GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GENERACIÓN SINTÉTICA DE DATOS PARA MODELOS DE MACHINE LEARNING ORIENTADOS A LA CARACTERIZACIÓN DIELÉCTRICA DE LÍQUIDOS MEDIANTE UN RESONADOR DIELÉCTRICO

Autor: Javier Villacampa Porta

Directores: Miguel Monteagudo Honrubia

Francisco Javier Herraiz Martínez

Madrid – Junio 2023

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

"Generación sintética de datos para modelos de Machine Learning

orientados a la caracterización dieléctrica de líquidos

mediante un resonador dieléctrico"

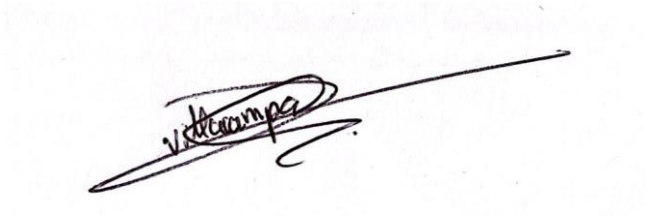
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2022/23 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

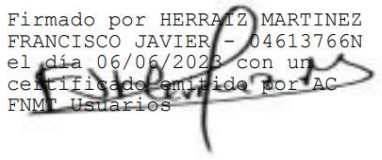
tomada de otros documentos está debidamente referenciada.



Fdo.: Javier Villacampa Porta Fecha: …04…/ …06…/ …2023…

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Miguel Monteagudo Honrubia, Francisco Javier Herraiz Martínez

Fecha: …05…/ …06…/ …2023…

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GENERACIÓN SINTÉTICA DE DATOS PARA MODELOS DE MACHINE LEARNING ORIENTADOS A LA CARACTERIZACIÓN DIELÉCTRICA DE LÍQUIDOS MEDIANTE UN RESONADOR DIELÉCTRICO

Autor: Javier Villacampa Porta

Directores: Miguel Monteagudo Honrubia

Francisco Javier Herraiz Martínez

Madrid – Junio 2023

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este Trabajo Final de Grado. Sin su apoyo y orientación, este proyecto hubiera sido considerablemente más desafiante.

En primer lugar, quiero agradecer a mi familia. En particular, a mis padres, por su confianza en mí, su paciencia ilimitada y el inmenso amor que siempre me han brindado. Su apoyo constante ha sido la base que me ha permitido alcanzar este hito en mi vida académica.

También quiero agradecer a mis directores de proyecto, Miguel y Javier. Su orientación y dedicación a la excelencia académica me han permitido comprender y abordar mejor los desafíos presentados en este trabajo. Su disposición constante para ayudar a resolver mis dudas ha sido esencial para la realización de este proyecto.

Por último, pero no menos importante, deseo agradecer a Anaí, Dani y Alfonso por su apoyo, interés y contribuciones constructivas a mi Trabajo Final de Grado. Su retroalimentación y asistencia desinteresada han sido determinantes para mejorar la calidad de mi trabajo y para mitigar posibles errores. Su apoyo y disponibilidad en todas las circunstancias han sido un verdadero regalo.

GENERACIÓN SINTÉTICA DE DATOS PARA MODELOS DE MACHINE LEARNING ORIENTADOS A LA CARACTERIZACIÓN DIELÉCTRICA DE LÍQUIDOS MEDIANTE UN RESONADOR DIELÉCTRICO.

Autor: Villacampa Porta, Javier.
Directores: Monteagudo Honrubia, Miguel y Herraiz Martínez, Francisco Javier
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este estudio, se evaluaron y compararon dos modelos generativos profundos, concretamente TVAE y CTGAN, para la generación sintética de datos aplicada a la mejora de modelos de ML para la caracterización de disoluciones de compuestos orgánicos. Los resultados mostraron que el TVAE superó al CTGAN en rendimiento en computación, similitud y utilidad, destacándose por su eficiencia computacional y su capacidad para generar datos sintéticos de alta calidad y utilidad para las tareas de clasificación y predicción.

Palabras clave: Aprendizaje Profundo, Sensor basado en resonador dieléctrico, SDG, VAE, GAN

1. Introducción

En los últimos años, el aprendizaje automático (ML) ha alcanzado notables avances como rama de la inteligencia artificial (IA), en particular, el aprendizaje profundo (DL) que ha impulsado innovaciones en diversas áreas como el análisis de imágenes, el reconocimiento facial y el reconocimiento de voz. No obstante, su aplicación en el campo de los biosensores se encuentra en una fase incipiente [1]–[3], a pesar de su potencial para mejorar la eficacia de dichos dispositivos. En este marco, este estudio busca optimizar la caracterización de disoluciones de compuestos orgánicos usando sensores de microondas (MW) basados en resonadores dieléctricos. Estos dispositivos presentan un gran potencial para la caracterización y clasificación de dichas disoluciones, dada su rentabilidad, baja complejidad y versatilidad. No obstante, la formación de modelos de clasificación para estas disoluciones conlleva un incremento significativo en los costes asociados al entrenamiento, dada la necesidad de generar conjuntos de datos de suficiente tamaño para disoluciones de compuestos orgánicos más complejos. Para mitigar este desafío, la generación de datos sintéticos y en especial el uso de modelos generativos profundos se presenta como una solución prometedora, de cara a producir un conjunto de datos de suficiente envergadura que permita mejorar los modelos de

caracterización de dichas disoluciones. Este trabajo se centrará en explorar y validar esta propuesta.

2. Definición del proyecto

Este proyecto tiene como objetivo principal explorar y aplicar técnicas de generación de datos sintéticos (SDG) mediante el uso de modelos generativos profundos. Se busca desarrollar un conjunto de datos robusto y fiable que represente una muestra de disoluciones de compuestos orgánicos, en el ámbito de los sensores MW que emplean resonadores dieléctricos. Entre los objetivos específicos se encuentran: realizar un análisis detallado de las técnicas de DL, desarrollar y optimizar un modelo de DL capaz de producir datos sintéticos indiferenciables de los datos reales, evaluar y analizar la eficacia y eficiencia de las técnicas SDG, y comprobar la aplicabilidad de los datos sintéticos generados en la identificación y categorización de disoluciones de compuestos orgánicos. La metodología empleada implicó una extensa revisión bibliográfica, la recopilación y preparación de datos reales, el entrenamiento y validación de los modelos, y la generación y validación de los datos sintéticos.

3. Descripción del modelo

En el desarrollo del modelo de este estudio, se emplearon dos conjuntos de datos, derivados de experimentos realizados con disoluciones de glicerina en concentraciones que varían del 0% al 80%. Los datos fueron adquiridos a través de dos métodos distintos: un Analizador Vectorial de Redes (VNA), un instrumento de laboratorio de alta complejidad y coste; y un Lector electrónico de bajo coste diseñado por la Universidad [4].

El conjunto de datos del VNA consiste en 100 espectros recogidos para cada concentración de la disolución de glicerina, mientras que el conjunto de datos del Lector electrónico contiene 180 señales en el dominio del tiempo para cada disolución, obtenidas a partir de 35 muestras de gotas, con cinco a seis repeticiones por muestra y un intervalo de seis segundos entre cada repetición.

Debido a la alta dimensionalidad de los datos espectrales recogidos, se aplicó un Análisis de Componentes Principales (PCA) para reducir características y maximizar la variabilidad de los datos. Como resultado, se obtuvieron dos conjuntos de datos tabulares, uno relacionado con el VNA y otro con el Lector electrónico.

Para garantizar un aprendizaje equilibrado de los modelos generativos, se mantuvo un balance adecuado entre todas las clases. El estudio se centró en la generación de datos sintéticos utilizando dos modelos principales de la biblioteca de DataCebo, SDV: CTGAN y TVAE [5]. Se implementaron dos estrategias de entrenamiento para estos modelos: una con todas las disoluciones simultáneamente y otra con un modelo individual entrenado para cada disolución.

