucción de la distribución $\mathbf{P}$. Por esta razón, es necesario realizar una serie de suposiciones sobre la distribución de los datos normales, la frecuencia y distribución de las anomalías, y el manejo que se da a los errores del modelo, tanto falsos positivos (las detecciones erróneas) como falsos negativos (las anomalías que se pasen por alto), entre otros [85].

Por lo general, la salida de un método de detección de anomalías ante un nuevo dato suele ser de dos tipos: o bien una etiqueta que identifica al dato como normal o como anómalo, o bien un puntaje o *score*, que indica el grado en que el dato puede considerarse anómalo. Los scores suelen presentar mayor información acerca de los datos anómalos, permitiendo construir rankings o encontrar valores apropiados del umbral τ [29, 10]. La forma de evaluar la calidad de estos métodos depende de la disponibilidad de información a priori sobre cuáles datos son verdaderamente anómalos; si se conoce qué datos lo son, es posible aplicar las métricas usuales en problemas de clasificación, particularmente *precision* o *recall*, que permiten abordar el desbalance de clases [4]. Si no se conoce esta información, existen otras medidas dependientes del modelo elegido [10] o del contexto del problema abordado [25].

2.1.2. Algoritmos de Detección de Anomalías

Existen múltiples tipologías para categorizar los algoritmos en detección de anomalías, dependiendo de diversos aspectos. Uno de estos aspectos básicos se relaciona con la presencia de información sobre las verdaderas etiquetas de los datos, es decir, si son realmente anómalos o no. En general, cualquier mecanismo de aprendizaje automático puede enfrentarse a cuatro escenarios, de acuerdo con [17]:

- Aprendizaje Supervisado: en este escenario, se conocen las etiquetas de todos los datos, es decir, se sabe a priori para cada dato si es anómalo o no. Esto permite evaluar la calidad de los métodos contrastando sus predicciones con el conocimiento previo; por esta razón, la evaluación del algoritmo propuesto se realiza utilizando este esquema. Sin embargo, la obtención de datos etiquetados es un proceso costoso e intensivo.
- Aprendizaje Semi Supervisado: sólo se conocen las etiquetas de algunos de los datos, de modo que se busca inferir una relación general entre datos y etiquetas, o bien determinar las etiquetas faltantes, bajo la suposición de que los datos con la misma etiqueta tienden a encontrarse cerca o dentro de un mismo grupo.

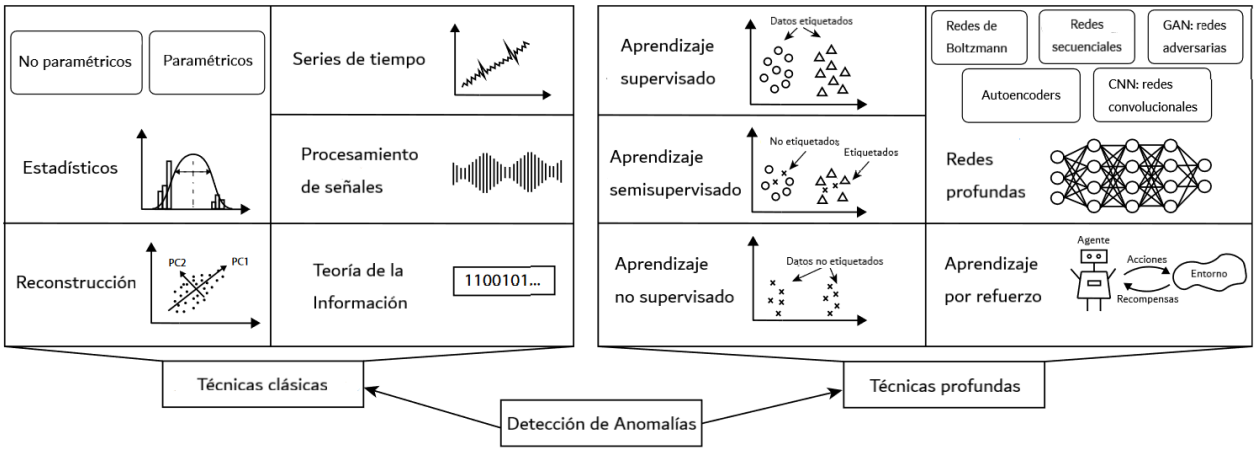

Figura 2-1: Caracterización de los principales tipos de algoritmos de detección de anomalías. Tomado de [23].

- Aprendizaje No Supervisado: no se conoce ninguna etiqueta para los datos, por lo que la orientación en este caso consiste en encontrar los patrones o reglas subyacentes de los datos. Esto puede incluir técnicas como la reducción dimensional, o abordar problemas como la construcción de grupos de datos similares (también llamada *clustering*).

- Aprendizaje por Refuerzo: este escenario se utiliza en problemas que pueden expresarse mediante la interacción de un agente y su entorno. El agente busca aprender una serie de acciones que modifiquen su entorno de tal manera que se maximice una función de recompensa definida, por lo que su entrenamiento puede ser muy lento dependiendo de cómo se exploran las posibles alternativas. Este es un enfoque que se utiliza mayormente en problemas complejos que no admiten soluciones analíticas [8].

En cuanto a las técnicas subyacentes a cada método de detección, la categorización más habitual distingue entre métodos clásicos basados en diversas consideraciones teóricas, y métodos profundos que se apoyan en el uso de redes neuronales. Cada una de estas clasificaciones también presenta diversas subcategorías, como se puede ver en la Figura **2-1**. A continuación, se presenta un panorama general de estos dos tipos de métodos, reseñando algunas de sus subcategorías.

Métodos Clásicos

Los métodos clásicos o convencionales suelen clasificarse en base a su formulación, es decir, el área matemática, estadística o informática de la que surgen [87]. Así, se pueden distinguir, entre muchas otras, las siguientes categorías:

Métodos Estadísticos. Estos modelos tratan de aproximar directamente la distribución **P** de los datos normales, ya sea a partir de distribuciones bien conocidas (métodos pa-

ramétricos) o construyendo una distribución directamente a partir de los datos (métodos no paramétricos). La distribución más utilizada en los métodos paramétricos es la normal o gaussiana multivariada, cuya función de densidad está dada por:

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma^{1/2}|} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right)$$

donde n corresponde a la dimensionalidad del vector $\mathbf{x}$, y los parámetros μ y Σ corresponden al vector media y a la matriz de covarianza, respectivamente. En este caso, un *score* de anomalía común para el punto $\mathbf{x}$ puede ser su distancia (comúnmente, la distancia de Mahalanobis) al punto central de la distribución, el vector media. Este *score* está altamente relacionado con la función de densidad $f_X(\mathbf{x})$ evaluada en $\mathbf{x}$ [24, 70]. En caso de que la distribución de datos normales no sea bien aproximada por una distribución gaussiana, una aproximación habitual es construir un "modelo de mixtura gaussiana" [76], en el que se construye una aproximación a la distribución formada por sumas de múltiples gaussianas con diferentes valores de μ y Σ, y el *score* de un punto está dado por la distancia al valor medio más cercano.

Por otra parte, los modelos no paramétricos tratan de derivar una distribución (no necesariamente conocida) a partir de los datos. La aproximación más sencilla de este tipo son los histogramas, en los cuales los datos se ubican dentro de bins de un ancho dado, y un dato es anómalo si no cae en ninguno de los bins o cae en un bin de baja frecuencia. Cada bin representa la probabilidad de los datos en una cierta región, de modo que el histograma puede aproximar a la distribución [87]. Sin embargo, el método no paramétrico más común es la Estimación de Densidad basada en Kernels (conocida como *Kernel Density Estimation* o KDE), en el que la agrupación de los datos en bins se deja de lado a favor de medir la aportación de cada punto dada su ubicación [19]. Para ello, se define una función, el *kernel*, centrada en cada dato; la aproximación a la distribución se construye sumando todos los kernels asociados a todos los datos, de tal forma que los valores más altos de la aproximación se centran en las regiones con mayor densidad de los datos. De esta forma, es posible aproximar distribuciones para las que no hay forma conocida (como cuando se presentan varias modas [40]) o distribuciones condicionales cuando se desea evaluar la influencia de variables externas (un ejemplo interesante de este escenario se presenta en [37]).

En KDE, un kernel k es una función real estrictamente positiva que cumple dos propiedades importantes: que el área bajo su curva sea finita, y que sea simétrica alrededor del valor cero (es decir, $k(x) = k(-x)$ para todo x). Los kernels representan la aportación de los puntos de datos a la función de densidad, de tal forma que las áreas con más puntos aportan un peso mayor en la aproximación. La geometría de los kernels depende no sólo de su forma analítica, sino también de un parámetro de *bandwidth*, comúnmente denotado como h, que determina en qué grado la influencia de los puntos se disemina por su vecindad [83]. Los

valores bajos de h concentran la aportación de un punto en éste, llevando a distribuciones con alta variabilidad que pueden contener variaciones poco relevantes; los valores altos de h, por otro lado, dispersan la aportación del punto en un gran área, llevando a distribuciones muy suaves que pueden pasar por alto la verdadera estructura de los datos [91]. La elección de un h adecuado es una tarea difícil que suele llevarse a cabo probando varios valores y evaluando la calidad de cada aproximación generada.

Hallar el *score* de un punto $\mathbf{y}$ requiere primero de la estimación del valor de la distribución en ese punto, denotada como $\hat{f}_h(\mathbf{y})$. Dado un conjunto de puntos de entrenamiento $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$, dicha estimación se realiza a través del "estimador de Parzen-Rosenblatt" [82]:

$$\hat{f}_h(\mathbf{y}) = \frac{1}{nM_h} \sum_{i=1}^{n} k(\mathbf{x}_i - \mathbf{y}; h)$$

Los kernels más utilizados en este proceso incluyen el kernel gaussiano ($k(x) \propto \exp\left(-\frac{x^2}{2h^2}\right)$), el kernel exponencial ($k(x) \propto \exp\left(-|x|\right)$), o el kernel Epanechnikov ($k(x) \propto 1 - \frac{x^2}{h^2}$) [91]. M_h es una constante de normalización, que garantiza que la estimación mantenga un área bajo la curva igual a 1. La estimación $\hat{f}_h(\mathbf{y})$ converge a la verdadera distribución a medida que aumenta la cantidad de puntos $\mathbf{x}_i$, y el *score* del punto $\mathbf{y}$ suele calcularse como el inverso de su densidad estimada, de tal forma que las anomalías pueden ser explicadas como puntos en áreas de baja densidad. Sin embargo, la estimación requiere calcular un kernel para cada punto de entrenamiento, por lo que este método puede hacerse demasiado costoso para conjuntos de datos grandes.

Métodos basados en Distancia. Estos métodos detectan anomalías a partir de la suposición de que dichos puntos suelen encontrarse distantes de otros, en contraste con los puntos normales que suelen ser cercanos entre sí. Estos modelos dependen de una métrica de distancia previamente definida, que debe tener sentido dentro del contexto del problema [46]. Al contrario que los métodos anteriores, estos no construyen un modelo general de normalidad, sino que se basan en un análisis de la vecindad de los puntos. La definición de esta área cercana al punto es el principal parámetro que determina el comportamiento de los métodos; algunas aproximaciones para establecer dicha área incluyen la revisión de la distancia del punto a sus vecinos más cercanos [72], la cantidad de vecinos que se encuentran a menos de cierta distancia del punto, o pueden depender de métricas de similaridad entre puntos [39] o medidas de la densidad teniendo en cuenta el contexto [11]. En todo caso, estos métodos presentan dos debilidades notables: su complejidad generalmente crece de forma cuadrática con respecto a la cantidad de datos, y su desempeño varía notablemente dependiendo del tamaño del área cercana que se elija, ya que vecindades pequeñas pueden pasar por alto anomalías parecidas entre sí, y vecindades grandes pueden generar detecciones falsas si hay grupos pequeños y distantes de datos normales [65].

Métodos basados en Reconstrucción. La suposición básica de estos métodos indica que el concepto de normalidad de los datos depende de patrones que no requieren de la totalidad de las características de los datos, sino que se pueden expresar en espacios de menor dimensionalidad [7]. Para ello, se utilizan mecanismos de reducción dimensional como el Análisis de Componentes Principales (PCA), en el cual las nuevas dimensiones se construyen como combinaciones lineales de las originales, de tal forma que las nuevas dimensiones concentran la mayor varianza de los datos, y por ende explican la mayoría de sus patrones [59]. Esta reducción dimensional puede ser revertida de forma aproximada, por lo que es posible pasar un punto por el proceso de reducción y luego por su proceso inverso para obtener una reconstrucción de dicho punto. Un dato normal, que se conforma a los patrones de la reducción (es decir, cuya variación está explicada por las dimensiones construidas por PCA), podrá ser reconstruido de forma sencilla, y su diferencia con la reconstrucción será pequeña. En contraste, un dato anómalo no se conforma con estos patrones, por lo que no será bien reconstruido. De esta forma, la diferencia o error de la reconstrucción puede servir como un *score* de anomalías [35]. Este tipo de métodos puede abordar distintos tipos de datos que no se conforman a ninguna distribución particular, pero la obtención de las nuevas dimensiones suele ser lenta si existe una gran cantidad de datos.

Teoría de la Información. De acuerdo con los postulados de esta teoría, si se supone que los datos hacen parte de las posibles salidas de una variable aleatoria, la información que un dato particular posee está relacionada inversamente con la probabilidad de su ocurrencia. De este modo, un dato normal con alta probabilidad no posee una cantidad alta de información, mientras que un dato anómalo, al tener una probabilidad baja de ocurrir, puede poseer una gran cantidad de información [9]. Existen algunas medidas de la información de un conjunto de datos, como la entropía o la complejidad de Kolmogorov, con las que se puede medir la información total de los datos [22]; el *score* de un dato particular se puede determinar midiendo la información de los datos restantes y comprobando cuánta se ha perdido al remover el dato. Estos métodos no requieren de conocimiento previo sobre las anomalías, pero en general las medidas de información son altamente complejas de calcular y no suelen variar mucho si las anomalías son escasas.

Métodos Profundos

Estos métodos están basados en el uso de grandes redes neuronales, y se suelen utilizar en escenarios donde los métodos clásicos no son capaces de abordar los conjuntos de datos debido a su tamaño o porque su desempeño es muy bajo. Una neurona puede verse como un mecanismo que recibe varias entradas y retorna una única salida consistente en una combinación no lineal de sus entradas. Las neuronas suelen organizarse en capas, conjuntos de neuronas apiladas de varias formas; una red neuronal consta de una serie de capas conectadas entre sí de tal forma que las salidas de unas sirven como entradas de las siguientes [89]. Se ha

demostrado que este modelo multicapa (también llamado secuencial) es capaz de aproximar funciones con precisión arbitraria cuando no hay restricciones a la cantidad de neuronas [73].

Las redes neuronales han experimentado un auge notable en los últimos años, con el surgimiento de enormes redes compuestas por millones de neuronas que son capaces de ejecutar con precisión tareas muy complejas (incluyendo el procesamiento de lenguaje natural, la anotación automática de imágenes o el análisis de audio) [84]. Además, el desarrollo de hardware especializado para ejecutar estas redes, incluyendo las conocidas GPU (Unidades de Procesamiento de Gráficos) y TPU (Unidades de Procesamiento de Tensores), que cuentan con una gran capacidad de computación en paralelo [78], ha permitido que se reduzcan considerablemente los tiempos de entrenamiento y ejecución de estos modelos.