Además, se llevó a cabo una afinación de hiperparámetros utilizando optimización bayesiana, con el objetivo de mejorar el rendimiento y desempeño de ambos modelos. Este enfoque sistemático para la generación de datos sintéticos y la optimización de hiperparámetros tuvo como objetivo explorar la factibilidad de estas técnicas para mejorar la caracterización de disoluciones de compuestos orgánicos.

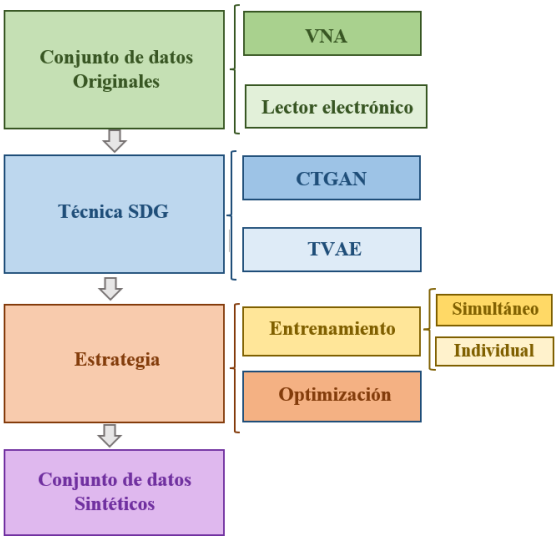


Figura 1: *Esquema de trabajo.*

4. Resultados

Los resultados del análisis comparativo entre los modelos TVAE y CTGAN para la generación de datos sintéticos revelaron que los VAEs como TVAE son más adecuados para la generación de datos sintéticos en este contexto. Su estabilidad, capacidad para reconstruir datos y manipular el espacio latente, además de la reducción de la varianza y la borrosidad, podrían contribuir a su superioridad. Las conclusiones extraídas de este análisis fueron:

- El entrenamiento individual de ambos modelos generativos para cada una de las disoluciones mostró resultados superiores al entrenamiento simultáneo en términos de calidad de los datos sintéticos generados.

- La aplicación de ambos modelos a los conjuntos de datos del VNA y del Lector electrónico demostró la viabilidad de estos para mejorar los modelos de caracterización de las disoluciones. Aunque ambos modelos mostraron resultados prometedores, el modelo TVAE superó al modelo CTGAN.
- Ambos modelos tuvieron un rendimiento similar en computación, pero TVAE fue más eficiente en términos de uso de memoria y tiempo de entrenamiento y generación de los datos sintéticos (Tabla 1).

Rendimiento en computación			
	Memoria	CPU	Tiempo
TVAE	1	1	1
CTGAN	2.5	1.2	3

Tabla 1: *Comparativa de rendimiento en computación.*

- En términos de similitud de los datos generados, el modelo TVAE demostró una superioridad sobre el CTGAN, reflejando mejor las dependencias en los datos y capturando de manera más efectiva los patrones circulares presentes en los datos originales del Lector electrónico (Tabla 2).

Similitud		
	Columns Shapes	Column Pair Trends
TVAE	98.16%	97.16%
CTGAN	96.09%	95.49%

Tabla 2: *Comparativa de los modelos en similitud.*

- Finalmente, en términos de utilidad, los datos sintéticos generados por el TVAE demostraron ser más útiles para las tareas de clasificación y predicción, ayudando a mejorar la precisión del modelo de clasificación y a disminuir el error del modelo de regresión (Tabla 3).

Utilidad		
	Precisión SVM	RMSE SVR
Original	90.7%	2.11
TVAE	**95.8%**	**2.075**
CTGAN	87.4%	8.654

Tabla 3: *Comparativa de los modelos en utilidad.*

5. Conclusiones

El estudio abordó el uso de técnicas de DL para mejorar la caracterización de disoluciones de compuestos orgánicos. Se utilizaron modelos generativos profundos para generar datos sintéticos que permitieron formar un conjunto de datos más robusto y representativo de muestras de distintas concentraciones de disoluciones de glicerina. La investigación mostró el potencial de modelos como los VAEs y las GANs en la generación de datos sintéticos de alta calidad. Un enfoque prometedor fue el uso de Keras para diseñar un VAE, el cual presentó ventajas significativas en rendimiento y flexibilidad respecto al modelo presentado en SDV. Los datos sintéticos resultantes permitieron mejorar la precisión y eficiencia de los métodos de caracterización de disoluciones orgánicos. En líneas de investigación futuras, se contempla la optimización del VAE en Keras, la exploración de otras técnicas de SDG y la expansión de la metodología a otros compuestos orgánicos, extendiendo así el alcance y aplicabilidad de los hallazgos de este estudio.

6. Referencias

[1] F. Cui, Y. Yue, Y. Zhang, Z. Zhang, y H. S. Zhou, «Advancing Biosensors with Machine Learning», *ACS Sens*, vol. 5, n.º 11, pp. 3346-3364, nov. 2020, doi: 10.1021/acssensors.0c01424.

[2] M. Monteagudo Honrubia, J. Matanza Domingo, F. J. Herraiz-Martínez, y R. Giannetti, «Low-Cost Electronics for Automatic Classification and Permittivity Estimation of Glycerin Solutions Using a Dielectric Resonator Sensor and Machine Learning Techniques», *Sensors*, vol. 23, n.º 8, p. 3940, abr. 2023, doi: 10.3390/s23083940.

[3] K. E. Schackart y J.-Y. Yoon, «Machine Learning Enhances the Performance of Bioreceptor-Free Biosensors», *Sensors*, vol. 21, n.º 16, p. 5519, ago. 2021, doi: 10.3390/s21165519.

[4] M. Monteagudo Honrubia, T. Ul Haq, B. Ali Fraea Esmail, J. Matanza Domingo, F. Javier Herraiz-Martínez, y R. Giannetti, «Low-Cost Electronics for Automatic Classification and Permittivity Estimation of Glycerin Solutions Using a Dielectric Resonator Sensor and Machine Learning Techniques», Sensors 2023, Vol. 23, Page 3940, vol. 23, n.o 8, p. 3940, abr. 2023, doi: 10.3390/S23083940.

[5] L. Xu, M. Skoularidou, A. Cuesta-Infante, y K. Veeramachaneni, «Modeling Tabular Data using Conditional GAN», 33rd Conference on Neural Information Processing Systems, pp. 3-6, oct. 2019, Accedido: 15 de mayo de 2023. [En línea]. Disponible en: https://github.com/DAI-Lab/CTGAN

SYNTHETIC DATA GENERATION FOR MACHINE LEARNING MODELS ORIENTED TO THE DIELECTRIC CHARACTERIZATION OF LIQUIDS USING A DIELECTRIC RESONATOR.

Author: Villacampa Porta, Javier.
Supervisors: Monteagudo Honrubia, Miguel and Herraiz Martínez, Francisco Javier.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

In this study, two deep generative models, namely TVAE and CTGAN, were evaluated and compared for synthetic data generation applied to the improvement of ML models for the characterization of organic compound dissolutions. The results showed that TVAE outperformed CTGAN in computational performance, similarity, and utility, standing out for its computational efficiency and its ability to generate synthetic data of high quality and utility for classification and prediction tasks.

Keywords: Deep Learning, Dielectric Resonator-Based Sensor, SDG, VAE, GAN

1. Introduction.

In recent years, machine learning (ML) has achieved remarkable advances as a branch of artificial intelligence (AI), in particular, deep learning (DL) that has driven innovations in various areas such as image analysis, facial recognition, and speech recognition. However, its application in the field of biosensors is at a nascent stage [1]-[3], despite its potential to improve the efficiency of such devices. In this framework, this study seeks to optimize the characterization of organic compound solutions using microwave (MW) sensors based on dielectric resonators. These devices present an immense potential for the characterization and classification of such solutions, given their cost-effectiveness, low complexity, and versatility. However, training classification models for these dissolutions leads to a significant increase in the costs associated with training, given the need to generate data sets of sufficient size for more complex organic compound dissolutions. To mitigate this challenge, the generation of synthetic data and in particular the use of deep generative models presents itself as a promising solution, with the goal of producing a dataset of sufficient size to enable improved characterization models for these complex organic compound solutions. This work will focus on exploring and validating this proposal.

2. Project definition

The main objective of this work is to explore and apply synthetic data generation techniques using deep generative models. The aim is to develop a robust and reliable data set representing a sample of organic compound dissolutions, in the field of microwave sensors using dielectric resonators. Specific objectives include: to perform a detailed analysis of deep learning techniques, to develop and optimize a deep learning model capable of producing synthetic data indistinguishable from real data, evaluating and analyse the effectiveness and efficiency of synthetic data generation techniques, and testing the applicability of the generated synthetic data in the identification and categorization of organic compounds. The methodology employed involved an extensive literature review, the collection and preparation of real data, the training and validation of models, and the generation and validation of synthetic data.

3. Model description

In developing the model for this study, two sets of data were used, derived from experiments performed with glycerol solutions at concentrations ranging from 0% to 80%. The data were acquired through two different methods: a Vector Network Analyzer (VNA), a laboratory instrument of high complexity and cost; and a low-cost electronic Reader designed by the University [4].

The VNA data set consists of 100 spectra collected for each concentration of the glycerol dilution, while the electronic Reader data set contains 180 time domain signals for each dilution obtained from 35 droplet samples, with five to six repetitions per sample and a six-second interval between each repetition.

Due to the high dimensionality of the collected spectral data, a Principal Component Analysis (PCA) was applied to reduce features and maximize the variability of the data. As a result, two tabular data sets were obtained, one related to the VNA and the other to the e-Reader.

To ensure balanced learning of the generative models, an appropriate balance between all classes was maintained. The study focused on synthetic data generation using two main models from the DataCebo library, SDV: CTGAN and TVAE [5]. Two training strategies were implemented for these models: one with all solvents simultaneously and one with an individual model trained for each solvent.

In addition, hyperparameter tuning was conducted using Bayesian optimization, with the aim of improving the throughput and performance of both models. This systematic approach to synthetic data generation and hyperparameter optimization aimed to explore the feasibility of these techniques to improve the characterization of organic compound solutions.

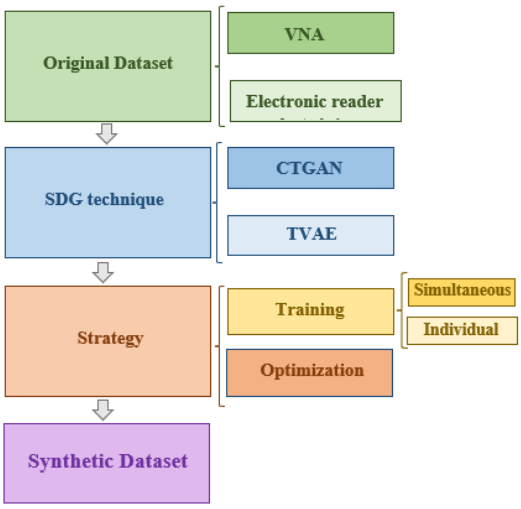


Figure 1: *Scheme of work.*

5. Results

The results of the comparative analysis between TVAE and CTGAN models for synthetic data generation revealed that VAEs such as TVAE are more suitable for synthetic data generation. Their stability, ability to reconstruct data and manipulate latent space, in addition to reduced variance and fuzziness, could contribute to their superiority. The conclusions drawn from this analysis were:

- Individual training of both generative models for each of the dissolutions showed superior results to simultaneous training in terms of quality of the synthetic data generated.
- Both models had similar computational performance, but TVAE was more efficient in terms of memory usage and time to train and generate the synthetic data (Table 1).

Performance			
	Memory	CPU	Time
TVAE	1	1	1
CTGAN	2.5	1.2	3

Table 1: *Comparison in computational performance*

- The application of both models to the VNA and eReader datasets demonstrated the feasibility of these for improving the dissolution characterization models. Although both models showed promising results, the TVAE model outperformed the CTGAN model.
- In terms of similarity of the generated data, the TVAE model demonstrated superiority over CTGAN, better reflecting the dependencies in the data and more effectively capturing the circular patterns present in the original eReader data (Table 2).

	Similarity	
	Columns Shapes	Column Pair Trends
TVAE	98.16%	97.16%
CTGAN	96.09%	95.49%

Table 2: *Comparison in similarity of the models*

- Finally, in terms of utility, the synthetic data generated by the TVAE proved to be more useful for classification and prediction tasks, helping to improve the accuracy of the classification model and decrease the error of the regression model (Table 3).

	Utility	
	Accuracy SVM	RMSE SVR
Original	90.7%	2.11
TVAE	95.8%	2.075
CTGAN	87.4%	8.654

Table 3: *Comparison in similarity of the models*

6. Conclusions

The study addressed the use of DL techniques to improve the characterization of organic compound solutions. Deep generative models were used to generate synthetic data to form a more robust and representative data set for samples of different concentrations of glycerol solutions. The research showed the potential of models such as VAEs and GANs in generating high quality synthetic data. One promising approach was the use of Keras to design a VAE, which presented significant advantages in performance and flexibility over the model presented in SDV. The resulting synthetic data allowed improving the accuracy and efficiency of organic solution characterization methods. In future lines of research, the optimization of the VAE in Keras, the exploration of other SDG techniques and the expansion of the methodology to other

organic compounds are contemplated, thus extending the scope and applicability of the findings of this study.

7. References

[1] F. Cui, Y. Yue, Y. Zhang, Z. Zhang, y H. S. Zhou, «Advancing Biosensors with Machine Learning», *ACS Sens*, vol. 5, n.º 11, pp. 3346-3364, nov. 2020, doi: 10.1021/acssensors.0c01424.

[2] M. Monteagudo Honrubia, J. Matanza Domingo, F. J. Herraiz-Martínez, y R. Giannetti, «Low-Cost Electronics for Automatic Classification and Permittivity Estimation of Glycerin Solutions Using a Dielectric Resonator Sensor and Machine Learning Techniques», *Sensors*, vol. 23, n.º 8, p. 3940, apr. 2023, doi: 10.3390/s23083940.

[3] K. E. Schackart y J.-Y. Yoon, «Machine Learning Enhances the Performance of Bioreceptor-Free Biosensors», *Sensors*, vol. 21, n.º 16, p. 5519, aug. 2021, doi: 10.3390/s21165519.

[4] M. Monteagudo Honrubia, T. Ul Haq, B. Ali Fraea Esmail, J. Matanza Domingo, F. Javier Herraiz-Martínez, y R. Giannetti, «Low-Cost Electronics for Automatic Classification and Permittivity Estimation of Glycerin Solutions Using a Dielectric Resonator Sensor and Machine Learning Techniques», Sensors 2023, Vol. 23, Page 3940, vol. 23, n.º 8, p. 3940, apr. 2023, doi: 10.3390/S23083940.

[5] L. Xu, M. Skoularidou, A. Cuesta-Infante, y K. Veeramachaneni, «Modeling Tabular Data using Conditional GAN», 33rd Conference on Neural Information Processing Systems, pp. 3-6, oct. 2019, Accessed: 15 May 2023. [Online]. Available from: https://github.com/DAI-Lab/CTGAN

Índice

Índice de ilustraciones

Índice de tablas

CAPITULO 1. Introducción

En el transcurso de la última década, la disciplina del aprendizaje automático (ML, por sus siglas en inglés), una rama de la inteligencia artificial (IA), ha experimentado un progreso impresionante. La incursión de métodos más avanzados, como el aprendizaje profundo (DL, por sus siglas en inglés), ha llevado a innovaciones en múltiples dominios como el análisis de imágenes, reconocimiento facial y reconocimiento de voz, por nombrar algunos. No obstante, la incorporación de dichas técnicas en el sector de los biosensores se encuentra en una fase incipiente, pese a que investigaciones recientes sugieren que determinados modelos de ML poseen una potencialidad considerable y podrían ser una herramienta esencial para impulsar la eficacia de los biosensores [1].