Existe una amplia variedad de arquitecturas de redes neuronales, diseñadas de acuerdo con diferentes criterios y para abordar diferentes problemas sobre diferentes tipos de datos. En lo que se refiere a detección de anomalías, las redes neuronales pueden presentar dos tipos diferentes de enfoques [66].

Extracción de Características. En este esquema, las redes neuronales desempeñan un papel similar al de los mecanismos de reducción dimensional como PCA, actuando sólo como una función de mapeo de los datos desde su espacio original a un nuevo espacio en el cual se aplica algún método clásico de detección de anomalías. Dada la capacidad de las redes de aproximar funciones no lineales, este enfoque suele ser mejor que los mecanismos clásicos para hallar relaciones complejas entre los datos, incluyendo patrones semánticos. Dentro de este tipo de esquemas se puede encontrar a las redes convolucionales o CNN, que utilizan capas diseñadas para realizar operaciones específicas de reducción dimensional (llamadas pooling y convolución) que permiten extraer patrones sencillos y características complejas de datos de muy alta dimensionalidad, especialmente imágenes [5] y video [63], transformándolas en representaciones numéricas compactas que pueden ser usadas por otros métodos. Otro tipo de red comúnmente usado en este escenario son las máquinas restringidas de Boltzmann (RBM). Estas se componen de dos capas, en las que las salidas de una capa sirven como entradas de la otra; estas redes pueden juntarse unas con otras para construir redes profundas denominadas *Deep Belief Networks* (DBN), que se han probado exitosamente en tareas de búsqueda de patrones en datos con dependencias complejas [94].

Aprendizaje End-to-End. En este esquema, las redes neuronales no aprenden una representación latente de los datos que no depende del objetivo para el que se ha construido, como en el caso anterior, sino que aprenden su representación de tal forma que se maximiza el aprendizaje de los patrones subyacentes de los datos. Estos patrones aprendidos pueden ser procesados de forma externa a la red neuronal, o se puede diseñar a la red de forma que su salida sea directamente el *score* de anomalía de su entrada.

Los modelos más utilizados que siguen este esquema pertenecen a dos clases. Los autoencoders son redes compuestas de dos partes: un encoder que codifica los datos en espacios latentes de menor dimensión, gracias a que el tamaño de sus capas se va reduciendo progresivamente, y un decoder, una red que busca aprender el proceso inverso tomando como entradas puntos del espacio latente y tratando de obtener los puntos del espacio original [45]. El entrenamiento conjunto de las dos redes obliga al modelo a aprender las características más importantes de los datos para reconstruirlos apropiadamente, de modo que pueden seguir ideas similares a los métodos de reconstrucción clásicos, o servir como etapa inicial para otros algoritmos [4]. Por otra parte, las redes generativas adversarias (GAN) hacen referencia a un esquema en el que dos redes entrenan de forma conjunta: una red generadora construye datos artificiales a partir de analizar los datos originales, y una red discriminante busca separar estos datos generados de los reales. La generadora aprende los patrones del conjunto de datos, por lo que los datos normales son generados con mayor precisión que los anómalos; para la discriminante, estos últimos datos son más fáciles de distinguir [44].

2.1.3. Métodos de línea base

Para llevar a cabo la evaluación experimental del algoritmo propuesto en este documento, se realizó una selección de doce métodos de detección de anomalías, tanto clásicos como basados en redes neuronales. A continuación, se presenta una breve descripción del funcionamiento de cada método.

Métodos Clásicos

- **Minimum Covariance Estimator** [77]**:** este método asume que los datos presentan una distribución normal multivariada en el espacio de datos, por lo que utiliza la distancia de Mahalanobis (que toma en cuenta las correlaciones entre los datos) para construir una serie de curvas de nivel con forma elipsoidal que envuelven progresivamente los datos normales. Una de estas curvas es elegida como la frontera entre datos normales y anómalos.

- **Local Outlier Factor (LOF)** [11]**:** el método requiere que se defina una función de distancia entre puntos de datos, ya que construye una medida de la densidad de los datos calculada a partir de la distancia de cada punto a sus k vecinos más cercanos. Si un punto está lejos de sus vecinos, esta medida de densidad será baja y el dato será establecido como anómalo.

- **Isolation Forest** [55]**:** este algoritmo construye árboles de decisión, donde cada nodo corresponde a una separación aleatoria a partir de una única dimensión elegida al azar. Estas particiones aleatorias se realizan sucesivamente, haciendo más profundo el árbol, hasta que todos los puntos han sido separados. Los puntos que requieren de menos divisiones para ser aislados (que están más arriba dentro del árbol de decisión)

tienen una mayor posibilidad de ser anomalías. Puesto que las divisiones se realizan aleatoriamente, lo usual es construir varios árboles con diferentes divisiones y ponderar sus resultados.

- **OneClassSVM** [80]**:** este es un algoritmo basado en kernels cuyo objetivo es encontrar una frontera cerrada (generalmente, una hiper-esfera en el espacio de características) que encapsula los datos normales sin aproximar explícitamente su distribución, y clasificando nuevos puntos de datos como anomalías si se ubican por fuera de esta frontera. El kernel más usado en este método es el kernel gaussiano (RBF).

- **K-nearest Neighbors (KNN)** [72]**:** también basado en funciones de distancia entre puntos de datos, este método calcula para cada dato un puntaje basado en la distancia entre éste y su k-ésimo vecino más cercano. Utilizando un valor apropiado de k, si la distancia a dicho vecino es alta, se infiere que el punto está lejos de otros y por ende se clasifica el dato como anómalo.

- **Stochastic Outlier Selection (SOS)** [39]**:** el algoritmo construye una matriz de similaridad entre puntos de datos a partir de un concepto denominado "afinidad". La afinidad se calcula para cada par de puntos, y su valor depende de un parámetro llamado *perplexity*. Los puntos de datos con menor afinidad a los otros datos tienen más probabilidad de ser etiquetados como anómalos.

- **LODA** [69]**:** este método, un ejemplo de algoritmos de ensamble, construye un conjunto de clasificadores simples que operan sobre proyecciones de los datos en espacios de bajas dimensiones. Cada pequeño clasificador usa histogramas para detectar los puntos anómalos, y el algoritmo combina las salidas de estos detectores para determinar una medida más robusta de la anormalidad de los datos.

- **COPOD** [52]**:** la base de este algoritmo son las funciones de *copula*, capaces de modelar eficientemente distribuciones de probabilidad univariadas. El algoritmo encuentra una función de *copula* para cada dimensión de los datos, y luego las une en una única distribución conjunta que busca aproximar la verdadera distribución de los datos. De esta forma, el algoritmo etiqueta como anomalías a los puntos de datos que se encuentran en regiones donde dicha función conjunta tiene un valor bajo.

Métodos profundos

- **VAE-Bayes** [45]**:** diseñado a partir del concepto de modelos Bayesianos, este algoritmo supone que los datos provienen de un proceso generativo que puede ser codificado en un espacio latente. La transformación entre el espacio de datos y el espacio latente se realiza mediante un autoencoder variacional (VAE), y los datos codificados pueden ser usados como medio para obtener la distribución de probabilidad original usando suposiciones de tipo bayesiano.

- **DeepSVDD** [79]**:** basado en OneClassSVM, el método utiliza una red neuronal para transformar los datos a un espacio de características en el cual las anomalías pueden ser separadas. La red es entrenada de tal forma que todos los datos normales transformados quedan ubicados cerca de un punto central, y la frontera entre datos normales y anómalos es el borde de una hiperesfera cuyo radio se trata de minimizar.

- **Adversarially Learned Anomaly Detection (ALAD)** [93]**:** este algoritmo utiliza el concepto de redes generativas adversarias (GAN). En este algoritmo, una red codifica los datos normales en un espacio latente, mientras que la generadora trata de crear datos falsos a partir de ese espacio; varias discriminantes tratan de separar los datos reales de los generados, de modo que ante una anomalía, al menos una de ellas sea capaz de detectar que es diferente a los datos originales.

- **LAKE** [56]**:** este algoritmo procesa los datos en dos etapas: un autoencoder variacional que codifica los datos en un espacio latente de menor dimensionalidad, tratando de mantener la distribución de datos original mediante el uso de algunas métricas de distancia; y la aplicación de un método de estimación de densidad (KDE) sobre los datos codificados, expresando la función de densidad como una suma ponderada de funciones gaussianas. Los puntos de datos en regiones de baja densidad se clasifican como anomalías.

2.2. Computación Cuántica

2.2.1. Conceptos Básicos

La computación cuántica se define como el estudio del procesamiento de información que puede llevarse a cabo a través de sistemas que exhiben propiedades cuánticas. La mecánica cuántica, cuyos primeros desarrollos surgieron en la década de 1920, establece un nuevo conjunto de reglas físicas que aplican a escalas de moléculas, átomos y partículas subatómicas, y que se diferencian fundamentalmente de los comportamientos de los objetos físicos a escala macroscópica [64]. En los años setenta, con los primeros experimentos que lograron aislar y manipular con precisión partículas individuales, aparece el campo de la *información cuántica*, que analiza cómo los sistemas sujetos a las leyes cuánticas procesan y almacenan información [42]. Posteriormente, los años ochenta y noventa vieron el surgimiento de los primeros algoritmos cuánticos, aquellos que aprovechan las propiedades cuánticas para resolver problemas de formas más eficientes que cualquier contraparte clásica.

El elemento básico en información cuántica es el *qubit*, una entidad análoga a los bits de la computación clásica, en el sentido de que puede presentar dos posibles estados, denominados ‘cero’ y ‘uno’. Los bits, que son la base de los sistemas computacionales digitales, sólo pueden

presentar uno de esos dos estados; pero los qubits, gracias a una propiedad cuántica denominada "Principio de Superposición", pueden no sólo presentar los estados 'cero' y 'uno', sino cualquier combinación lineal de los dos valores, donde los coeficientes pueden ser valores complejos [43]. Esta multiplicidad de los estados cuánticos posibles se denomina *superposición*. Algunos sistemas cuánticos comunes que exhiben este comportamiento incluyen, por ejemplo, la polarización de los fotones [47] (las partículas que conforman la luz y median la interacción electromagnética) o el espín del electrón [14], una propiedad cuántica que se suele equiparar al sentido del giro de la partícula.

Matemáticamente, un estado cuántico se representa como un vector columna cuyos elementos corresponden a valores complejos que almacenan la contribución de los estados 'cero' y 'uno' en ese orden. Normalmente, los estados cuánticos se representan a través de la *notación de Dirac* [38]: los vectores columna se representan como $|\phi\rangle$, y los vectores fila como $\langle\phi|$. Los estados base 'cero' y 'uno' corresponden a los vectores columna $[1,0]^T$ y $[0,1]^T$ y se representan como $|0\rangle$ y $|1\rangle$, respectivamente. De este modo, un estado en superposición arbitrario $|\phi\rangle$ dado por el vector columna $\phi = [\alpha, \beta]^T$ puede escribirse como una combinación lineal de la siguiente forma:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$$

No todos los valores de α y β conforman estados cuánticos válidos. La restricción impuesta por las reglas cuánticas a estos coeficientes es la siguiente: $|\alpha|^2 + |\beta|^2 = 1$. Esto sugiere que los estados cuánticos están limitados geométricamente a vectores normalizados (es decir, con norma igual a 1); de esta forma, se hace posible expresar el estado $|\phi\rangle$ en términos de números reales mediante la ecuación:

$$|\phi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

en la cual θ y φ son valores reales que se comportan como ángulos, y que definen un único punto sobre una esfera tridimensional de radio 1. Esta construcción se conoce como "esfera de Bloch", y es la representación gráfica usual para los estados cuánticos de un qubit. La Figura **2-2** presenta un ejemplo de la representación de un estado cuántico. Nótese que $|0\rangle$ y $|1\rangle$ actúan como los polos de esta esfera, a pesar de que son vectores ortogonales.

Si se tiene un número mayor de qubits, la representación gráfica dada por la esfera de Bloch no es útil en la mayoría de los casos [58]; pero el estado de un sistema aún puede expresarse como combinación lineal de una serie de estados base que cumplen un rol similar a $|0\rangle$ y $|1\rangle$. Por ejemplo, un estado arbitrario de dos qubits se puede escribir de la forma:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

donde los coeficientes α_{ij} pueden ser números complejos, que se encuentran sujetos a la condición $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. Siguiendo un patrón similar, se pueden definir

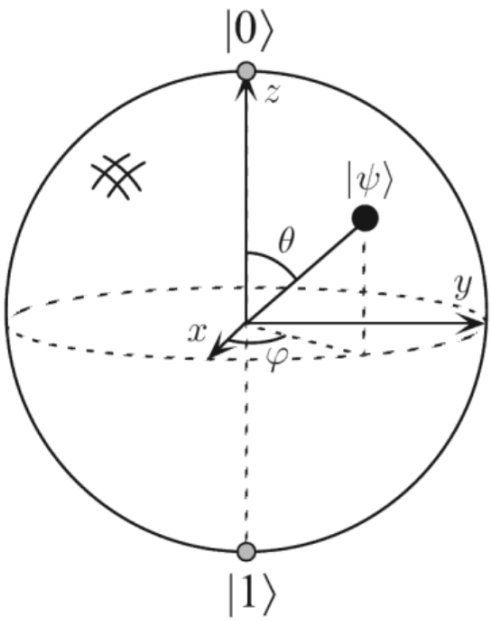

Figura 2-2: Representación de un estado cuántico $|\psi\rangle$ mediante la esfera de Bloch. Tomado de [64].

estados para cantidades arbitrarias de qubits, usando como vectores base todas las posibles combinaciones de ceros y unos. De lo anterior se puede inferir que un estado cuántico de n qubits es capaz de almacenar 2^n coeficientes cuando presenta superposición [20]. Esta capacidad exponencial de los qubits es una de las mayores ventajas que presenta la computación cuántica respecto de la computación clásica.

2.2.2. Medición Cuántica y Regla de Born

En un computador clásico, cuando se desea conocer el estado de un bit específico, basta con consultar la posición en memoria en la que éste se encuentra. Esta lectura del bit es un proceso determinista que no altera su contenido. Por otra parte, si se desea conocer el estado de un qubit, las reglas cuánticas restringen la información que puede obtenerse [36]. Dado un qubit en el estado cuántico $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, el resultado de su observación (o "medición cuántica") puede ser o bien el estado cero con probabilidad $|\alpha|^2$, o bien el estado uno con probabilidad $|\beta|^2$. No es posible obtener otro resultado, ya que estas dos probabilidades siempre suman 1. Además, toda medición posterior del mismo qubit arrojará siempre un resultado igual al de la primera medición realizada, lo que implica que el estado original $|\phi\rangle$ es efectivamente borrado en el proceso de medición [49]. Este comportamiento inherentemente estocástico y destructivo de la medición cuántica, aunque respaldado por la experimentación, no está aún del todo entendido, y sus implicaciones conforman una de las preguntas abiertas más importantes de la física cuántica [51, 12].

La generalización del concepto de medición cuántica previamente presentado se conoce como "Regla de Born". Las reglas cuánticas permiten que un qubit se pueda medir no sólo en términos de los estados base $|0\rangle$ y $|1\rangle$, sino que pueden definirse cualesquiera estados base para la medición, siempre que éstos sean estados válidos (es decir, puntos dentro de la esfera de Bloch), y sean ortogonales [64]. De forma general, la Regla de Born se puede expresar de la siguiente manera [50]. Sea A un sistema cuántico discreto tal que sus estados puedan expresarse en términos de estados base ortogonales $|e_i\rangle$, y en el que los posibles resultados de la medición sobre A, denotados como λ_i, están asociados uno a uno con estos estados base. Supóngase que dicho sistema se encuentra en un estado cuántico $|\phi\rangle$ que se puede expresar como superposición de los estados base. Entonces, la probabilidad de que el resultado de la medición de A sea λ_i está dado por $|\langle e_i|\phi\rangle|^2$, siendo $\langle e_i|\phi\rangle$ el producto interno de los dos estados. En términos de qubits, la Regla de Born señala que si un qubit se encuentra en un estado cuántico $|\phi\rangle$, y uno de los estados base de la medición es el estado cuántico $|\psi\rangle$, la probabilidad de medir el qubit en el estado $|\psi\rangle$ está dada por $|\langle\psi|\phi\rangle|^2$. El producto interno de $|\phi\rangle$ y $|\psi\rangle$ es equivalente al producto punto de sus vectores asociados si todos los valores en éstos son reales; es decir, si $\psi = [\psi_1, \psi_2]^T$ y $\phi = [\phi_1, \phi_2]^T$, con $\psi_1, \psi_2, \phi_1, \phi_2 \in \mathbb{R}$, entonces $\langle\psi|\phi\rangle = \psi^T\phi = \phi^T\psi$.

2.2.3. Matrices de Densidad

Todo estado cuántico que puede expresarse como un único vector se denomina un "estado puro". Aunque los estados puros permiten modelar una gran variedad de estados cuánticos, existen algunos escenarios en los que un estado simplemente no puede ser expresado como un único vector de estado cuántico [34]. Uno de estos escenarios está relacionado con la incertidumbre clásica, es decir, se presenta cuando no se tiene certeza de en qué estado se encuentra un qubit. Por ejemplo, si de un qubit se sabe que su estado puede ser $|0\rangle$ con probabilidad $\frac{1}{3}$ o que puede ser $|1\rangle$ con probabilidad $\frac{2}{3}$, este estado presenta incertidumbre clásica y se denomina "estado mixto". En este caso, la probabilidad del qubit de estar en un estado u otro no guarda relación con la Regla de Born o el proceso de medición cuántica, sino con el hecho de que se desconoce la preparación del estado del qubit.

El otro escenario donde los estados cuánticos no son suficientes está relacionado con un fenómeno llamado "entrelazamiento" [32]. Sea un sistema formado por dos qubits cuyo estado se puede describir de la forma $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$; de aquí se puede inferir que el primer qubit está en el estado $|0\rangle$, y el segundo qubit está en el estado $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Sin embargo, si el sistema presenta el estado $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, ya no es posible establecer cuáles son los estados individuales de los qubits. En este caso, se dice que los qubits están entrelazados, ya que una variación en un qubit lleva a una variación inmediata en el otro [43]; por ejemplo, si uno de los dos qubits es medido y su nuevo estado es $|1\rangle$, el estado total del sistema pasa a ser $|11\rangle$, y por ende el estado del primer qubit también pasa a ser $|1\rangle$, aunque no haya

sido medido. Este fenómeno, netamente cuántico, es otra de las ventajas que los algoritmos cuánticos suelen utilizar a su favor.

Las matrices de densidad sirven como representaciones más generales de los estados cuánticos, que permiten tomar en cuenta los dos escenarios discutidos previamente [31]. Un estado puro $|\phi\rangle$ puede ser descrito por la matriz de densidad dada por $\rho = |\phi\rangle\langle\phi| = \phi\phi^T$. Una combinación probabilística de estados puros (es decir, un estado mixto) puede representarse mediante la suma de las matrices asociadas a estos estados, ponderadas mediante sus respectivas probabilidades. La matriz de densidad de un sistema de varios qubits puede ser reducida (mediante una traza parcial) para modelar el estado de los qubits individuales. Además, la medición cuántica sobre matrices de densidad se puede aplicar mediante una variación de la Regla de Born [13]: si el estado de un sistema se puede describir mediante la matriz de densidad ρ, entonces la probabilidad de medir el sistema en un estado dado $|\pi\rangle$ está dada por la traza de $\rho|\pi\rangle\langle\pi|$, que es equivalente a $\langle\pi|\rho|\pi\rangle = \pi^T\rho\,\pi$. Desde su formulación original, se sabe que las matrices de densidad poseen cierta equivalencia con las distribuciones de probabilidad en escenarios clásicos, por lo que pueden usarse de forma similar [90]. Esta relación entre matrices de densidad y funciones de densidad de probabilidad forma parte esencial del método que se presenta en el próximo capítulo.

3 Anomaly Detection through Density Matrices and Fourier Features

En este Capítulo se presenta una explicación detallada del método propuesto, llamado *Anomaly Detection through Density Matrices and Fourier Features* (AD-DMKDE), describiendo su arquitectura (sus componentes básicos y las fases en que se divide) y su formulación matemática, y presentando un análisis de complejidad computacional que muestra las ventajas que este algoritmo puede presentar con respecto a KDE.

3.1. Introducción a AD-DMKDE

La principal base metodológica del algoritmo propuesto es DMKDE [30], un método orientado hacia estimación de densidad que utiliza los conceptos cuánticos explorados en el capítulo previo; estos elementos se combinan con Random Fourier Features (RFF) [71], una transformación que envía los puntos de datos hacia un espacio de alta dimensionalidad con propiedades matemáticas que facilitan el análisis, para construir una aproximación eficiente a KDE que además es susceptible de incluir redes neuronales dentro de su estructura. Sobre esta propuesta, AD-DMKDE establece una serie de mejoras, incluyendo una variación sobre la transformación de Random Fourier Features que utiliza una red neuronal para optimizar sus parámetros; este proceso de mejora, denominado Adaptive Fourier Features (AFF), se planteó originalmente en [26], y sus pormenores se detallan en la sección 3.1.

Tras la aplicación de este mapeo mejorado sobre los puntos de datos, AD-DMKDE realiza la construcción de la matriz de densidad de tal forma que ésta incluya información de todos los puntos (sección 3.2). La densidad de nuevos puntos se calcula a partir de la aplicación de la función de mapeo y del uso de la medición cuántica, obteniendo así un valor estimado que puede usarse como un *score* de la normalidad de los datos (sección 3.3). Finalmente, para establecer qué puntos se etiquetan como anómalos, se define un umbral de clasificación que sirve como discriminante, y cuyo valor se obtiene tras un proceso de validación cruzada (sección 3.4).

La Figura **3-1** resume gráficamente la arquitectura del algoritmo. Las tres etapas del algoritmo (entrenamiento, validación y prueba) corresponden a una convención comúnmente utilizada en algoritmos de aprendizaje automático, en los que el conjunto de datos bajo

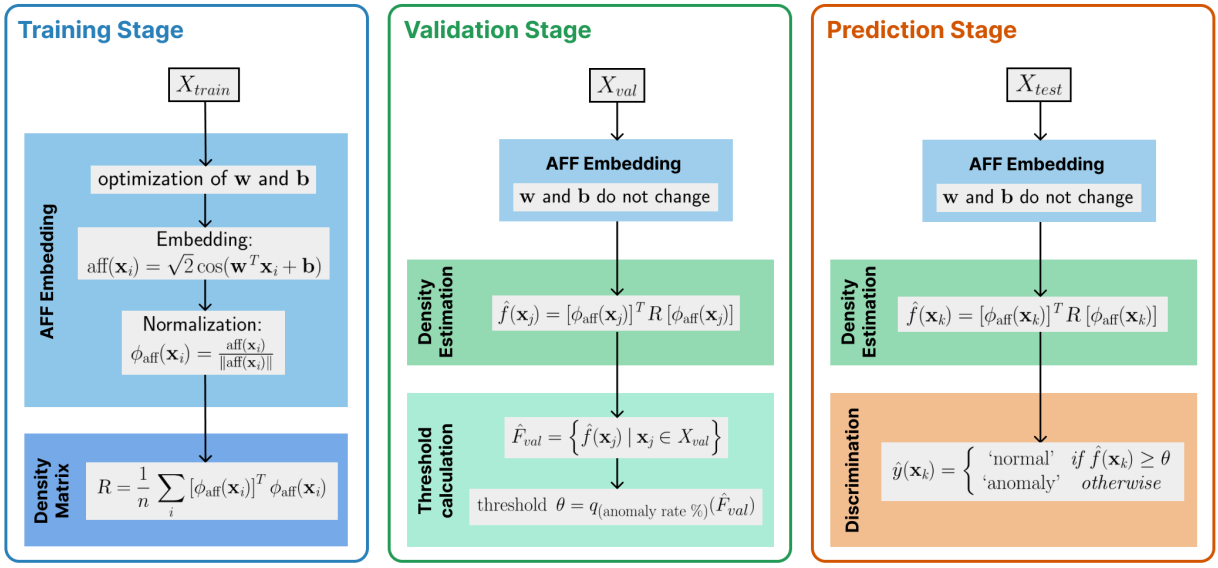

Figura 3-1: Arquitectura de AD-DMKDE. Cada sección corresponde a una etapa del algoritmo. Los parámetros optimizados de la función de mapeo y los valores en la matriz de densidad no varían una vez han sido calculados.

análisis es dividido (previamente a la aplicación del algoritmo) en tres particiones, denominadas también como entrenamiento, validación y prueba. Las primeras dos particiones se utilizan para probar distintas configuraciones del algoritmo y determinar cuál es la más adecuada para el problema, utilizando la primera como insumo del aprendizaje y evaluando el rendimiento sobre la segunda; una vez determinada la mejor configuración, el algoritmo es aplicado sobre la tercera partición para probar la capacidad de generalización del algoritmo ante nuevos datos.

3.2. Random Fourier Features y Adaptive Fourier Features

Sea $\mathbf{x}_i$ un punto de datos dentro de la partición de entrenamiento. La función de mapeo definida por las Random Fourier Features, de acuerdo con su definición original en [71], se puede expresar de la forma:

$$rff\,(\mathbf{x}_i) = \sqrt{2}\cos(\mathbf{w}^T\mathbf{x}_i + \mathbf{b})$$

donde $\mathbf{w}$ es una matriz cuyos valores son muestras aleatorias de una distribución normal estandarizada, y $\mathbf{b}$ es un vector cuyos valores son muestras aleatorias de una distribución uniforme entre 0 y 2π. Esta transformación define un espacio explícito de características para los puntos de datos, de tal forma que las dimensiones de este nuevo espacio están dadas por

los tamaños de $\mathbf{w}$ y $\mathbf{b}$. El valor de dimensionalidad del espacio de características se puede considerar como un hiperparámetro del algoritmo.

Sea k un kernel gaussiano dado por $k(\mathbf{x}-\mathbf{y}) = -\exp\left(-\gamma \left\|\mathbf{x}-\mathbf{y}\right\|^2\right)$, donde γ es un parámetro que cumple un rol similar al *bandwidth* discutido en la sección 2.1.2. De acuerdo con un resultado teórico conocido como Teorema de Bochner [60], este kernel es aproximado por el valor esperado del producto punto (*inner product*) de las transformaciones asociadas a los vectores $\mathbf{x}$ y $\mathbf{y}$; esta aproximación será más precisa entre mayor sea la dimensión del nuevo espacio. De esta manera, el espacio construido por la transformación RFF permite calcular los valores del kernel de forma relativamente sencilla. Dada la estrecha relación entre el kernel k y la transformación RFF, el valor γ es uno de los elementos más influyentes sobre la selección de los valores en $\mathbf{w}$ y $\mathbf{b}$, y se puede ver como otro hiperparámetro del algoritmo.

Al estar formados por muestras aleatorias, $\mathbf{w}$ y $\mathbf{b}$ (y por ende la eficacia de la función de mapeo $\mathit{rff}\,(\mathbf{x})$ para aproximar el kernel k) son altamente influenciados por las semillas aleatorias, lo que puede repercutir en la estabilidad y reproducibilidad del algoritmo entero. Por este motivo, AD-DMKDE plantea la optimizción de $\mathbf{w}$ y $\mathbf{b}$ a través del proceso Adaptive Fourier Features (AFF) [26]. Este proceso toma como entradas a parejas de elementos del conjunto de entrenamiento, seleccionadas aleatoriamente, y a un valor inicial de $\mathbf{w}$ y $\mathbf{b}$. Las parejas de datos son usadas para entrenar una red neuronal mediante el mecanismo de aprendizaje por gradiente descendente; la red busca minimizar una función de error dada por el cuadrado de la diferencia entre el producto punto de los elementos de las parejas y el kernel calculado sobre ellas, alterando poco a poco los valores en $\mathbf{w}$ y $\mathbf{b}$. Así, el resultado del proceso AFF es una nueva función de mapeo $\mathit{aff}\,(\mathbf{x})$ similar a la de RFF, en la que sus parámetros optimizados garantizan que ésta es más eficiente a la hora de aproximar el kernel.

3.3. Construcción de la Matriz de Densidad

Una vez se define la función de mapeo optimizada tras aplicar el proceso AFF, cada punto de datos $\mathbf{x}_i$ en el conjunto de entrenamiento es transformado a su versión en el espacio de características dada por $\mathit{aff}(\mathbf{x}_i)$. Estos puntos son luego normalizados, es decir, multiplicados por el inverso de su norma; con ello se puede obtener una representación de los datos que tenga mayor similitud con los estados cuánticos, los cuales, de acuerdo con lo presentado en la sección 2.2, están asociados a vectores columna normalizados. Esta representación, denotada como $\phi_{\mathit{aff}}(\mathbf{x}_i)$, está dada por:

$$\phi_{\mathit{aff}}(\mathbf{x}_i) := \frac{\mathit{aff}\,(\mathbf{x}_i)}{\left\|\mathit{aff}\,(\mathbf{x}_i)\right\|}$$

Los $\phi_{\mathit{aff}}(\mathbf{x}_i)$, aunque representados como vectores fila, pueden equipararse a estados cuánticos, permitiendo el uso de las matrices de densidad. Para ello, AD-DMKDE construye un

estado mixto que incluye a todos los "estados" asociados a los puntos de datos de entrenamiento, bajo el supuesto de que todos ellos son igualmente probables. Luego, el método calcula la matriz de densidad R asociada a dicho estado mixto:

$$R = \frac{1}{n} \sum_{i=1}^{n} \left(\phi_{\mathit{aff}}(\mathbf{x}_i)\right)^T \phi_{\mathit{aff}}(\mathbf{x}_i)$$

donde n es la cantidad de datos de entrenamiento. Así, R contiene, de forma compacta, la información de los "estados" de todos estos puntos de datos. Es importante mencionar que el tamaño de esta matriz no depende de n, sino de la dimensionalidad del espacio definido por la función de mapeo. Otra ventaja notable en el uso de las matrices de densidad como esquema de almacenamiento está ligada al hecho de que el cálculo de R sólo ocurre una única vez en todo el proceso de entrenamiento, lo que contrasta con el uso intensivo de los datos en varios métodos basados en aprendizaje.

AD-DMKDE permite, como una variación del algoritmo, construir la descomposición espectral de R, lo que permite expresar la matriz de la forma $R = V^T \Lambda V$, donde V contiene los eigenvectores de la matriz R, y Λ es una matriz cuadrada cuya diagonal contiene los respectivos eigenvalores de la matriz R. Muchos de estos eigenvalores son cercanos a cero, por lo que es posible definir un número máximo de eigenvalores a considerar, y utilizar una aproximación de gradiente descendente (SGD) para encontrar los valores de éstos que mejor aproximen a R. La aplicación de esta variante de AD-DMKDE reduce la cantidad de memoria requerida por R, a costa de una pequeña pérdida de precisión.