Uno de los desafíos más acuciantes que la comunidad de biosensores afronta es la carencia de conjuntos de datos suficientemente grandes para entrenar los modelos. Los algoritmos de aprendizaje supervisado requieren vastas bases de datos etiquetadas para su formación. Estos conjuntos de datos suelen incluir clases minoritarias, con proporciones que podrían llegar a ser de 1000 a 1, lo que podría llevar a que los modelos de aprendizaje omitan estas clases minoritarias o fallen en la generalización de sus características esenciales [4]. Si bien los avances recientes en las redes neuronales profundas han catalizado un progreso significativo en tareas de aprendizaje supervisado con conjuntos de datos de imágenes, el entrenamiento de estas redes exige enormes volúmenes de datos etiquetados, lo que ha propiciado la aparición de métodos para generar el volumen de datos requerido.

Este estudio se enfoca en optimizar la caracterización de disoluciones de compuestos orgánicos mediante el uso de sensores de microondas (MW) basados en resonadores dieléctricos. Las más recientes revisiones bibliográficas [1]–[3] sugieren que, a pesar del notable avance en la aplicación de técnicas de ML en el campo de los biosensores, existe un déficit de conocimiento en el uso de estos métodos para sensores de MW basados en resonadores dieléctricos. Esta limitación puede deberse a la falta de especificidad, que representa un obstáculo considerable para las aplicaciones biomédicas. No obstante, se ha postulado que este inconveniente podría ser atenuado mediante la aplicación de técnicas de ML para aumentar la sensibilidad, lo que

permitiría desarrollar dispositivos de detección capaces de identificar mínimas variaciones en la permitividad [5].

El presente trabajo se inserta en este ámbito de investigación y se enfoca en abordar el desafío palpable asociado con el coste de usar muestras de disoluciones de compuestos más sofisticados para su ulterior caracterización. Para contrarrestar este obstáculo, este estudio empleará modelos generativos profundos, como las redes generativas adversarias (GAN) y los *autoencoders* variacionales (VAE), con el objetivo de producir un volumen considerable de datos sintéticos que permitan un entrenamiento eficaz de los modelos de ML implementados.

CAPITULO 2. Definición del trabajo

2.1. Justificación

Hoy en día las técnicas de ML son cada vez más utilizadas en todos los sectores. En el ámbito de los sensores MW esta tecnología se encuentra en auge y ya se ha utilizado para modelar muchos dispositivos; sin embargo, el objetivo de este trabajo es aplicar una rama concreta del DL, los modelos generativos profundos, para estudiar la viabilidad de generar un conjunto de datos sintéticos sobre los datos recopilados de una serie de disoluciones orgánicos, usando sensores MW basados en resonadores dieléctricos.

El uso de sensores MW basados en resonadores dieléctricos se presenta como una tecnología prometedora para caracterizar y clasificar disoluciones de compuestos orgánicos, en términos de rentabilidad, baja complejidad y versatilidad. Dentro de este área, se busca seguir entrenando modelos de clasificación para caracterizar distintas disoluciones de compuestos orgánicos como la glicerina o el PEG. Sin embargo, generar un conjunto de datos suficientemente grande sobre disoluciones de compuestos complejos supone un incremento considerable en el coste del entrenamiento. Para intentar solventar esta problemática, en este trabajo se propone el uso de técnicas de generación de datos sintéticos (SDG), con el objetivo de obtener un conjunto de datos suficientemente grande para mejorar los modelos de caracterización de una serie de disoluciones de compuestos orgánicos complejos.

2.2. Objetivos

Acorde con la justificación presentada, se definieron para el desarrollo de este estudio los siguientes objetivos:

Objetivo General:

1. Explorar y poner en práctica técnicas de SDG mediante el uso de modelos generativos profundos para crear un conjunto de datos robusto y fiable que represente una muestra de disoluciones de compuestos orgánicos, en el ámbito de los sensores MW basados en resonadores dieléctricos.

Objetivos Específicos:

2. Desarrollar un análisis en profundidad de las técnicas de DL, con particular énfasis en los modelos generativos profundos, y su aplicabilidad en la creación de datos sintéticos.
3. Evaluar y analizar la efectividad y eficiencia de las técnicas de SDG y contrastarlas con los métodos tradicionales de recopilación de datos.
4. Construir y perfeccionar un modelo de DL que sea capaz de producir datos sintéticos que sean indiferenciables de los datos reales recopilados a través de sensores de MW basados en resonadores dieléctricos.
5. Comprobar la aplicabilidad de los datos sintéticos generados en la identificación y categorización de disoluciones de compuestos orgánicos, haciendo uso de modelos de clasificación y regresión ya existentes.

2.3.Metodología y Planificación

Acorde con los objetivos descritos este estudio se ha llevado a cabo en varias etapas bien definidas, con el fin de investigar y aplicar técnicas de SDG mediante el uso de modelos generativos profundos para crear un conjunto de datos robusto y fiable que permita mejorar los modelos de ML de caracterización de disoluciones de compuestos biológicos.

En primer lugar, se llevó a cabo una extensa revisión de la literatura existente para comprender en profundidad el funcionamiento de los modelos generativos profundos, más concretamente los VAE y GAN, y su uso en la generación de datos sintéticos. Una vez se hubieron identificado los métodos y recursos a utilizar, la siguiente etapa consistió en recopilar y preparar conjuntos de datos reales. Para este estudio se utilizaron dos conjuntos de datos provenientes de experimentos realizados con aire y disoluciones de glicerina a concentraciones variables, desde el 0% hasta el 80%. El primer conjunto de datos obtenido haciendo uso de un VNA, instrumento de laboratorio de alta complejidad y coste, incluía los espectros para cada concentración de disolución de glicerina. Por otro lado, el segundo conjunto de datos incluía señales de estas mismas disoluciones obtenidas a través de un Lector electrónico, un equipo electrónico de bajo coste diseñado por la Universidad.

En la segunda fase del estudio, se empleó el Análisis de Componentes Principales (PCA, por sus siglas en inglés) para reducir la alta dimensionalidad de los datos espectrales recopilados. Esta fase generó dos conjuntos de datos tabulares con dimensiones específicas para las mediciones del VNA y el Lector electrónico.

En la tercera fase, el estudio empleó técnicas de SDG para optimizar los modelos de caracterización de disoluciones de compuestos orgánicos. Se evaluaron varios modelos, en particular, CTGAN y TVAE, empleados para generar datos sintéticos condicionados a una etiqueta o variable de entrada.

Se implementaron diversas estrategias de entrenamiento y optimización de hiperparámetros para cada modelo, y se evaluó su rendimiento en dos escenarios diferentes, utilizando ambos conjuntos de datos. Este enfoque permitió una evaluación en profundidad de la eficacia y la viabilidad de estas técnicas para mejorar la caracterización de disoluciones de compuestos orgánicos.

Por último, el estudio incluyó una evaluación de los resultados generados por estas técnicas, basándose en múltiples dimensiones (privacidad, similitud, utilidad y dimensiones de rendimiento). Este análisis minucioso permitió la eliminación de modelos con rendimientos insatisfactorios y la elección de aquellos con un desempeño superior.

La ejecución de este estudio se realizó respetando la secuencialidad de las etapas descritas y garantizando un control riguroso de las variables y condiciones experimentales. La metodología se diseñó con la intención de optimizar los recursos disponibles, asegurar la replicabilidad del estudio y maximizar la validez de los resultados obtenidos.