3.4. Estimación de Densidad mediante Medición Cuántica

Una vez se ha calculado R a partir de los datos de entrenamiento, es posible utilizarla para estimar la densidad de nuevos puntos de datos. Sea $\mathbf{x}_j$ un punto de datos en la partición de validación. Sobre este punto se aplica el mapeo optimizado $\mathit{aff}(\mathbf{x}_j)$ (exactamente con los mismos parámetros obtenidos en el proceso AFF) y luego la normalización, para construir $\phi_{\mathit{aff}}(\mathbf{x}_j)$. Ya que esta representación es similar a un estado cuántico, esto permite aplicar un mecanismo de medición cuántica equivalente a la versión de la regla de Born para matrices de densidad discutida en la subsección 2.3.3. De esta forma, la densidad estimada de $\mathbf{x}_j$, denotada por $\hat{f}(\mathbf{x}_j)$, estará dada por:

$$\hat{f}(\mathbf{x}_j) = \left(\phi_{\mathit{aff}}(\mathbf{x}_j)\right)^T \; R \; \left(\phi_{\mathit{aff}}(\mathbf{x}_j)\right)$$

Este proceso de transformación y cálculo de la densidad estimada se realiza para todos los puntos de datos en el conjunto de validación, lo que permite construir un conjunto $\hat{F}_{val}$ que contiene todos los valores de densidad obtenidos para los datos de validación. Este conjunto es luego ordenado de forma ascendente.

3.5. Cálculo del umbral y Predicción

Para determinar la frontera entre los datos normales y anómalos, AD-DMKDE utiliza el conjunto ordenado de estimaciones de densidad de los datos de validación, $\hat{F}_{val}$. Dada la tasa de anomalías en el conjunto de datos bajo análisis, denotada r (la cual puede ser un valor conocido a priori, una estimación razonable para el conjunto de datos, o la proporción de datos anómalos que se desea encontrar si no hay mayor información), se calcula el percentil r-ésimo sobre $\hat{F}_{val}$, es decir, el valor de densidad tal que el $r\,\%$ de las densidades estimadas en $\hat{F}_{val}$ quedan por debajo de dicho valor. Este percentil, denominado en adelante como θ, es el valor "umbral" que servirá como discriminante para determinar qué valores de densidad son lo suficientemente bajos como para ser considerados anómalos.

Así, la última etapa de AD-DMKDE utiliza los datos de prueba de la siguiente manera. Sea $\mathbf{x}_k$ un punto de datos en la partición de prueba. Este dato es transformado (usando la misma función de mapeo del proceso AFF) y luego normalizado para obtener su respectiva representación de estado cuántico $\phi_{aff}(\mathbf{x}_k)$. Usando el mecanismo de medición cuántica previamente definido, $\phi_{aff}(\mathbf{x}_k)$ es operado con la matriz de densidad R para obtener la densidad estimada $\hat{f}(\mathbf{x}_k) = (\phi_{aff}(\mathbf{x}_k))^T\ R\ (\phi_{aff}(\mathbf{x}_k))$. Finalmente, el valor de densidad $\hat{f}(\mathbf{x}_k)$ es comparado contra el umbral θ para determinar la etiqueta $\hat{y}(\mathbf{x}_k)$ que el algoritmo otorgará al punto $\mathbf{x}_k$:

$$\hat{y}(\mathbf{x}_k) = \begin{cases} \textit{'normal'} & \textit{si } \hat{f}(\mathbf{x}_k) \geq \theta \\ \textit{'anómalo'} & \textit{en otro caso} \end{cases}$$

En resumen, AD-DMKDE se compone de tres etapas, centradas en las particiones de datos de entrenamiento, prueba y validación:

- Los datos de entrenamiento son utilizados para definir la transformación de los datos, iniciando con la función de mapeo de RFF, y optimizándola mediante el proceso AFF, que está basado en redes neuronales. Este proceso depende de dos hiperparámetros, la dimensión del espacio de características y el valor γ asociado al kernel gaussiano que se busca aproximar. El mapeo optimizado se complementa con la normalización, para obtener representaciones de los datos similares a estados cuánticos, y éstos son utilizados para construir una matriz de densidad que contiene, de forma compacta, toda la información del conjunto de entrenamiento. Puesto que se busca que la matriz de densidad represente la distribución de probabilidad de los datos normales, los datos anómalos no son considerados en el cálculo de ésta.

- Los datos de validación son procesados mediante los mecanismos definidos en la etapa previa, de tal forma que se convierten en representaciones de estado cuántico. Cada representación se opera con la matriz de densidad, usando una aproximación a la

Regla de Born para calcular su densidad estimada. El conjunto de densidades que se obtiene al procesar todos los datos de validación es ordenado y se utiliza un percentil para determinar el umbral de separación entre datos normales y anómalos. A través de varias pruebas sobre estos datos, es posible establecer los valores de los hiperparámetros que generan una mejor reconstrucción de la función de densidad de los datos.

- Los datos de prueba son transformados, con las mismas operaciones que en las etapas previas, en representaciones de estados cuánticos, a partir de las cuales se estiman sus densidades operando con la matriz de densidad. Los valores de densidad estimados son comparados con el valor de umbral, y de esta forma se asigna una etiqueta ('normal' o 'anómalo') a cada dato de prueba, dependiendo de si su densidad es mayor o menor a dicho umbral.

3.6. Análisis de Complejidad y Comparación con KDE

De acuerdo con la definición de KDE presentada en el capítulo anterior, este método se basa en la aplicación del estimador de Parzen-Rosenblatt:

$$\hat{f}_h(\mathbf{y}) = \frac{1}{nM_h} \sum_{i=1}^{n} k(\mathbf{x}_i - \mathbf{y}; h)$$

Para estimar la densidad de un punto $\mathbf{y}$, KDE requiere almacenar todos sus datos de entrenamiento $\mathbf{x}_i$ para calcular el kernel entre el punto de interés y cada uno de ellos. En contraste, AD-DMKDE utiliza una matriz de densidad como mecanismo de almacenamiento, cuyo tamaño depende únicamente de la dimensión del espacio de características creado por la transformación AFF. Esta matriz, que se calcula por una única vez, elimina la necesidad de almacenar los datos.

La ventaja de AD-DMKDE sobre KDE se puede expresar de forma más precisa a partir de un análisis de complejidad computacional. Sea n el número de puntos de datos de entrenamiento, m el número de puntos de datos de consulta o prueba, y d la dimensionalidad del conjunto de datos. El cálculo del kernel gaussiano es lineal en términos de la dimensión de los puntos, es decir, su complejidad es $O(d)$. KDE requiere calcular n kernels para cada punto de prueba, de modo que la estimación de un punto dado presenta complejidad $O(nd)$. Esto implica que la estimación de los m puntos de prueba tiene una complejidad de $O(mnd)$ en tiempo. En cuanto a memoria, ya que se necesita acceder a todos los puntos de entrenamiento para utilizar el estimador de Parzen-Rosenblatt, el almacenamiento requerido por KDE presenta complejidad $O(nd)$.

La complejidad de AD-DMKDE, por otra parte, puede determinarse a partir de sus etapas. Sea D el tamaño del espacio definido por la función AFF. En la fase de entrenamiento, es

necesario un cálculo proporcional a $O(dD)$ para obtener las características optimizadas del proceso AFF; además, se realiza un cálculo proporcional a $O(nD^2)$ para calcular la matriz de densidad. En las fases posteriores, el cálculo de la estimación de todos los datos de prueba muestra complejidad $O(mdD)$ para su transformación y $O(mD^2)$ para operar con la matriz de densidad. En términos de memoria, AD-DMKDE almacena las características optimizadas del proceso AFF en una matriz de tamaño $O(dD)$, y la matriz de densidad presenta un tamaño $O(D^2)$. Si se supone que $D \gg d$, la complejidad de AD-DMKDE se puede resumir como $O(D^2(n+m))$ en tiempo, y $O(D^2)$ en memoria. La ventaja de AD-DMKDE, aunque menos notable en tiempo, es bastante relevante en memoria, ya que sus requerimientos son independientes de la cantidad de datos; esto permitiría a AD-DMKDE abordar conjuntos de datos grandes que para KDE serían prohibitivos.

4 Evaluación Experimental

Con el propósito de determinar el desempeño de AD-DMKDE ante diversos escenarios (y establecer si logra alcanzar o superar a los algoritmos de detección de anomalías más reconocidos), se define una configuración experimental que consiste en la aplicación del método propuesto y de los métodos de línea base presentados en la sección 2.1.3, sobre una serie de conjuntos de datos numéricos multivariados. Para ello, se utilizan las implementaciones provistas por las siguientes librerías del lenguaje Python:

- **Scikit-Learn** [68]. Esta librería, una de las más utilizadas en las áreas de análisis de datos y aprendizaje automático, implementa los algoritmos Minimum Covariance Estimator, OneClassSVM, Isolation Forest y Local Outlier Factor.

- **PyOD** [95]. Esta librería contiene más de cuarenta algoritmos para realizar detección de anomalías en datos numéricos. Se utilizaron las implementaciones de los algoritmos KNN, SOS, COPOD, LODA, VAE-Bayes, DeepSVDD y ALAD.

- **LAKE** [56]. La implementación de este algoritmo se ha tomado del repositorio Github de sus autores[1], con algunas modificaciones concernientes a mantener la misma proporción de muestras normales y anómalas en las particiones de datos.

Para controlar los comportamientos aleatorios que AD-DMKDE y otros métodos considerados pueden presentar en algunas etapas (y garantizar la reproducibilidad de la experimentación), se seleccionó un valor constante para cada semilla aleatoria que pudiera influenciar a los métodos. Todos los experimentos se llevaron a cabo en una máquina que ejecuta el sistema operativo Ubuntu 20.04.2., y que contiene un procesador Intel Xeon de 64 núcleos a 2.1 GHz, con 128 GB de RAM y dos tarjetas de procesamiento gráfico NVIDIA RTX A5000.

4.1. Conjuntos de Datos

Como parte de la experimentación planteada, cada algoritmo considerado fue aplicado sobre veinticuatro conjuntos de datos numéricos previamente etiquetados, los cuales fueron seleccionados para presentar una amplia variedad en términos de su tamaño (cantidad de muestras), dimensionalidad (cantidad de características) y tasa de anomalías (la proporción

[1] https://github.com/1246170471/LAKE

Tabla 4-1: Características de los conjuntos de datos.

Dataset	Tamaño	Dimensiones	Tasa de Anomalías
Annthyroid	7200	6	0,0742
Arrhythmia	452	274	0,146
Breastw	683	9	0,35
Cardio	1831	21	0,096
ForestCover	286048	10	0,0096
Glass	214	9	0,042
Ionosphere	351	33	0,359
Letter	1600	32	0,0625
Mammography	11183	6	0,02325
MNIST	7603	100	0,092
Musk	3062	166	0,0317
OptDigits	5216	64	0,0288
PenDigits	6870	16	0,0227
Pima	768	8	0,349
Satellite	6435	36	0,3164
SatImage	5803	36	0,0122
Shuttle	49097	9	0,0715
SpamBase	3485	58	0,2
Speech	3686	400	0,01655
Thyroid	3772	6	0,025
Vertebral	240	6	0,125
Vowels	1456	12	0,03434
WBC	378	30	0,0556
Wine	129	13	0,07752

de datos etiquetados como anómalos). Todos los conjuntos de datos seleccionados provienen del repositorio ODDS [74] de la Universidad Stony Brook[2], el cual está especializado en detección de anomalías y outliers. Un resumen de las principales características de los conjuntos de datos puede observarse en la Tabla **4-1**.

A continuación, se presenta, para cada conjunto de datos, una breve descripción.

- Annthyroid: proveniente de un dataset con tres clases original del Repositorio de Machine Learning de UCI, está relacionado con el diagnóstico de enfermedad de tiroides. Los datos de pacientes con hipertiroidismo o con hipotiroidismo son etiquetadas como anomalías, y los pacientes sanos como datos normales.
- Arrhythmia: establecido originalmente como un dataset para clasificación con múltiples etiquetas asociadas a enfermedad coronaria, ha sido modificado de tal forma que las ocho clases más pequeñas se etiquetan como anómalas, y las restantes como normales.
- Breastw: conformado por datos de casos de cáncer de seno en un hospital de Wisconsin, USA, los datos anómalos corresponden a pacientes con diagnósticos de tumor maligno.
- Cardio: este dataset contiene información sobre mediciones cardiacas obstétricas. Originalmente con tres clases (correspondientes a pacientes sanos, sospechosos y enfermos),

[2]https://odds.cs.stonybrook.edu

los datos de pacientes sospechosos han sido eliminados, y los pacientes enfermos se han etiquetado como anomalías.

- ForestCover: conformado por datos de campo de pequeñas zonas de bosque en una reserva forestal de Colorado, USA, sus 54 variables originales han sido reducidas a diez valores numéricos. Las clases originales están relacionadas con el tipo de cobertura vegetal; de estas, la clase 4 se considera normal, y la clase 2 anómala.

- Glass: este dataset, construido originalmente para investigación criminalística, contiene información acerca de seis tipos diferentes de vidrio, en términos de su contenido de distintos óxidos metálicos. El tipo 6 se considera como anómalo y los restantes como normales.

- Ionosphere: el dataset consta de mediciones de radar realizadas desde tierra y dirigidas a capas altas de la atmósfera. Las mediciones son de dos tipos, dependiendo de si se detectan estructuras atmosféricas o no; los datos sin detección se consideran anómalos.

- Letter: la versión original de los datos (provenientes del Repositorio de Machine Learning de UCI) corresponde a arreglos de tamaño 4×4 de pixeles que representan letras. La versión de ODDS combina las letras en pares para obtener arreglos de tamaño 32, y elige tres letras como valores normales; los datos anómalos son los pares donde una letra no pertenece a las tres elegidas como normales.

- Mammography: este dataset fue originalmente planteado para detectar zonas con calcificación en imágenes diagnósticas de seno (mamografías); los datos asociados a imágenes con calcificaciones se etiquetan como anomalías.

- MNIST: este es uno de los datasets más conocidos para tareas de aprendizaje automático, conformado por imágenes de dígitos de tamaño 26×26. La versión de ODDS utiliza un subconjunto de cien dimensiones que sólo contiene dígitos cero y seis; los datos de digíto cero se consideran datos normales, y los de dígito seis se consideran anómalos.

- Musk: este dataset contiene información de mediciones realizadas sobre varias configuraciones de moléculas de las que se desea saber si presentan aroma o no. Las moléculas con aroma son menos frecuentes, por lo que se etiquetan como datos anómalos.

- OptDigits: el dataset consta de dígitos escritos a mano, representados mediante una matriz de tamaño 8×8, donde cada valor cuenta los píxeles encendidos en un área pequeña de la imagen original del dígito. Para este experimento, los datos asociados a dígitos cero son los anómalos, y los dígitos restantes conforman los datos normales.

- PenDigits: en este dataset también se hace referencia a dígitos escritos a mano, pero representados aquí como series de ocho pares de coordenadas (x, y) por las que pasa el

trazo del dígito. Los datos asociados al dígito cero son considerados como anómalos, y los datos de los demás dígitos como normales.

- Pima: conformado por datos médicos de mujeres jóvenes, en este dataset los datos anómalos corresponden a pacientes diagnosticadas con diabetes.

- Satellite: tomado del dataset Statlog de imágenes satelitales de Landsat, este dataset se compone de los valores de cuatro canales de color para pequeñas secciones de 3×3 pixeles, y las clases representan diferentes tipos de suelo. La versión de ODDS etiqueta como anómalas a las tres clases más pequeñas (las clases '2', '4' y '5').

- SatImage: con un origen similar al conjunto anterior, en este caso la clase '2' se reduce en cantidad y se etiqueta como anómala, mientras todas las demás clases se etiquetan como datos normales.

- Shuttle: el dataset, originario del Repositorio de Machine Learning de UCI, está relacionado con el posicionamiento de transbordadores espaciales, y consta de nueve dimensiones y siete clases; en este caso, la clase '4' es eliminada, la clase '1' es etiquetada como datos normales, y las clases restantes se etiquetan como anomalías.

- SpamBase: este dataset, tomado del Repositorio de Machine Learning de UCI, contiene información sobre la frecuencia de algunas palabras y caracteres en correos electrónicos, así como sobre la ocurrencia de letras mayúsculas; los datos anómalos corresponden a los correos identificados como spam (mensajes no deseados).

- Speech: este dataset está conformado por una serie de vectores que representan segmentos de grabaciones de palabras en inglés, grabadas con diferentes acentos, y cada acento corresponde a una clase distinta. En este experimento, los datos normales corresponden al acento americano, y los datos anómalos a los demás acentos.

- Thyroid: proveniente del mismo conjunto de datos que Annthyroid, en este caso se seleccionan 3772 muestras del total, y se etiqueta como anomalías únicamente a los datos de la clase hipertiroidismo, siendo los datos normales los de las otras dos clases.

- Vertebral: este dataset contiene características biomecánicas de 240 pacientes, relacionadas con la pelvis y el área lumbar. Los pacientes sin afectaciones son minoría, por lo que en este caso se toman como anómalos.

- Vowels: originalmente del Repositorio de Machine Learning de UCI, cada dato representa una serie de tiempo correspondiente a la grabación de una vocal japonesa. Cada clase representa a un hablante distinto; la versión de ODDS elige sólo los datos de cuatro de ellos, y selecciona al primer hablante como anómalo.

- WBC: basado en una serie de imágenes diagnósticas sobre biopsias de tejido mamario, de estas imágenes se extraen las características de los núcleos de las células tumorales. Los datos de tumores malignos se consideran anomalías, y los datos de tumores benignos se consideran normales.
- Wine: las muestras en este dataset corresponden al resultado del análisis químico de varios vinos producidos en tres viñedos italianos diferentes. Cada clase hace referencia a un viñedo; la versión de ODDS elige unas pocas muestras de la clase '1' como datos anómalos, y las clases '2' y '3' como datos normales.

4.2. Configuración Experimental

Cada algoritmo considerado depende de uno o varios hiperparámetros ajustables, que influyen profundamente en el desempeño y rapidez ante una tarea dada. Existen valores de los hiperparámetros que pueden hacer que su respectivo método presente el mejor desempeño posible dadas sus limitaciones; estos "valores óptimos" suelen depender de las características del conjunto de datos, y encontrarlos requiere de probar el algoritmo muchas veces con diferentes combinaciones de valores. Esta búsqueda de los mejores hiperparámetros, conocida en inglés como *grid search*, fue planteada para cada algoritmo y para cada conjunto de datos. Al trabajar con la implementación de los algoritmos de línea base, se eligió mantener las configuraciones por defecto (es decir, fijar la mayoría de sus hiperparámetros a los valores recomendados por la documentación de la librería fuente o por el artículo original del respectivo algoritmo), dejando sólo unos pocos valores abiertos a modificación.

Los hiperparámetros sujetos a la búsqueda fueron los siguientes:

- Todos los algoritmos, con excepción de LAKE y AD-DMKDE, fueron probados con varios valores de contaminación (nu), un hiperparámetro que hace referencia a la proporción de anomalías en la partición de entrenamiento.
- LOF y KNN, al ser métodos que dependen de medidas locales de distancia a un número dado de puntos vecinos, fueron probados con varios valores para el número k de vecinos a considerar.
- OneClassSVM y AD-DMKDE, al depender de la formulación previa de un kernel gaussiano, requieren del hiperparámetro *gamma* (γ) relacionado con la geometría de dicho kernel, para el cual se probaron varios valores en una escala logarítmica.
- Otros hiperparámetros para los cuales se buscaron valores óptimos son: en Isolation Forest, la cantidad de árboles de decisión y el tamaño de las muestras con las que trabajan; en SOS, el valor *perplexity* involucrado en el cálculo de la afinidad; en ALAD, si la función de pérdida depende o no del error de reconstrucción; en LAKE, el tamaño del

batch y la arquitectura de la red neuronal que codifica los datos; y en AD-DMKDE, el tamaño de la codificación generada por el proceso AFF, también denominada "número de componentes".

Todos los conjuntos de datos fueron particionados de forma estratificada (es decir, conservando en cada partición la misma proporción de anomalías), teniendo en cuenta la siguiente regla: un 25 % de los datos son elegidos aleatoriamente en la partición de prueba, y del 75 % restante, nuevamente se elige un 25 % de los datos como partición de validación, y los restantes como partición de entrenamiento. Esto deja poco más del 56 % de los datos para que los métodos aprendan. La separación de los datos fue realizada una única vez por cada dataset, de modo que todos los métodos trabajaron con exactamente las mismas particiones.

La medición del desempeño de los algoritmos se realizó mediante la aplicación de dos métricas diferentes sobre los resultados de los modelos al aplicarse sobre la partición de prueba. Estas métricas, elegidas por su amplio uso en escenarios supervisados de detección de anomalías y por su robustez para evaluar métodos en conjuntos de datos altamente desbalanceados [4], como los listados en la sección anterior, son las siguientes:

- **Área bajo la curva ROC (AUC-ROC).** La curva ROC (*Receiver Operating Characteristic*) contrasta gráficamente la sensibilidad del modelo (la proporción de datos normales correctamente clasificados) con su especificidad (la proporción de datos anómalos correctamente clasificados) para diferentes valores del umbral de separación. El área bajo dicha curva se puede interpretar como la probabilidad de que un dato anómalo elegido al azar muestre un *score* de anomalía mayor que el de un dato normal elegido al azar [21].
- **Área bajo la curva Precision-Recall (AUC-PR).** La curva Precision-Recall contrasta la precisión del modelo en la clase anómala (la proporción de predicciones de anomalías que corresponden a verdaderas anomalías) con el recall de la clase anómala (equivalente a la especificidad) para distintos valores del umbral de separación. Un área bajo esta curva cercana a 1 sugiere que el modelo tiende a asignar las etiquetas anómalas a los datos realmente anómalos, con pocos errores.

El cálculo de estas dos métricas se efectuó mediante las implementaciones provistas en la librería Scikit-Learn. Los hiperparámetros elegidos como óptimos en cada caso fueron aquellos que maximizaron el valor del AUC-ROC, y en caso de empate, los que maximizaron también el AUC-PR.

5 Resultados

En este capítulo, se presentan los resultados de la evaluación experimental discutida en el capítulo anterior. En particular, se reportan los hiperparámetros óptimos y las métricas de desempeño obtenidas tras realizar la aplicación de los métodos de línea base y AD-DMKDE sobre los conjuntos de datos reseñados en la sección 4.1. Sobre estos resultados se realiza una revisión detallada para identificar los escenarios en los cuales AD-DMKDE presenta un desempeño sobresaliente o un desempeño bajo, con el ánimo de establecer las fortalezas y debilidades del método. Además, se presentan los resultados de un análisis estadístico sobre los resultados, para determinar si el método se diferencia significativamente de la línea base.

5.1. Discusión de Resultados

En la tabla **5-1**, se presentan los valores óptimos encontrados al realizar la búsqueda de hiperparámetros para cada algoritmo al ser aplicado sobre cada uno de los conjuntos de datos. Los hiperparámetros considerados en la tabla hacen referencia a aquellos indicados en la sección 4.2; los nombres con los que se identifican en la tabla son los mismos con los que fueron denominados al realizar la implementación en código de cada método. En particular, el parámetro *nu* indica la proporción de anomalías que, de acuerdo con cada método, representa mejor a cada conjunto de datos. La ausencia del valor que corresponde al método SOS aplicado sobre ForestCover se debe a que dicho método (que requiere de la construcción de una matriz de afinidades entre todos los puntos de datos) presentó un consumo de memoria RAM imposible de abordar, dado el tamaño de ese conjunto de datos en particular.

Al analizar los mejores hiperparámetros presentados en la Tabla **5-1**, se puede observar, por ejemplo, que los métodos que dependen de *nu* obtienen, en su mayoría, valores relativamente similares del hiperparámetro al ser aplicados sobre un conjunto de datos en particular. Esto se puede notar especialmente en conjuntos de datos con tasas de anomalías muy bajas, como OptDigits, SatImage o Speech. Los conjuntos con los valores de *nu* más dispersos incluyen a Arrhythmia, Pima y Wine, todos ellos con tasas de anomalías más bien altas. Estos valores obtenidos para *nu* no suelen alejarse mucho de la proporción real de anomalías presentada en la Tabla **4-1**, excepto en algunos casos puntuales como ALAD sobre Vowels, LOF sobre Cardio o LODA sobre Glass. Otros hiperparámetros compartidos por varios métodos, como el número de vecinos en LOF y KNN o el *gamma* en OneClassSVM y AD-DMKDE, presentan una variabilidad mucho más alta.

Tabla 5-1: Mejores parámetros obtenidos en la configuración experimental, para cada algoritmo aplicado sobre cada conjunto de datos.

	OneClassSVM	Isolation Forest	Covariance	LOF	KNN	SOS
DATASET	[gamma, nu]	[n_estimators, max_samples, nu]	[nu]	[n_neighbors, nu]	[n_neighbors, nu]	[perplexity, nu]
Annthyroid	[0,25 0,02]	[20 40 0,04]	[0,05]	[48 0,03]	[60 0,04]	[10,0 0,02]
Arrhythmia	[0,03125 0,01]	[100 40 0,08]	[0,15]	[16 0,07]	[80 0,12]	[30,0 0,12]
Breastw	[4,0 0,29]	[80 40 0,35]	[0,40]	[2 0,36]	[50 0,36]	[100,0 0,40]
Cardio	[0,0039 0,10]	[100 100 0,15]	[0,09]	[10 0,02]	[100 0,13]	[100,0 0,06]
ForestCover	[0,03125 0,01]	[20 40 0,01]	[0,01]	[48 0,01]	[90 0,01]	-
Glass	[0,000977 0,01]	[100 40 0,01]	[0,02]	[2 0,01]	[10 0,01]	[10,0 0,02]
Ionosphere	[0,5 0,33]	[20 100 0,33]	[0,33]	[6 0,39]	[10 0,40]	[10,0 0,29]
Letter	[0,000977 0,01]	[100 100 0,01]	[0,02]	[6 0,03]	[10 0,03]	[20,0 0,05]
Mammography	[0,25 0,02]	[20 20 0,01]	[0,01]	[20 0,01]	[40 0,02]	[20,0 0,01]
MNIST	[0,125 0,03]	[20 100 0,07]	[0,12]	[50 0,04]	[50 0,07]	[100,0 0,02]
Musk	[0,000977 0,01]	[60 80 0,04]	[0,03]	[20 0,02]	[50 0,03]	[100,0 0,03]
OptDigits	[0,000977 0,01]	[40 20 0,01]	[0,01]	[10 0,01]	[40 0,01]	[60,0 0,01]
PenDigits	[0,125 0,01]	[60 60 0,01]	[0,01]	[44 0,01]	[80 0,04]	[90,0 0,01]
Pima	[0,0625 0,28]	[60 60 0,28]	[0,33]	[48 0,30]	[60 0,36]	[40,0 0,25]
Satellite	[0,5 0,28]	[20 20 0,25]	[0,26]	[24 0,27]	[100 0,29]	[100,0 0,35]
SatImage	[0,00195 0,02]	[100 100 0,01]	[0,02]	[6 0,01]	[90 0,01]	[100,0 0,01]
Shuttle	[0,0078 0,13]	[40 80 0,07]	[0,09]	[42 0,01]	[50 0,01]	[100,0 0,01]
SpamBase	[0,015625 0,17]	[100 20 0,19]	[0,19]	[40 0,17]	[10 0,22]	[90,0 0,17]
Speech	[0,000977 0,01]	[20 40 0,01]	[0,01]	[2 0,01]	[30 0,02]	[90,0 0,01]
Thyroid	[4,0 0,01]	[40 40 0,02]	[0,02]	[50 0,04]	[40 0,02]	[30,0 0,02]
Vertebral	[0,000977 0,01]	[60 100 0,01]	[0,01]	[8 0,12]	[10 0,12]	[20,0 0,05]
Vowels	[0,0625 0,02]	[80 40 0,01]	[0,02]	[6 0,02]	[10 0,02]	[10,0 0,01]
WBC	[0,000977 0,01]	[20 80 0,08]	[0,03]	[48 0,06]	[50 0,08]	[60,0 0,11]
Wine	[0,000977 0,01]	[20 60 0,06]	[0,05]	[24 0,15]	[10 0,13]	[100,0 0,15]

	COPOD	LODA	VAE-Bayes	DSVDD	ALAD	LAKE	AD-DMKDE
DATASET	[nu]	[nu]	[nu]	[nu]	[nu, add_recon_loss]	[batch_size, encoder_layers]	[rff_comps, gamma]
Annthyroid	[0,07]	[0,03]	[0,04]	[0,02]	[0,05 False]	[500 (45,35,30)]	[4000 32,0]
Arrhythmia	[0,14]	[0,04]	[0,12]	[0,01]	[0,09 False]	[500 (45,35,30)]	[2000 0,125]
Breastw	[0,37]	[0,33]	[0,34]	[0,34]	[0,36 False]	[1000 (45,35,30)]	[1000 1,0]
Cardio	[0,11]	[0,07]	[0,10]	[0,04]	[0,15 False]	[50 (20,15,15)]	[4000 0,001953]
ForestCover	[0,01]	[0,01]	[0,01]	[0,04]	[0,03 False]	[1000 (60,25,20)]	[4000 8,0]
Glass	[0,06]	[0,13]	[0,02]	[0,03]	[0,07 False]	[1000 (60,25,20)]	[4000 4,0]
Ionosphere	[0,35]	[0,34]	[0,34]	[0,31]	[0,30 True]	[1000 (60,25,20)]	[2000 1,5]
Letter	[0,04]	[0,04]	[0,03]	[0,03]	[0,11 True]	[1000 (45,35,30)]	[4000 2,0]
Mammography	[0,01]	[0,03]	[0,01]	[0,01]	[0,04 False]	[1000 (60,25,20)]	[2000 0,0078]
MNIST	[0,04]	[0,10]	[0,04]	[0,01]	[0,12 False]	[100 (60,25,20)]	[2000 0,5]
Musk	[0,02]	[0,02]	[0,03]	[0,03]	[0,04 False]	[500 (60,25,20)]	[2000 0,25]
OptDigits	[0,01]	[0,01]	[0,01]	[0,01]	[0,04 False]	[1000 (60,25,20)]	[1000 0,5]
PenDigits	[0,03]	[0,01]	[0,01]	[0,04]	[0,05 True]	[200 (20,15,15)]	[2000 2,0]
Pima	[0,37]	[0,27]	[0,39]	[0,29]	[0,37 True]	[200 (45,35,30)]	[4000 4,0]
Satellite	[0,25]	[0,27]	[0,29]	[0,39]	[0,30 True]	[100 (60,25,20)]	[1000 2,0]
SatImage	[0,01]	[0,01]	[0,01]	[0,01]	[0,01 True]	[200 (20,15,15)]	[2000 0,25]
Shuttle	[0,07]	[0,08]	[0,07]	[0,08]	[0,07 True]	[200 (20,15,15)]	[1000 8,0]
SpamBase	[0,21]	[0,17]	[0,16]	[0,22]	[0,24 False]	[100 (60,25,20)]	[4000 4,0]
Speech	[0,01]	[0,01]	[0,01]	[0,01]	[0,05 True]	[500 (60,25,20)]	[1000 64,0]
Thyroid	[0,01]	[0,01]	[0,02]	[0,03]	[0,02 True]	[500 (20,15,15)]	[1000 16,0]
Vertebral	[0,09]	[0,10]	[0,06]	[0,01]	[0,05 True]	[1000 (60,25,20)]	[1000 128,0]
Vowels	[0,01]	[0,02]	[0,02]	[0,01]	[0,14 False]	[1000 (45,35,30)]	[2000 4,0]
WBC	[0,09]	[0,11]	[0,08]	[0,07]	[0,02 False]	[1000 (60,25,20)]	[2000 0,015625]
Wine	[0,15]	[0,15]	[0,06]	[0,03]	[0,13 False]	[1000 (60,25,20)]	[4000 2,0]

Por otra parte, las tablas **5-2** y **5-3** presentan las métricas de desempeño previamente seleccionadas, que se calcularon al aplicar los algoritmos con los hiperparámetros óptimos sobre las particiones de prueba. En ambas tablas, el mejor resultado está resaltado en negrilla y el segundo mejor resultado está subrayado. AD-DMKDE muestra una notable ventaja en ambas tablas, siendo el mejor método para doce conjuntos de datos tanto al considerar AUC-ROC como al considerar AUC-PR. Si se observa el desempeño promedio de los métodos, existe una notable diferencia entre AD-DMKDE y los siguientes mejores métodos (KNN, Isolation Forest y Covariance). Los métodos profundos, aunque presentan algunos buenos resultados para los conjuntos de datos con altas dimensiones (como VAE-Bayes en Musk, DSVDD en Speech o LAKE en SpamBase), en general se quedan atrás de varios de los métodos clásicos.