Debido a la existencia de una fecha límite para realizar el desarrollo de este proyecto, el estudio se estructuró como se presenta en la **Tabla 1**:

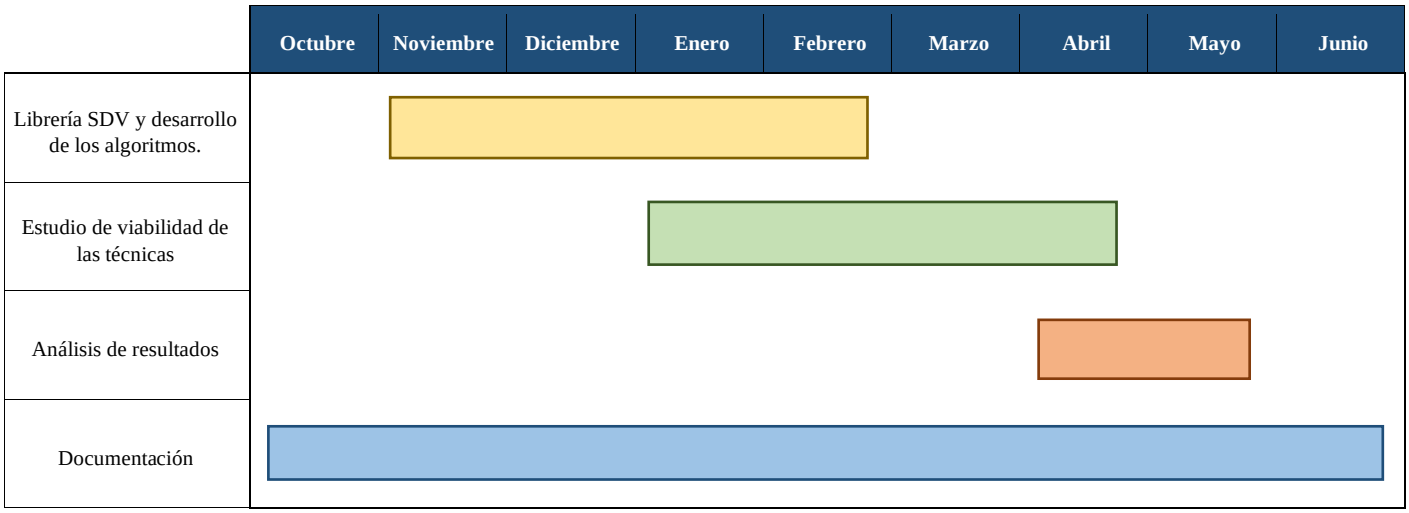


Tabla 1: *Diagrama de GANTT del cronograma del proyecto.*

CAPITULO 3. Estado de la cuestión

En este capítulo se detallarán los conceptos principales, los trabajos y soluciones que existen en el ámbito de este trabajo fin de grado, haciendo mayor hincapié en la tecnología principal de este desarrollo, las técnicas de SDG.

3.1 Sensores de microondas (MW)

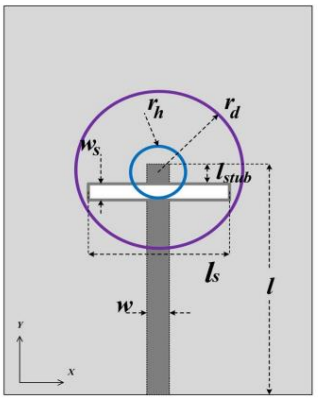
Los sensores MW se han erigido como soluciones rentables y versátiles para la identificación y cuantificación de sustancias [6]. Estos sensores, desde una óptica electrónica, pueden ser modelados como un circuito RLC de alta frecuencia con un elevado factor de calidad Q y son sensibles a las variaciones en la capacitancia, lo que conlleva a un cambio en la frecuencia de resonancia. La capacidad de estos sensores para identificar sustancias con permitividades superiores a las del aire, el medio de referencia común, ha propiciado investigaciones enfocadas en su potencial para detectar variaciones en la concentración de glicerina, analizando su efecto en la frecuencia de resonancia del dispositivo resonador dieléctrico (DR, por sus siglas en inglés) [6].

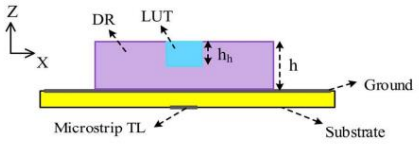
Los sensores DR, fabricados de manera sencilla a partir de un material dieléctrico masivo de geometría regular, normalmente cilíndrica, ofrecen beneficios significativos. Su alta eficiencia de radiación en una superficie 3D, acompañada de pérdidas de conducción mínimas, los convierten en dispositivos radiantes de microondas ideales [7], [8]. Se destacan también por su capacidad para convertirse en dispositivos de detección pasivos de bajo coste, eludiendo la necesidad de baterías [9]. No obstante, se han propuesto otras tecnologías resonantes para la detección de líquidos en la región de microondas, como los resonadores de anillo dividido (SRR), particularmente útiles en aplicaciones de microfluídica o inmersión [6], [10], [11]. Los resultados alcanzados señalan la aplicación potencial de estos sensores en áreas como el análisis químico, el control de calidad de alimentos, el diagnóstico biomédico, la monitorización medioambiental entre otras [12].

En este contexto, el sensor desarrollado por el Instituto de Investigación Tecnológica (I.I.T.) [13] ofrece una alternativa para la caracterización de disoluciones líquidas de compuestos. La

geometría de este sensor consiste en un DR cilíndrico, un material cerámico capaz de almacenar energía electromagnética a determinadas frecuencias, situado encima de una línea de transmisión *microstrip* [14]. La línea microstrip es una fina tira metálica sobre un sustrato dieléctrico y un plano de masa, capaz de transportar señales RF. El acoplamiento entre el DR y la línea microstrip se logra mediante una ranura en el plano de masa de la línea, que permite excitar sus modos resonantes [14] (ver Ilustración 1).

El DR posee un pequeño depósito en su superficie capaz de contener una gota de mezcla líquida, la cual afecta la permitividad y conductividad del DR, modificando así su frecuencia de resonancia y factor Q [12]. Los cambios en estos parámetros influyen en el coeficiente de reflexión entre el DR y la línea microstrip, lo cual puede medirse con un VNA.





Variable	Dimenssion [mm]
r_d	17
h	8
r_h	4
h_h	2.5
w	2.95
l	74
w_s	0.7
l_s	32
l_{stub}	6.65

Ilustración 1: *Diseño del sensor DR y sus especificaciones geométricas. Imagen tomada de [14]*

La calibración del sensor es crucial para obtener resultados precisos y valiosos. En este sentido, el ML, una rama de la IA, ha mostrado potencial para mejorar el rendimiento del sensor, especialmente en casos con una relación no lineal entre el proceso analizado y la señal adquirida [1], [5], [15]. La meta de estos estudios es aumentar la sensibilidad de los sensores DR mediante técnicas de ML para detectar correlaciones en los datos y realizar predicciones, mejorando así la detección de variaciones sutiles en la permitividad [5].

A pesar de estos avances, existen limitaciones significativas en la implementación de sensores DR debido a los sistemas de instrumentación electrónica para medidas de alta frecuencia [16].

Un ejemplo de ello es el VNA, que requiere una inversión económica muy considerable pudiendo llegar al medio millón de euros, un personal cualificado, una calibración compleja y cuyo tamaño dificulta su uso para medidas automáticas en tiempo real en entornos industriales. Estos factores también limitan la penetración de los dispositivos analíticos in situ, como la investigación ambiental [16] y dispositivos *Point of Care* (PoC) [17].

Para superar estas limitaciones, se propone la reducción del coste y el aumento de la portabilidad del equipo electrónico conectado a los sensores DR [13]. Gracias al desarrollo de tecnologías como el Internet de las Cosas (IoT) y el creciente interés por el hardware y software de código abierto, se han abierto nuevas posibilidades para desarrollar electrónica de bajo coste utilizando microcontroladores o microcomputadoras como Arduino o Raspberry Pi [18]. Estos dispositivos permiten el control de señales de entrada/salida (E/S) dentro de un sistema integrado y ofrecen ventajas de personalización, adaptándose a las necesidades del objetivo analítico.

Por lo tanto, una de las propuestas actuales propone usar un microcontrolador Arduino y componentes de electrónica de alta frecuencia económicos para reemplazar las medidas de un VNA a un precio más asequible, pero con un rendimiento comparable. Esta estrategia permitiría una mayor penetración de los sensores DR en diversas aplicaciones, superando las limitaciones económicas y de complejidad actuales, especialmente en países de bajos ingresos donde la necesidad de sensores de bajo coste es más crítica [19]. Es en este contexto donde se enmarca nuestro estudio, de tal manera que el objetivo será estudiar la viabilidad de aplicar técnicas SDG sobre datos extraídos a través de estos métodos.