Al tener en cuenta las características de los diferentes conjuntos de datos, es posible determinar en qué escenarios se puede esperar que AD-DMKDE presente un mejor desempeño que otros métodos. El tamaño del conjunto de datos parece no tener influencia, ya que el método se desempeña bien en conjuntos de menos de mil muestras (Glass, Pima), en conjuntos de entre mil y cinco mil muestras (Letter, SpamBase) y en conjuntos de más de cinco mil muestras (Optdigits, Satellite). Esto puede deberse a que la matriz de densidad que AD-DMKDE usa es independiente del tamaño del conjunto de datos, y sólo depende del tamaño definido por la función de mapeo del proceso AFF. Por su parte, la tasa de anomalías parece mostrar una influencia débil en el desempeño de AD-DMKDE, ya que el método presenta buenos resultados en conjuntos de datos con más de 25 % de anomalías (Ionosphere, Pima) pero levemente menores en conjuntos de datos con anomalías por debajo del 20 % (Vowels, Mnist, WBC). En general, cuando hay un número muy bajo de anomalías, se hace más difícil para los modelos de detección de anomalías construir una separación efectiva debido a la falta de información sobre éstas.

El determinante más fuerte sobre el desempeño del método es la dimensionalidad de los datos, ya que AD-DMKDE destaca más en conjuntos de datos de menos de cien dimensiones (Vertebral, Pendigits, Vowels), especialmente cuando los datos presentan más de veinte dimensiones (Satellite, SpamBase, Ionosphere), pero no es tan competitivo ante conjuntos de datos de más de cien dimensiones (sólo es primero en Musk, y no logra primer ni segundo lugar en Arrhythmia o Speech). Para explicar este resultado, es necesario acudir a la columna de AD-DMKDE de la Tabla **5-1**. En esta tabla, podemos ver que el hiperparámetro *rff_comps*, que indica la dimensión del espacio de características, sólo toma los valores 1000, 2000 o 4000. Para conjuntos de datos con alta dimensionalidad, estos tamaños de la transformación pueden no ser suficientes para captar adecuadamente todos los detalles de los datos originales, lo que explicaría la baja en el desempeño del método. Sin embargo, valores muy altos de este parámetro afectarían la rapidez del método, aumentando el tiempo de ejecución y el consumo de memoria.

Tabla 5-2: Área bajo la curva ROC (AUC-ROC) para todos los algoritmos aplicados sobre todos los conjuntos de datos. Los valores más altos han sido resaltados, el primero en negrilla y el segundo en subrayado.

	OCSVM	IForest	Cov.	LOF	KNN	SOS	COPOD	LODA	VAE-B	DSVDD	ALAD	LAKE	AD-DMKDE
Annthyroid	0,617	0,833	**0,918**	0,710	0,704	0,642	0,799	0,660	0,704	0,510	0,687	0,700	0,759
Arrhythmia	0,807	**0,863**	0,817	0,832	0,776	0,738	0,787	0,676	0,769	0,707	0,724	0,310	0,790
Breastw	0,899	**0,997**	0,985	0,518	0,993	0,927	0,993	0,962	0,961	0,946	0,969	0,930	0,993
Cardio	0,893	0,951	0,863	0,645	0,937	0,829	0,919	0,677	**0,953**	0,655	0,924	0,474	0,951
ForestCover	0,877	0,939	0,686	0,558	0,891	-	0,877	0,905	0,933	0,500	0,859	0,878	**0,956**
Glass	0,212	0,635	0,740	0,221	0,788	0,558	0,654	0,490	0,663	0,760	0,529	0,250	**0,856**
Ionosphere	0,743	0,772	0,916	0,842	0,964	0,840	0,853	0,456	0,833	0,965	0,874	0,931	**0,989**
Letter	0,462	0,647	0,830	0,837	0,822	0,816	0,532	0,464	0,499	0,554	0,509	0,138	**0,890**
Mammography	0,557	0,880	0,747	0,786	0,845	0,626	**0,908**	0,773	0,883	0,531	0,840	0,703	0,841
MNIST	0,693	0,800	**0,908**	0,767	0,876	0,739	0,784	0,658	0,856	0,543	0,734	0,842	0,900
Musk	0,813	0,993	0,999	0,442	**1,000**	0,856	0,958	0,938	**1,000**	0,408	0,717	0,881	**1,000**
OptDigits	0,424	0,568	0,355	0,417	0,394	0,548	0,694	0,215	0,522	0,433	0,368	0,330	**0,972**
PenDigits	0,865	0,941	0,862	0,374	0,960	0,683	0,907	0,925	0,939	0,516	0,896	0,650	**0,989**
Pima	0,533	0,666	0,685	0,638	0,677	0,589	0,613	0,378	0,608	0,574	0,604	0,465	**0,705**
Satellite	0,582	0,674	0,790	0,542	0,758	0,563	0,658	0,598	0,625	0,498	0,609	0,294	**0,880**
SatImage	0,924	0,997	0,994	0,690	**0,998**	0,816	0,979	0,995	0,990	0,540	0,939	0,837	0,997
Shuttle	0,927	**0,994**	0,989	0,541	0,716	0,505	0,993	0,973	0,988	0,501	0,988	0,634	0,987
SpamBase	0,667	0,738	0,669	0,384	0,722	0,581	0,766	0,637	0,692	0,536	0,630	0,851	**0,962**
Speech	0,495	0,483	0,516	0,597	0,507	0,568	0,534	0,562	0,505	**0,651**	0,582	0,531	0,416
Thyroid	0,929	0,990	**0,993**	0,951	0,945	0,741	0,937	0,585	0,958	0,605	0,972	0,925	0,944
Vertebral	0,690	0,483	0,498	0,524	0,477	0,450	0,477	0,383	0,566	0,443	0,501	0,523	**0,779**
Vowels	0,580	0,668	0,653	0,912	0,961	0,825	0,539	0,540	0,613	0,555	0,733	0,298	**0,985**
WBC	0,464	0,940	0,940	**0,976**	0,909	0,918	0,940	0,909	0,898	0,633	0,762	0,371	0,940
Wine	0,800	0,922	**0,967**	**0,967**	0,900	0,500	0,944	0,767	0,833	0,556	0,900	0,700	0,922
Promedio	0,686	0,807	0,805	0,653	0,813	0,689	0,794	0,672	0,783	0,576	0,744	0,531	**0,894**

Tabla 5-3: Área bajo la curva Precision-Recall (AUC-PR) para todos los algoritmos aplicados sobre todos los conjuntos de datos. Los valores más altos han sido resaltados, el primero en negrilla y el segundo en subrayado.

	OCSVM	IForest	Cov.	LOF	KNN	SOS	COPOD	LODA	VAE-B	DSVDD	ALAD	LAKE	AD-DMKDE
Annthyroid	0,127	0,340	**0,504**	0,186	0,212	0,142	0,197	0,147	0,209	0,083	0,192	0,174	0,187
Arrhythmia	0,425	0,558	0,459	0,433	0,514	0,472	**0,566**	0,365	0,532	0,320	0,393	0,116	0,534
Breastw	0,825	**0,994**	0,969	0,335	0,988	0,886	0,987	0,963	0,968	0,904	0,958	0,864	0,987
Cardio	0,465	**0,693**	0,469	0,185	0,555	0,254	0,581	0,198	0,595	0,267	0,507	0,116	0,627
ForestCover	**0,116**	0,097	0,015	0,015	0,060	-	0,060	0,065	0,070	0,010	0,043	0,095	0,115
Glass	0,035	0,072	0,097	0,035	0,121	0,072	0,076	0,058	0,077	0,143	0,056	0,036	**0,167**
Ionosphere	0,754	0,711	0,920	0,837	0,951	0,843	0,770	0,449	0,778	0,958	0,848	0,228	**0,984**
Letter	0,075	0,111	0,241	0,375	0,228	0,288	0,072	0,063	0,087	0,271	0,081	0,036	**0,489**
Mammography	0,042	0,307	0,145	0,113	0,193	0,072	**0,416**	0,154	0,196	0,078	0,190	0,072	0,187
MNIST	0,216	0,326	0,494	0,309	0,468	0,228	0,246	0,185	0,431	0,148	0,264	0,333	**0,549**
Musk	0,214	0,891	0,982	0,032	0,992	0,251	0,449	0,545	**1,000**	0,026	0,142	0,853	**1,000**
OptDigits	0,024	0,034	0,021	0,032	0,022	0,032	0,044	0,017	0,028	0,026	0,021	0,021	**0,526**
PenDigits	0,118	0,339	0,092	0,018	0,208	0,063	0,162	0,289	0,208	0,048	0,121	0,060	**0,614**
Pima	0,393	0,463	0,490	0,463	0,497	0,415	0,479	0,339	0,45	0,453	0,424	0,319	**0,596**
Satellite	0,540	0,660	0,760	0,360	0,610	0,354	0,601	0,596	0,631	0,320	0,561	0,230	**0,882**
SatImage	0,394	0,912	0,635	0,06	**0,955**	0,056	0,751	0,923	0,822	0,014	0,575	0,175	0,764
Shuttle	0,568	**0,968**	0,842	0,100	0,190	0,111	0,954	0,911	0,922	0,072	0,878	0,163	0,957
SpamBase	0,273	0,372	0,307	0,155	0,345	0,260	0,440	0,302	0,348	0,231	0,264	0,576	**0,818**
Speech	0,019	0,023	0,022	**0,044**	0,019	0,024	0,020	0,029	0,020	0,028	0,020	0,026	0,016
Thyroid	0,286	0,675	**0,688**	0,293	0,324	0,082	0,223	0,131	0,442	0,125	0,402	0,013	0,374
Vertebral	0,218	0,133	0,138	0,153	0,125	0,121	0,119	0,106	0,165	0,077	0,123	0,164	**0,260**
Vowels	0,083	0,074	0,050	0,374	0,641	0,199	0,049	0,050	0,148	0,217	0,075	0,267	**0,757**
WBC	0,229	0,587	0,597	**0,760**	0,484	0,476	0,700	0,626	0,506	0,192	0,161	0,237	0,703
Wine	0,242	0,610	**0,756**	0,639	0,444	0,091	0,533	0,494	0,297	0,137	0,421	0,274	0,467
Promedio	0,278	0,456	0,446	0,263	0,423	0,252	0,396	0,334	0,414	0,215	0,322	0,227	**0,565**

5.2. Análisis estadístico

Existen varios mecanismos estadísticos para determinar si las diferencias encontradas entre los métodos son verdaderamente significativas. Uno de estos mecanismos es el Test de Friedman, que permite comparar varias poblaciones o grupos al mismo tiempo al analizar los rankings de sus observaciones. El test no trabaja con los datos originales; para cada observación, se toma su posición al ordenarla respecto a todos los valores dentro de su grupo. Por ejemplo, si una observación dada es la quinta más alta en su grupo, es cambiada por el número 5 (su ranking) para efectos del test. En este caso, los rankings se calculan para cada conjunto de datos, es decir, el mejor valor para cada conjunto tendrá un ranking de 1, el segundo mejor un ranking de 2, y así sucesivamente. La implementación de este test se realizó a través de la librería Scipy del lenguaje Python.

El estadístico del test de Friedman está dado por:

$$Q = \left[\frac{12}{Nk(k+1)} \sum_{j=1}^{k} R_j^2 \right] - 3N(k+1)$$

donde N es el número de mediciones (en este caso, la cantidad de pruebas realizadas), y k es el número de poblaciones (que sería la cantidad de algoritmos utilizados). R_j^2 simboliza la suma elevada al cuadrado de todos los rankings del algoritmo j-ésimo. Una vez obtenido el valor de Q, se calcula un p-valor dado por $\mathbf{P}[\chi_{k-1}^2 \geq Q]$, siendo χ_{k-1}^2 la distribución chi-cuadrado con $(k-1)$ grados de libertad; el p-valor se compara con un valor de confianza predeterminado, denotado α, de tal forma que si el p-valor es menor a α se asegura que la diferencia entre los grupos es significativa. Para este test se eligió un valor $\alpha = 0{,}05$, que garantiza una confianza del 95 %. La aplicación del test de Friedman sobre los resultados en AUCROC y AUC-PR entregó los estadísticos $Q = 110{,}795$ y $Q = 92{,}233$ respectivamente, cuyos p-valores son $4{,}16 \times 10^{-18}$ y $1{,}82 \times 10^{-14}$, lo que representa una sólida evidencia a favor de la significancia de la diferencia entre los métodos.

Sin embargo, el test de Friedman no indica cuáles son los grupos (o en este caso, algoritmos) responsables de dicha diferencia. Para identificarlos, se utilizan las pruebas *post-hoc*, dentro de las cuales se eligió utilizar una extensión del test de Friedman conocida como Test de Friedman-Nemenyi, que sí permite determinar si la diferencia es significativa entre todos los posibles pares de algoritmos. Este test fue aplicado dos veces, una por cada métrica, y sus resultados pueden verse de forma gráfica en las Figuras **5-1** y **5-2**. En estas retículas, conocidas como *chessboards*, los cuadrados negros corresponden a pares de algoritmos que difieren significativamente, y los cuadrados blancos a pares que no difieren, con un valor de confianza $\alpha = 0{,}05$. Los cuadrados grises en la diagonal señalan que los pares formados por el mismo método no se tienen en cuenta.