3.2 Machine Learning y Deep Learning

El ML es una rama dentro de la IA, que se enfoca en el uso de algoritmos y modelos para analizar un conjunto de datos con el objetivo de extraer información valiosa y mejorar su rendimiento en una tarea determinada. Dentro de esta área existen distintos tipos de aprendizaje entre los que se destacan, el aprendizaje supervisado y el no supervisado [20].

- En el aprendizaje supervisado, se parte de un conjunto finito de categorías predefinidas y los datos se etiquetan, es decir, se sabe a qué categoría pertenecen los datos sin necesidad de entrenar la red [21].
- En el aprendizaje no supervisado, en cambio, no hay una tabla de clasificación fija y no se asigna ningún resultado a los datos. Lo que interesa es que la red descubra patrones similares que relacionen algunas de las entradas [22].

Este estudio se engloba dentro del marco del aprendizaje supervisado; puesto que para la creación del conjunto de datos se elegirán previamente las disoluciones de compuestos a medir.

El DL representa un subcampo dentro de la IA que se especializa en la implementación de algoritmos de ML, conocidos como redes neuronales artificiales, particularmente aquellas de múltiples capas o "profundas". Inspiradas en la estructura y funcionalidad del cerebro humano, estas redes neuronales son esencialmente sistemas de cálculo que aprenden a realizar tareas identificando y categorizando patrones en los conjuntos de datos de entrada.

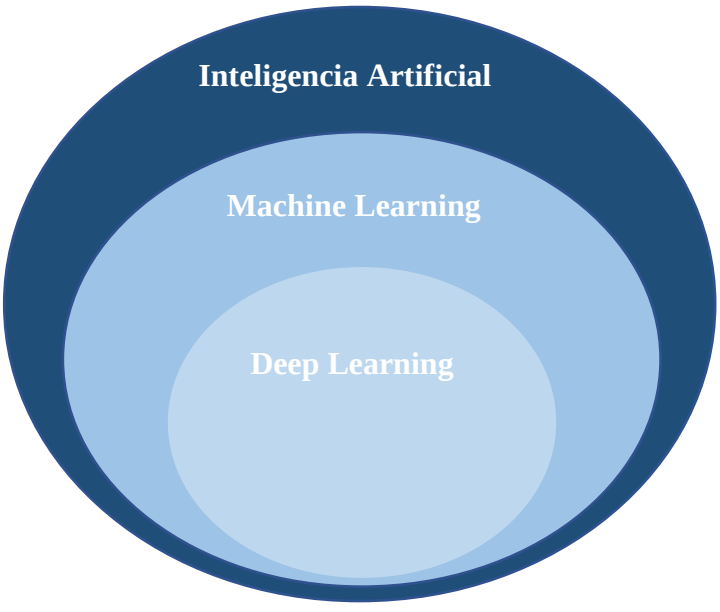


Ilustración 2: *Diagrama de IA, ML y DL.*

Fuente: Elaboración propia

Las redes neuronales artificiales se componen de una serie de capas jerárquicas, cada una de las cuales consta de una multitud de nodos o "neuronas". Cada neurona recibe entradas de las neuronas de la capa anterior, procesa esa información, y transfiere el resultado a las neuronas de la capa siguiente, generando representaciones cada vez más sofisticadas [23], [24]. Este proceso se repite sucesivamente a través de todas las capas de la red, desde la capa de entrada hasta la capa de salida. (ver Ilustración 3)

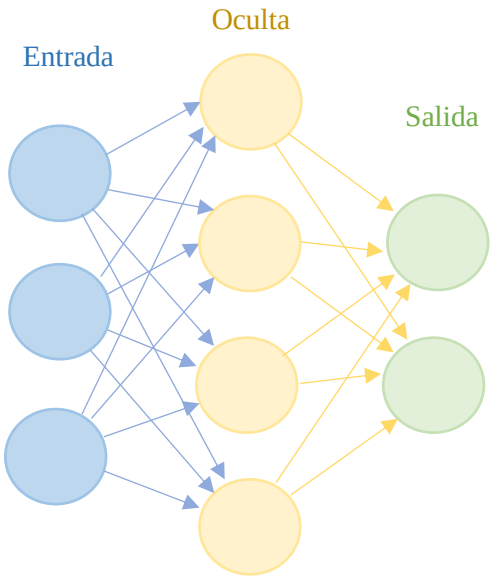


Ilustración 3: *Diagrama de red neuronal.*
Fuente: Elaboración propia.

Cada neurona se encuentra asociada a parámetros denominados pesos, que se ajustan durante el entrenamiento de la red para producir la salida deseada. Estos pesos, valores numéricos vinculados a cada conexión entre nodos de dos capas consecutivas, controlan la fuerza y la dirección de la influencia de cada nodo en la salida final. La actualización de estos pesos se lleva a cabo iterativamente mediante un algoritmo de optimización, como el descenso del gradiente, y se expresan matemáticamente mediante la multiplicación de un factor de aprendizaje por el gradiente de la función de coste con respecto a los pesos (1).

$$w(i,j) = w(i,j) - \alpha \frac{\partial E}{\partial w(i,j)} \quad (1)$$

Donde $w(i,j)$ es el peso que conecta el nodo i de la capa anterior con el nodo j de la capa siguiente, α es el factor de aprendizaje y $\frac{\partial E}{\partial w(i,j)}$ es el gradiente de la función de coste con respecto a ese peso.

En el procesamiento interno de cada neurona, se aplica una función de activación no lineal, que introduce no linealidad en la red y permite modelar relaciones complejas entre los datos de entrada y salida. Las funciones de activación más utilizadas son la sigmoide, ReLU y *softmax*, cada una con características y aplicaciones específicas.

La función sigmoide es comúnmente aplicada en problemas de clasificación por su suavidad, rango acotado y su interpretación probabilística [25], aunque presenta dificultades en redes profundas debido a la saturación en los extremos de su rango. La función ReLU, por otro lado, evita el problema del gradiente desvaneciente y es más eficiente computacionalmente, siendo comúnmente utilizada en redes profundas [26] [27]. Finalmente, la función *softmax*, usada en la última capa de una red para problemas de clasificación multiclase [28][29], transforma un vector de números reales en un vector de probabilidades.

Uno de los aspectos más característicos del DL es su capacidad para aprender características de alto nivel a partir de los datos de entrada. Durante el proceso de aprendizaje, las primeras capas de la red extraen características de bajo nivel, como líneas y colores en el caso de imágenes, y las capas posteriores combinan y transforman estas características de bajo nivel en características de más alto nivel, como formas y estructuras. Este proceso de aprendizaje jerárquico permite a las redes neuronales profundas modelar relaciones complejas y no lineales en los datos.

Las técnicas de DL han demostrado ser especialmente eficaces en una amplia variedad de tareas, como la clasificación y generación de imágenes, el reconocimiento de voz y texto, y la predicción en series de tiempo, entre otras. Sin embargo, también presentan retos significativos, incluyendo la necesidad de grandes cantidades de datos y potencia computacional para su entrenamiento, así como la interpretación de los modelos, que a menudo son considerados como "cajas negras" debido a la dificultad de entender cómo toman decisiones específicas.

3.3 Técnicas de generación de datos sintéticos (SDG)

Los datos sintéticos (SD, por sus siglas en inglés) son datos artificiales generados por un modelo entrenado o construido para replicar datos reales (RD, por sus siglas en inglés) basados en sus distribuciones (es decir, forma y varianza) y estructura (es decir, correlaciones entre los atributos). Los SD tienen dos casos de uso principales: (i) preservación de la privacidad: para permitir el intercambio seguro y privado de datos sensibles [30]; (ii) aumento de datos: para equilibrar conjuntos de datos o complementar los datos disponibles antes de entrenar un modelo de ML (el objetivo de este trabajo) [31]. Además, los SD pueden ser potencialmente utilizado para obtener conclusiones estadísticas o entrenar modelos de ML, evitando la divulgación de datos sensibles [32].