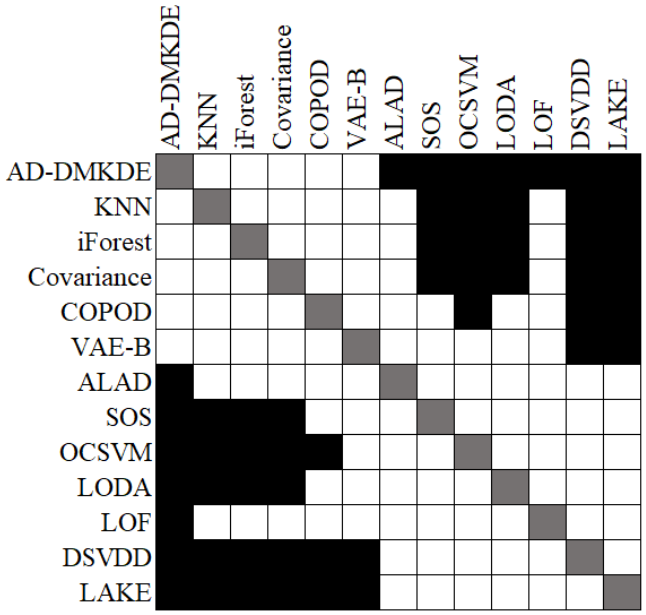

Figura 5-1: Resultados del test de Friedman-Nemenyi sobre los datos de la tabla **5-2**.

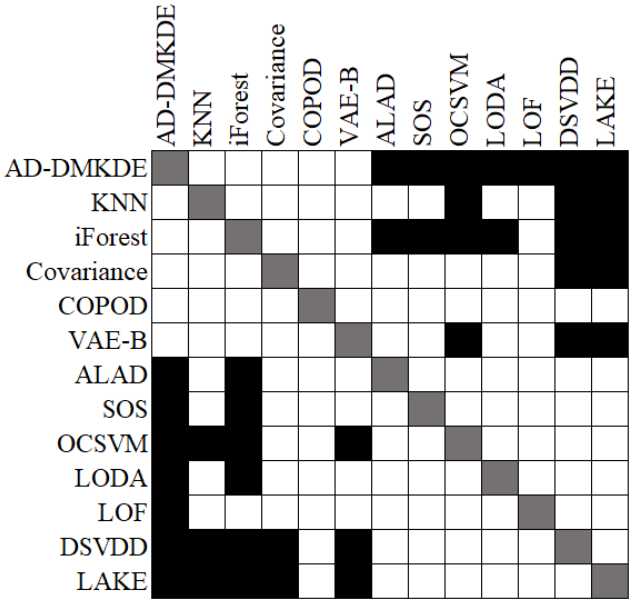

Figura 5-2: Resultados del test de Friedman-Nemenyi sobre los datos de la tabla **5-3**.

Al observar la Figura **5-1**, se puede observar que AD-DMKDE es el método más diferente a los otros, presentando diferencias significativas con siete de los doce métodos de línea base. En este caso, los algoritmos pueden ser separados en varias categorías que muestran comportamientos similares: por un lado, los métodos con mejor desempeño, como Isolation

Forest, Covariance y KNN, muestran el mismo patrón, el cual es altamente similar al de AD-DMKDE; VAE-Bayes, COPOD, ALAD y LOF pueden señalarse como una categoría intermedia, ya que se distinguen de pocos métodos; y los métodos restantes (OCSVM, SOS, LODA, DeepSVDD y LAKE) muestran patrones parecidos entre sí que suelen contrariar al de AD-DMKDE, por lo que se podrían señalar como los de peor desempeño.

Respecto a la Figura **5-2**, nuevamente AD-DMKDE es el más diferente de los demás, al distinguirse de siete de los métodos, seguido por Isolation Forest, con un comportamiento muy parecido. Aunque una categorización completa no es evidente como en el caso anterior, se pueden distinguir algunos métodos con patrones similares, incluyendo a KNN con VAE-Bayes, a SOS con LODA y ALAD, y a DeepSVDD con LAKE y con OCSVM en menor medida. Cabe mencionar a COPOD, el único método que no presentó diferencias significativas con ningún otro; esto es un indicador de que el comportamiento general de AUC-PR fue menos disperso entre algoritmos que el de AUCROC.

6 Conclusiones y Trabajo Futuro

6.1. Conclusiones

Este documento ha presentado los resultados de la investigación realizada en el marco de la Tesis de Maestría de su autor, la cual se centra en el desarrollo, implementación y prueba de AD-DMKDE (*Anomaly Detection through Density Matrices and Fourier Features*), un algoritmo para detección de anomalías que integra Kernel Density Estimation (KDE) con conceptos como las Random Fourier Features y las matrices de densidad, con el fin de preservar las ventajas de KDE en términos de la interpretabilidad de sus resultados, y abordar sus debilidades en términos de complejidad computacional. En este sentido, AD-DMKDE genera *scores* de anomalías de tal forma que los *scores* más altos corresponden a los puntos ubicados en regiones de baja densidad; así, los datos anómalos se entienden como aquellos que tienen una probabilidad muy baja de aparecer. Además, al contrario que en KDE, la complejidad del algoritmo propuesto en términos de memoria no depende del tamaño del conjunto de datos, sino de un hiperparámetro cuyo valor se puede controlar *a priori*; esto hace que AD-DMKDE presente una ventaja sobre KDE a la hora de procesar conjuntos de datos muy grandes.

La ventaja en memoria de AD-DMKDE sobre KDE se puede explicar gracias al uso de las matrices de densidad en combinación con la medición cuántica, dos elementos que permiten crear y utilizar representaciones compactas de la distribución de los datos normales. El algoritmo también hace uso de las Random Fourier Features como un mecanismo para transformar los datos de tal forma que se reduce la complejidad del cálculo de los kernels sobre éstos; esta función de mapeo puede optimizarse a través de una red neuronal, en un proceso denominado AFF (Adaptive Fourier Features). El modelo plantea una posible variación, en la que se representa la matriz de densidad a través de su descomposición espectral, permitiendo una reducción aún mayor en sus requerimientos de memoria.

El desempeño del algoritmo propuesto AD-DMKDE fue medido a través del planteamiento de un marco de evaluación experimental, en el cual fue comparado contra doce métodos de línea base, tanto clásicos como profundos, cuyos detalles se exploran en la sección 2.1.3. Todos los métodos fueron aplicados sobre veinticuatro conjuntos de datos previamente etiquetados, los cuales presentan una amplia variedad de características y provienen de diversas áreas del conocimiento. Los algoritmos fueron evaluados a través de dos métricas, el área

bajo la curva ROC (AUCROC) y el área bajo la curva Precision-Recall (AUC-PR), que son comúnmente utilizadas en publicaciones del área de detección de anomalías. Los resultados de los distintos algoritmos muestran una ventaja en el desempeño promedio de AD-DMKDE sobre la línea base, presentando las métricas más altas en la mitad de los conjuntos de datos considerados. Esta diferencia a favor del método propuesto se evaluó a través de tests estadísticos que confirmaron la significancia de esta ventaja. Además, aunque el método ha sido probado mediante un esquema supervisado, no requiere de la presencia de etiquetas pre-existentes para generar sus salidas, por lo que también puede ser utilizado en escenarios de carácter semi-supervisado o no supervisado.

6.2. Trabajo Futuro

Aunque los resultados del desempeño de AD-DMKDE son generalmente positivos al realizar la comparativa con los demás métodos, el algoritmo propuesto también presentó algunos problemas para abordar ciertos conjuntos de datos, en especial aquellos con alta dimensionalidad. Este es un fenómeno al que muchos métodos clásicos ya se han enfrentado, y para el que la solución más común ha sido la adopción de métodos profundos capaces de procesar estos datos directamente o de reducir su dimensionalidad de forma robusta como una etapa de procesamiento previa. AD-DMKDE hace uso de redes neuronales en algunas etapas (específicamente, en el proceso AFF durante la fase de entrenamiento), pero no aprovecha todo el potencial de estos modelos de la forma en que otras propuestas lo hacen.

Con el ánimo de probar que AD-DMKDE también es susceptible de combinarse con redes neuronales para superar las limitaciones encontradas, el autor de esta investigación, junto con otros miembros del grupo de investigación MindLab, ha hecho parte del desarrollo de LEAN-DMKDE, un nuevo algoritmo basado en AD-DMKDE que agrega un autoencoder como primera fase del método. Este autoencoder procesa los datos para transformarlos a un espacio latente de menor dimensionalidad, y los datos codificados en este espacio son los que ingresan como entradas a la segunda parte del método, que es esencialmente similar al aquí presentado. Sin embargo, una adición interesante de LEAN-DMKDE es el uso de dos *scores* de anomalías, incluyendo tanto el asociado a la estimación como el asociado al error de reconstrucción. Estas dos medidas se combinan mediante un parámetro de *tradeoff*, que balancea la contribución de cada *score*, y el cual también es susceptible de ser optimizado mediante un proceso de búsqueda exhaustiva como el realizado en esta investigación.

Otra limitación en AD-DMKDE se relaciona con el diseño del método, orientado al análisis de conjuntos de datos en los que se supone que se ha realizado un proceso previo de limpieza y cribado de los datos. Este no suele ser el caso en escenarios realistas, por lo que una de las líneas de trabajo que el autor quisiera retomar a futuro se centra en la adaptación de

AD-DMKDE a problemas reales con datos que requieran de análisis y preprocesamiento exhaustivo antes de aplicar el método propiamente dicho. En este sentido, un desarrollo adicional sobre AD-DMKDE, denominado InQMAD, busca adaptar el método a escenarios de *streaming*, en los que los datos llegan de forma continua y permanente, por lo que se exigen tiempos de respuesta mucho menores y la capacidad de aprender constantemente de los nuevos datos sin olvidar el conocimiento acumulado. Este desarrollo, aunque más incipiente que AD-DMKDE o LEAN-DMKDE, ha mostrado un potencial interesante ante varios conjuntos de datos de este tipo; esto hace que InQMAD también sea una línea de trabajo prometedora, a medida que los datos de *streaming* se hacen cada vez más comunes y demandan mejores análisis. Los artículos con los resultados preliminares de estas dos líneas de trabajo pueden encontrarse en la sección 1.3 del documento. El autor espera continuar trabajando en ellas en los próximos años, además de sentar las bases para el desarrollo de nuevas ideas en detección de anomalías que utilicen a AD-DMKDE como elemento central.

Bibliografía

[1] Aggarwal, Charu C. ; Aggarwal, Charu C.: *An introduction to outlier analysis*. Springer, 2017

[2] Agrawal, Shikha ; Agrawal, Jitendra: Survey on anomaly detection using data mining techniques. En: *Procedia Computer Science* 60 (2015), p. 708–713

[3] Ahmed, Faruk ; Courville, Aaron: Detecting semantic anomalies. En: *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 34, 2020, p. 3154–3162

[4] Albuquerque Filho, JE ; Brandão, Laislla C. ; Fernandes, Bruno J. ; Maciel, Alexandre M.: A Review of Neural Networks for Anomaly Detection. En: *IEEE Access* (2022)

[5] Alloqmani, Ahad ; Abushark, Yoosef B. ; Khan, Asif I. ; Alsolami, Fawaz: Deep learning based anomaly detection in images: insights, challenges and recommendations. En: *International Journal of Advanced Computer Science and Applications* 12 (2021), Nr. 4

[6] Alpaydin, Ethem: *Introduction to machine learning*. MIT press, 2020

[7] An, Jinwon ; Cho, Sungzoon: Variational autoencoder based anomaly detection using reconstruction probability. En: *Special lecture on IE* 2 (2015), Nr. 1, p. 1–18

[8] Arshad, Kinza ; Ali, Rao F. ; Muneer, Amgad ; Aziz, Izzatdin A. ; Naseer, Sheraz ; Khan, Nabeel S. ; Taib, Shakirah M.: Deep Reinforcement Learning for Anomaly Detection: A Systematic Review. En: *IEEE Access* (2022)

[9] Ash, Robert B.: *Information theory*. Courier Corporation, 2012

[10] Bergmann, Paul ; Batzner, Kilian ; Fauser, Michael ; Sattlegger, David ; Steger, Carsten: The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. En: *International Journal of Computer Vision* 129 (2021), Nr. 4, p. 1038–1059

[11] Breunig, Markus M. ; Kriegel, Hans-Peter ; Ng, Raymond T. ; Sander, Jörg: LOF: identifying density-based local outliers. En: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, p. 93–104

[12] Brukner, Časlav: On the quantum measurement problem. En: *Quantum [Un] Speakables II: Half a Century of Bell's Theorem* (2017), p. 95–117

[13] Brumer, Paul ; Gong, Jiangbin: Born rule in quantum and classical mechanics. En: *Physical Review A* 73 (2006), Nr. 5, p. 052109

[14] Burkard, Guido ; Ladd, Thaddeus D. ; Nichol, John M. ; Pan, Andrew ; Petta, Jason R.: Semiconductor spin qubits. En: *arXiv preprint arXiv:2112.08863* (2021)

[15] Bustos-Brinez, Oscar ; Gallego-Mejia, Joseph ; González, Fabio A.: AD-DMKDE: Anomaly Detection through Density Matrices and Fourier Features. (2022)

[16] Bustos-Brinez, Oscar ; Gallego-Mejia, Joseph ; González, Fabio A.: AD-DMKDE: Anomaly Detection Through Density Matrices and Fourier Features. En: *Information Technology and Systems: ICITS 2023, Volume 1*. Springer, 2023

[17] Chalapathy, Raghavendra ; Chawla, Sanjay: Deep learning for anomaly detection: A survey. En: *arXiv preprint arXiv:1901.03407* (2019)

[18] Chandola, Varun ; Banerjee, Arindam ; Kumar, Vipin: Anomaly detection: A survey. En: *ACM computing surveys (CSUR)* 41 (2009), Nr. 3, p. 1–58

[19] Chen, Yen-Chi: A tutorial on kernel density estimation and recent advances. En: *Biostatistics & Epidemiology* 1 (2017), Nr. 1, p. 161–187

[20] Ciliberto, Carlo ; Herbster, Mark ; Ialongo, Alessandro D. ; Pontil, Massimiliano ; Rocchetto, Andrea ; Severini, Simone ; Wossnig, Leonard: Quantum machine learning: a classical perspective. En: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474 (2018), Nr. 2209, p. 20170551

[21] Ding, Kaize ; Zhou, Qinghai ; Tong, Hanghang ; Liu, Huan: Few-shot network anomaly detection via cross-network meta-learning. En: *Proceedings of the Web Conference 2021*, 2021, p. 2448–2456

[22] Eimann, Raimund E.: *Network event detection with entropy measures*, ResearchSpace@ Auckland, Tesis de Grado, 2008

[23] Erhan, Laura ; Ndubuaku, M ; Di Mauro, Mario ; Song, Wei ; Chen, Min ; Fortino, Giancarlo ; Bagdasar, Ovidiu ; Liotta, Antonio: Smart anomaly detection in sensor systems: A multi-perspective review. En: *Information Fusion* 67 (2021), p. 64–79

[24] Filzmoser, Peter: *A multivariate outlier detection method*. Citeseer, 2004

[25] Fraser, Katherine ; Homiller, Samuel ; Mishra, Rashmish K. ; Ostdiek, Bryan ; Schwartz, Matthew D.: Challenges for unsupervised anomaly detection in particle physics. En: *Journal of High Energy Physics* 2022 (2022), Nr. 3, p. 1–31

[26] Gallego M, Joseph A. ; González, Fabio A.: Quantum Adaptive Fourier Features for Neural Density Estimation. En: *arXiv e-prints* (2022), p. arXiv–2208

[27] Gallego-Mejia, Joseph ; Bustos-Brinez, Oscar ; González, Fabio A.: LEAN-DMKDE: Quantum Latent Density Estimation for Anomaly Detection. En: *arXiv preprint arXiv:2211.08525* (2022)