De cara a su adopción, las técnicas SDG deben evaluarse en términos de privacidad (riesgo de divulgación de datos personales), similitud (cuán bien representa los SD los datos reales), utilidad (usabilidad de las conclusiones estadísticas extraídas de los SD o los resultados de los modelos de ML entrenados con SD) y dimensiones de rendimiento (huella, tiempo de generación y recursos computacionales).

El objetivo de las técnicas SDG es aprender acerca de una distribución arbitraria de datos para luego ser capaces de generar nuevos ejemplares lo más similares posible a los originales, sin ser idénticos. Estás técnicas reconocen los patrones presentes en los datos de entrenamiento permitiendo a una máquina aprender de estos y, basándose en ellos, generar nuevos datos similares prácticamente idénticos a los originales [33].

Hoy en día, existe un gran abanico de métodos, algoritmos y modelos aplicados a la generación sintética de datos (Ilustración 4), en lo que a este trabajo se refiere se profundizará principalmente en los modelos generativos profundos como son las GANs y los VAEs [32], [34].

A parte de las técnicas basadas en DL que se desarrollarán más en profundidad posteriormente, existen otra serie de técnicas muy extendidas en SDG.

En primer lugar, el muestreo de *bootstrapping* es una técnica de remuestreo que implica tomar muestras repetidas con reemplazo de un conjunto de datos original. El *bootstrapping* permite generar múltiples conjuntos de datos sintéticos y estimar la distribución de estadísticas de interés, como medias y varianzas [35]. Este método es ampliamente aplicado en inferencia estadística, validación cruzada y construcción de intervalos de confianza.

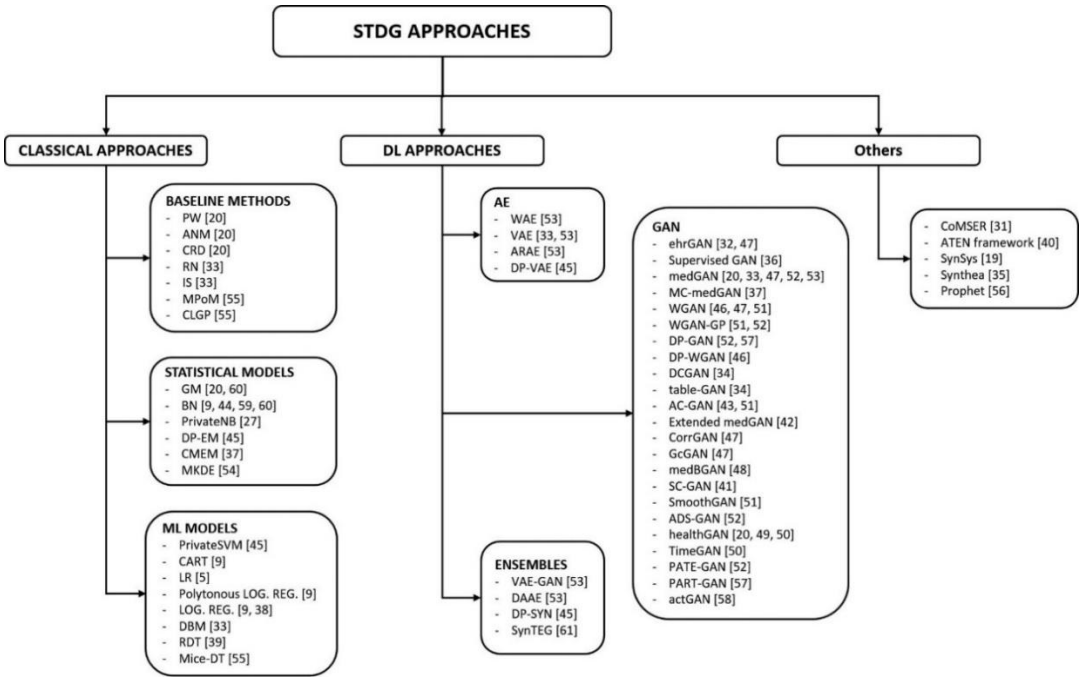


Ilustración 4: *Clasificación de técnicas SDG*
Imagen tomada de [32]

El método de Monte Carlo es una técnica numérica que se utiliza para resolver problemas mediante la generación de números aleatorios. Se basa en la generación de un gran número de muestras aleatorias a partir de una distribución de probabilidad conocida, que se utilizan para estimar la respuesta de un sistema complejo [36], [37]. Esta técnica se utiliza comúnmente en la generación de datos sintéticos para simular el comportamiento de un conjunto de datos en situaciones donde los datos reales no están disponibles o son limitados. Un ejemplo de esto se puede ver en su aplicación en el análisis de riesgos financieros, donde este método se utiliza para simular la variabilidad de los precios de los activos financieros y para estimar la probabilidad de pérdidas en una cartera de inversión [38].

Otra técnica presente en la literatura es el uso de una función cópula. Una cópula es una función matemática que describe la relación entre dos o más variables aleatorias sin especificar sus distribuciones marginales [39], y se utiliza para modelar la dependencia entre variables aleatorias. Las cópulas permiten generar datos sintéticos que conserven la estructura de dependencia de un conjunto de datos real, como por ejemplo el modelado de la dependencia entre los precios de los activos financieros o para simular escenarios de pérdida en una cartera de inversión [40].

Finalmente, los modelos de Markov de orden variable (VOM) son modelos probabilísticos que describen la dependencia entre elementos en una secuencia, como caracteres en texto o eventos en series temporales. Los VOM capturan la probabilidad de transición entre estados en función de la historia observada y pueden utilizarse para generar secuencias sintéticas que imitan las propiedades de la secuencia original [41]. Son especialmente útiles en el análisis de secuencias genéticas, generación de texto y modelado de series temporales.

En resumen, durante la última década, se ha logrado generar datos sintéticos mediante la modelización de una distribución de probabilidad multivariante conjunta para un conjunto de datos y luego el muestreo de esa distribución. Los conjuntos de datos complicados han requerido distribuciones bastante complejas como las explicadas previamente, por ejemplo, una secuencia de eventos podría haber sido modelada utilizando modelos VOM, o un conjunto de variables correlacionadas no linealmente podría haber sido modelado utilizando cópulas. Sin embargo, estos modelos generativos están restringidos por el tipo de funciones de distribución disponibles para los usuarios, limitando severamente las representaciones que se pueden utilizar para crear modelos generativos y limitando posteriormente la fidelidad de los datos sintéticos. Es por esto por lo que, con el avance del aprendizaje profundo y la capacidad de computación, durante los últimos años se han diseñado múltiples modelos generativos (Ilustración 5), entre los que destacan especialmente las GANs y los VAEs.

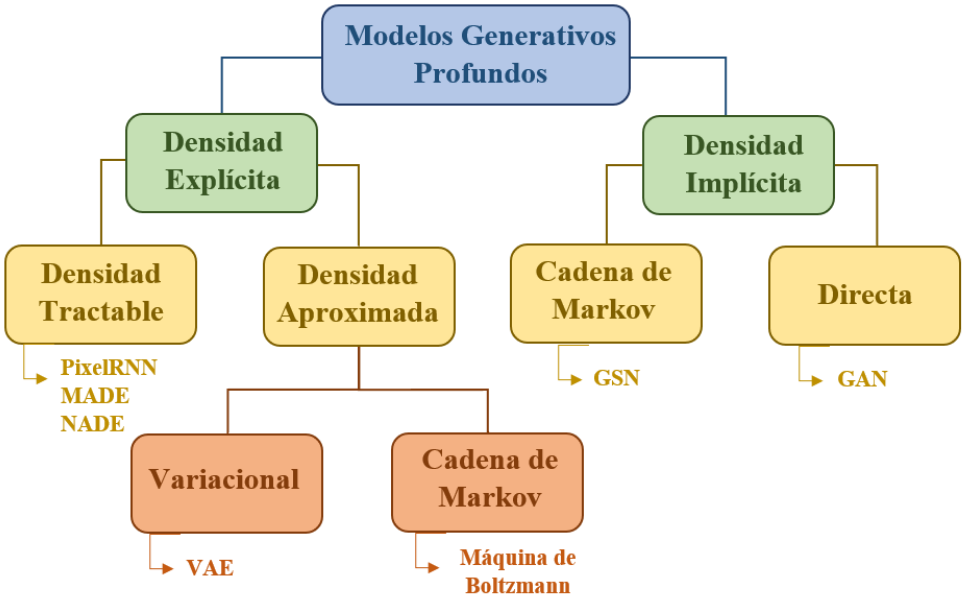


Ilustración 5: *Taxonomía de modelos generativos profundos*
Imagen basada en [42]