[28] Gallego-Mejia, Joseph A. ; Bustos-Brinez, Oscar A. ; González, Fabio A.: InQ-MAD: Incremental Quantum Measurement Anomaly Detection. En: *2022 IEEE International Conference on Data Mining Workshops (ICDMW)* IEEE, 2022, p. 787–796

[29] Goix, Nicolas ; Sabourin, Anne ; Clémençon, Stéphan: On anomaly ranking and excess-mass curves. En: *Artificial Intelligence and Statistics* PMLR, 2015, p. 287–295

[30] González, Fabio A. ; Gallego, Alejandro ; Toledo-Cortés, Santiago ; Vargas-Calderón, Vladimir: Learning with density matrices and random features. En: *Quantum Machine Intelligence* 4 (2022), Nr. 2

[31] González, Fabio A. ; Vargas-Calderón, Vladimir ; Vinck-Posada, Herbert: Classification with quantum measurements. En: *Journal of the Physical Society of Japan* 90 (2021), Nr. 4, p. 044002

[32] Gühne, Otfried ; Tóth, Géza: Entanglement detection. En: *Physics Reports* 474 (2009), Nr. 1-6, p. 1–75

[33] Hagemann, Tanja ; Katsarou, Katerina: A systematic review on anomaly detection for cloud computing environments. En: *2020 3rd Artificial Intelligence and Cloud Computing Conference*, 2020, p. 83–96

[34] Hall, Brian C.: Systems and subsystems, multiple particles. En: *Quantum theory for mathematicians*. Springer, 2013, p. 419–440

[35] Harrou, Fouzi ; Kadri, Farid ; Chaabane, Sondes ; Tahon, Christian ; Sun, Ying: Improved principal component analysis for anomaly detection: Application to an emergency department. En: *Computers & Industrial Engineering* 88 (2015), p. 63–77

[36] Hayashi, Masahito ; Ishizaka, Satoshi ; Kawachi, Akinori ; Kimura, Gen ; Ogawa, Tomohiro: *Introduction to quantum information science*. Springer, 2014

[37] Hewitt, Joshua ; Gelfand, Alan E. ; Quick, Nicola J. ; Cioffi, William R. ; Southall, Brandon L. ; DeRuiter, Stacy L. ; Schick, Robert S.: Kernel density estimation of conditional distributions to detect responses in satellite tag data. En: *Animal Biotelemetry* 10 (2022), Nr. 1, p. 28

[38] Jaeger, Gregg: *Quantum information*. Springer, 2007

[39] Janssens, J.H.M. ; Huszar, F. ; Postma, E.O. ; van den Herik, H.J.: Stochastic Outlier Selection. En: *Tilburg centre for Creative Computing, techreport* 1 (2012), p. 2012

[40] Kalair, Kieran ; Connaughton, Colm: Anomaly detection and classification in traffic flow data from fluctuations in the flow–density relationship. En: *Transportation Research Part C: Emerging Technologies* 127 (2021), p. 103178

[41] Kalinichenko, Leonid ; Shanin, Ivan ; Taraban, Ilia: Methods for anomaly detection: A survey. En: *CEUR workshop proceedings* Vol. 1297, 2014, p. 2025

[42] Keyl, Michael: Fundamentals of quantum information theory. En: *Physics reports* 369 (2002), Nr. 5, p. 431–548

[43] Khan, Tariq M. ; Robles-Kelly, Antonio: Machine learning: Quantum vs classical. En: *IEEE Access* 8 (2020), p. 219275–219294

[44] Kim, Junbong ; Jeong, Kwanghee ; Choi, Hyomin ; Seo, Kisung: GAN-based anomaly detection in imbalance problems. En: *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16* Springer, 2020, p. 128–145

[45] Kingma, Diederik P. ; Welling, Max: Auto-encoding variational bayes. En: *arXiv preprint arXiv:1312.6114* (2013)

[46] Knorr, Edwin M. ; Ng, Raymond T. ; Tucakov, Vladimir: Distance-based outliers: algorithms and applications. En: *The VLDB Journal* 8 (2000), Nr. 3, p. 237–253

[47] Kok, Pieter ; Munro, William J. ; Nemoto, Kae ; Ralph, Timothy C. ; Dowling, Jonathan P. ; Milburn, Gerard J.: Linear optical quantum computing with photonic qubits. En: *Reviews of modern physics* 79 (2007), Nr. 1, p. 135

[48] Kriegel, Hans-Peter ; Kröger, Peer ; Zimek, Arthur: Outlier detection techniques. En: *Tutorial at KDD* 10 (2010), p. 1–76

[49] Kulkarni, Viraj ; Kulkarni, Milind ; Pant, Aniruddha: Quantum computing methods for supervised learning. En: *Quantum Machine Intelligence* 3 (2021), Nr. 2, p. 1–14

[50] Landsman, Nicolaas P.: Born rule and its interpretation. En: *Compendium of quantum physics*. Springer, 2009, p. 64–70

[51] Leggett, Anthony J.: The quantum measurement problem. En: *science* 307 (2005), Nr. 5711, p. 871–872

[52] Li, Zheng ; Zhao, Yue ; Botta, Nicola ; Ionescu, Cezar ; Hu, Xiyang: COPOD: Copula-Based Outlier Detection, 2020. – ISSN 2374–8486, p. 1118–1123

[53] Lindemann, Benjamin ; Maschler, Benjamin ; Sahlab, Nada ; Weyrich, Michael: A survey on anomaly detection for technical systems using LSTM networks. En: *Computers in Industry* 131 (2021), p. 103498

[54] Liu, Boyang ; Tan, Pang-Ning ; Zhou, Jiayu: Unsupervised anomaly detection by robust density estimation. En: *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 36, 2022, p. 4101–4108

[55] Liu, Fei T. ; Ting, Kai M. ; Zhou, Zhi-Hua: Isolation forest. En: *2008 eighth ieee international conference on data mining* IEEE, 2008, p. 413–422

[56] Lv, Peng ; Yu, Yanwei ; Fan, Yangyang ; Tang, Xianfeng ; Tong, Xiangrong: Layer-constrained variational autoencoding kernel density estimation model for anomaly detection. En: *Knowledge-Based Systems* 196 (2020). – ISSN 09507051

[57] Ma, Xiaoxiao ; Wu, Jia ; Xue, Shan ; Yang, Jian ; Zhou, Chuan ; Sheng, Quan Z. ; Xiong, Hui ; Akoglu, Leman: A comprehensive survey on graph anomaly detection with deep learning. En: *IEEE Transactions on Knowledge and Data Engineering* (2021)

[58] Mäkelä, H ; Messina, Antonino: N-qubit states as points on the Bloch sphere. En: *Physica Scripta* 2010 (2010), Nr. T140, p. 014054

[59] Mishra, Sidharth P. ; Sarkar, Uttam ; Taraphder, Subhash ; Datta, Sanjay ; Swain, D ; Saikhom, Reshma ; Panda, Sasmita ; Laishram, Menalsh: Multivariate statistical data analysis-principal component analysis (PCA). En: *International Journal of Livestock Research* 7 (2017), Nr. 5, p. 60–78

[60] Moeller, John ; Srikumar, Vivek ; Swaminathan, Sarathkrishna ; Venkatasubramanian, Suresh ; Webb, Dustin: Continuous kernel learning. En: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II 16* Springer, 2016, p. 657–673

[61] Nachman, Benjamin ; Shih, David: Anomaly detection with density estimation. En: *Physical Review D* 101 (2020), Nr. 7, p. 075042

[62] Nassif, Ali B. ; Talib, Manar A. ; Nasir, Qassim ; Dakalbab, Fatima M.: Machine learning for anomaly detection: A systematic review. En: *Ieee Access* 9 (2021), p. 78658–78700

[63] Nayak, Rashmiranjan ; Pati, Umesh C. ; Das, Santos K.: A comprehensive review on deep learning-based methods for video anomaly detection. En: *Image and Vision Computing* 106 (2021), p. 104078

[64] Nielsen, Michael A. ; Chuang, Isaac L.: *Quantum computation and Quantum information. 10th Anniversary edition.* Cambridge University Press, 2010

[65] Nowak-Brzezińska, Agnieszka ; Horyń, Czesław: Outliers in rules-the comparision of LOF, COF and KMEANS algorithms. En: *Procedia Computer Science* 176 (2020), p. 1420–1429

[66] Pang, Guansong ; Shen, Chunhua ; Cao, Longbing ; Hengel, Anton Van D.: Deep learning for anomaly detection: A review. En: *ACM computing surveys (CSUR)* 54 (2021), Nr. 2, p. 1–38

[67] Park, Cheong H.: A Comparative Study for Outlier Detection Methods in High Dimensional Text Data. En: *Journal of Artificial Intelligence and Soft Computing Research* 13 (2023), Nr. 1, p. 5–17

[68] Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; Blondel, M. ; Prettenhofer, P. ; Weiss, R. ; Dubourg, V. ; Vanderplas, J. ; Passos, A. ; Cournapeau, D. ; Brucher, M. ; Perrot, M. ; Duchesnay, E.: Scikit-learn: Machine Learning in Python. En: *Journal of Machine Learning Research* 12 (2011), p. 2825–2830

[69] Pevný, Tomáš: Loda: Lightweight on-line detector of anomalies. En: *Machine Learning* 102 (2016), p. 275–304. – ISSN 1573–0565

[70] Pierna, JA F. ; Wahl, F ; De Noord, OE ; Massart, DL: Methods for outlier detection in prediction. En: *Chemometrics and Intelligent Laboratory Systems* 63 (2002), Nr. 1, p. 27–39

[71] Rahimi, Ali ; Recht, Benjamin: Random Features for Large-Scale Kernel Machines. En: *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Curran Associates Inc., 2007 (NIPS'07). – ISBN 9781605603520, p. 1177–1184

[72] Ramaswamy, Sridhar ; Rastogi, Rajeev ; Shim, Kyuseok: Efficient Algorithms for Mining Outliers from Large Data Sets, Association for Computing Machinery, 2000. – ISBN 1581132174, p. 427438

[73] Ramchoun, Hassan ; Ghanou, Youssef ; Ettaouil, Mohamed ; Janati Idrissi, Mohammed A.: Multilayer perceptron: Architecture optimization and training. (2016)

[74] Rayana, Shebuti. *ODDS Library*. 2016

[75] Rettig, Laura ; Khayati, Mourad ; Cudré-Mauroux, Philippe ; Piórkowski, Michał: Online anomaly detection over big data streams. En: *Applied Data Science: Lessons Learned for the Data-Driven Business* (2019), p. 289–312

[76] Reynolds, Douglas A. [u. a.]: Gaussian mixture models. En: *Encyclopedia of biometrics* 741 (2009), Nr. 659-663

[77] Rousseeuw, Peter J. ; Driessen, Katrien V.: A fast algorithm for the minimum covariance determinant estimator. En: *Technometrics* 41 (1999), Nr. 3, p. 212–223

[78] Ruff, L ; Kauffmann, J R. ; Vandermeulen, R A. ; Montavon, G ; Samek, W ; Kloft, M ; Dietterich, T G. ; Muller, K.-R.: A Unifying Review of Deep and Shallow Anomaly Detection. En: *Proceedings of the IEEE* 109 (2021), p. 756–795

[79] Ruff, Lukas ; Vandermeulen, Robert ; Goernitz, Nico ; Deecke, Lucas ; Siddiqui, Shoaib A. ; Binder, Alexander ; Müller, Emmanuel ; Kloft, Marius: Deep One-Class Classification, PMLR, 3 2018, p. 4393–4402

[80] Schölkopf, Bernhard ; Platt, John C. ; Shawe-Taylor, John ; Smola, Alex J. ; Williamson, Robert C.: Estimating the support of a high-dimensional distribution. En: *Neural computation* (2001)

[81] Shaukat, Kamran ; Alam, Talha M. ; Luo, Suhuai ; Shabbir, Shakir ; Hameed, Ibrahim A. ; Li, Jiaming ; Abbas, Syed K. ; Javed, Umair: A review of time-series anomaly detection techniques: A step to future perspectives. En: *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1* Springer, 2021, p. 865–877

[82] Silverman, Bernard W.: *Density estimation for statistics and data analysis*. Vol. 26. CRC press, 1986

[83] Soh, Youngsung ; Hae, Yongsuk ; Mehmood, Aamer ; Ashraf, R H. ; Kim, Intaek [u. a.]: Performance evaluation of various functions for kernel density estimation. En: *Open J Appl Sci* 3 (2013), Nr. 1, p. 58–64

[84] Stadelmann, Thilo ; Amirian, Mohammadreza ; Arabaci, Ismail ; Arnold, Marek ; Duivesteijn, Gilbert F. ; Elezi, Ismail ; Geiger, Melanie ; Lörwald, Stefan ; Meier, Benjamin B. ; Rombach, Katharina [u. a.]: Deep learning in the wild. En: *Artificial Neural Networks in Pattern Recognition: 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19–21, 2018, Proceedings 8* Springer, 2018, p. 17–38

[85] Steinwart, Ingo ; Hush, Don ; Scovel, Clint: A Classification Framework for Anomaly Detection. En: *Journal of Machine Learning Research* 6 (2005), Nr. 2

[86] Sun, Jiayu ; Wang, Xinzhou ; Xiong, Naixue ; Shao, Jie: Learning sparse representation with variational auto-encoder for anomaly detection. En: *IEEE Access* 6 (2018), p. 33353–33361

[87] Tan, Pang-Ning ; Steinbach, Michael ; Kumar, Vipin: *Introduction to data mining*. Pearson Education India, 2016

[88] Useche, Diego H. ; Bustos-Brinez, Oscar A. ; Gallego, Joseph A. ; González, Fabio A.: Computing expectation values of adaptive Fourier density matrices for quantum anomaly detection in NISQ devices. 2022

[89] Walczak, Steven: Artificial neural networks. En: *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction*. IGI global, 2019, p. 40–53

[90] Warmuth, Manfred K. ; Kuzmin, Dima: Bayesian generalized probability calculus for density matrices. En: *Machine learning* 78 (2010), Nr. 1-2, p. 63

[91] Weglarczyk, Stanisław: Kernel density estimation and its application. En: *ITM Web of Conferences* Vol. 23 EDP Sciences, 2018, p. 00037

[92] Yang, Jie ; Xu, Ruijie ; Qi, Zhiquan ; Shi, Yong: Visual anomaly detection for images: A systematic survey. En: *Procedia Computer Science* 199 (2022), p. 471–478

[93] Zenati, Houssam ; Romain, Manon ; Foo, Chuan-Sheng ; Lecouat, Bruno ; Chandrasekhar, Vijay: Adversarially learned anomaly detection. En: *2018 IEEE International conference on data mining (ICDM)* IEEE, 2018, p. 727–736

[94] Zhai, Shuangfei ; Cheng, Yu ; Lu, Weining ; Zhang, Zhongfei: Deep structured energy based models for anomaly detection. En: *International conference on machine learning* PMLR, 2016, p. 1100–1109

[95] Zhao, Yue ; Nasrullah, Zain ; Li, Zheng: PyOD: A Python Toolbox for Scalable Outlier Detection. En: *Journal of Machine Learning Research* 20 (2019), Nr. 96, p. 1–7

[96] Zhou, Fangrong ; Wen, Gang ; Ma, Yi ; Geng, Hao ; Huang, Ran ; Pei, Ling ; Yu, Wenxian ; Chu, Lei ; Qiu, Robert: A Comprehensive Survey for Deep-Learning-Based Abnormality Detection in Smart Grids with Multimodal Image Data. En: *Applied Sciences* 12 (2022), Nr. 11, p. 5336