3.3.1 Autoencoders

Un *autoencoder* es un tipo de red neuronal artificial que se utiliza para aprender una representación de baja dimensión de los datos de entrada. El objetivo de esta red es reducir la dimensión de los datos de entrada y luego reconstruirlos con la mayor precisión posible en la salida. La estructura de un *autoencoder* consta de dos partes: el codificador y el decodificador. El codificador toma los datos de entrada y los comprime en una representación de baja dimensión llamada código latente, que suele ser de menor dimensión que la entrada original. Luego, el decodificador toma el código latente y lo decodifica para reconstruir la entrada original. (ver Ilustración 6)

La función de pérdida utilizada para entrenar un *autoencoder* mide la diferencia entre los datos de entrada y los datos reconstruidos. Una función de pérdida comúnmente utilizada es el error cuadrático medio (MSE, por sus siglas en inglés), que se calcula como la media de la diferencia al cuadrado entre los datos de entrada y los datos reconstruidos.

La aplicación de los *autoencoders* en la generación sintética de datos consiste en generar nuevas muestras de datos a partir de la distribución de los datos de entrada. Esto se logra muestreando de forma aleatoria el espacio latente del *autoencoder* y luego decodificando el código latente en los datos de salida. Al entrenar el *autoencoder* para que aprenda una representación de baja dimensión de los datos de entrada, se espera que el espacio latente sea más suave y continuo, lo que permite una generación más coherente de datos sintéticos.

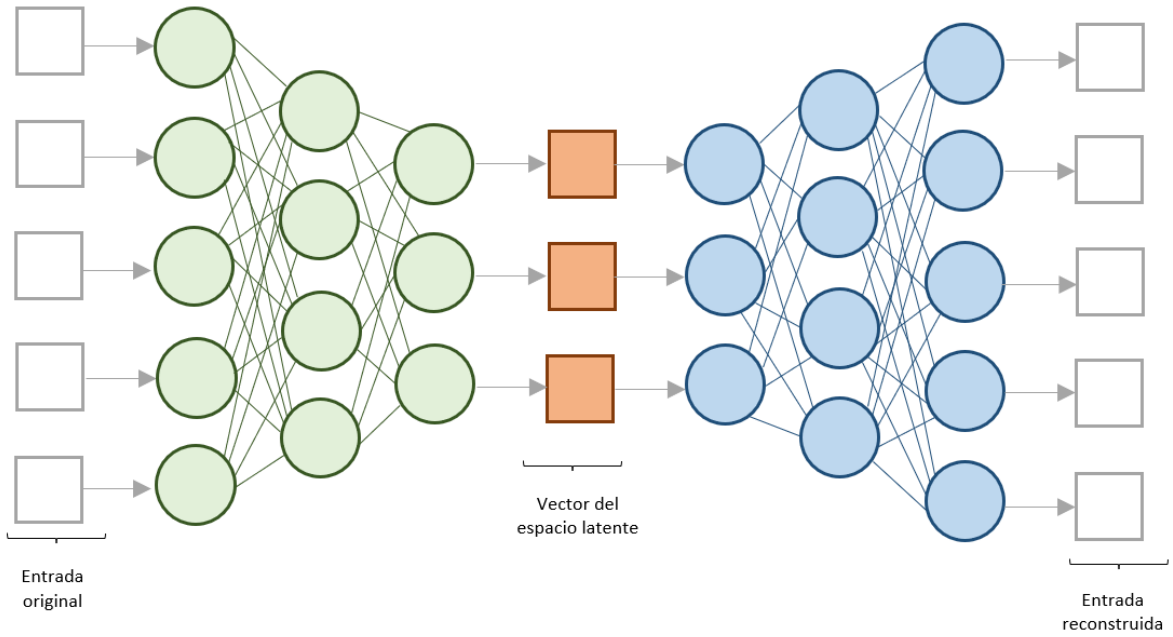


Ilustración 6: *Diagrama ejemplo de un autoencoder.*
Fuente: Elaboración propia.

Dentro del marco de los *autoencoders* se encuentran muchas variantes en función del problema a resolver. En primer lugar, los *autoencoders* básicos (*vanilla autoencoders*), son la versión más simple de *autoencoders* y consisten en redes neuronales *feedforward* de múltiples capas (Ilustración 6). Estos son adecuados para reducción de dimensionalidad, extracción de características y eliminación de ruido en conjuntos de datos más simples. Una variante de estos, los *denoising autoencoders* o *autoencoders* de eliminación de ruido, están diseñados para eliminar ruido en los datos de entrada. Para ello durante el entrenamiento, se añade ruido de forma intencionada a los datos, con el objetivo de reconstruir una versión "limpia" a partir de

la entrada ruidosa [43]. Son útiles en tareas de preprocesamiento de imágenes y señales, eliminando ruido y preservando las características importantes.

Otra variante, los *sparse autoencoders*, introducen una restricción de escasez en las activaciones de las neuronas en la capa oculta, lo que permite aprender representaciones más significativas y útiles. Son útiles para la extracción de características y la reducción de dimensionalidad en casos donde las características relevantes son escasas [44].

En tareas relacionadas con imágenes y datos en malla unos de los *autoencoders* más efectivos son los *autoencoders* convolucionales. Estos utilizan capas convolucionales en lugar de capas completamente conectadas en el codificador y el decodificador [45]. Y gracias a esta arquitectura, aprovechan la estructura espacial local de los datos y pueden aprender características jerárquicas.

Para datos secuenciales (como series temporales o texto) los *autoencoders* secuenciales o *sequence-to-sequence autoencoders* son la mejor opción. Estos *autoencoders* utilizan arquitecturas de redes neuronales recurrentes (RNN, por sus siglas en inglés), como LSTM o GRU, en el codificador y el decodificador; y son útiles en tareas como el análisis de series temporales, preprocesamiento de texto y traducción automática [46].

Finalmente, los VAEs son un tipo de *autoencoder* generativo que introduce una distribución probabilística en el espacio latente. Los VAEs son especialmente útiles en generación de datos, interpolación y exploración de espacios latentes complejos. En este respecto, el VAE es uno de los modelos que se utilizarán en este trabajo y por tanto se desarrollará más en profundidad a continuación.

3.3.1.1 Variational Autoencoder (VAE)

El VAE fue propuesto en 2013 por Kingma y Welling en Google y Qualcomm. Un VAE es una red neuronal generativa que proporciona una forma probabilística de describir una observación en un espacio latente. Así, en lugar de construir un codificador que genere un único valor para describir cada atributo del espacio latente, se formula un codificador para describir una distribución de probabilidad para cada atributo latente [47].

La arquitectura de un VAE es similar a la de un *vanilla autoencoder* en tanto en cuanto está compuesta por dos partes: un codificador, que aprende la codificación eficiente de un conjunto de datos y la transmite a la arquitectura de cuello de botella; y un decodificador, que utiliza el espacio latente en la capa de cuello de botella para regenerar los datos similares al conjunto de datos original [48]. El error se retropropaga desde la red neuronal en forma de función de pérdida.

La principal diferencia que presenta esta arquitectura con respecto a la de un *vanilla autoencoder* es que proporciona una forma estadística de describir las muestras del conjunto de datos en un espacio latente. Por lo tanto, en el VAE, el codificador genera una distribución de probabilidad en la capa *bottleneck* en lugar de un valor de salida único. Esta distribución con cierta varianza en lugar de un único punto permite expresar de forma muy natural la regularización del espacio latente de tal manera que se obliga a que las distribuciones devueltas por el codificador se aproximen a una distribución normal estándar. (ver Ilustración 7)

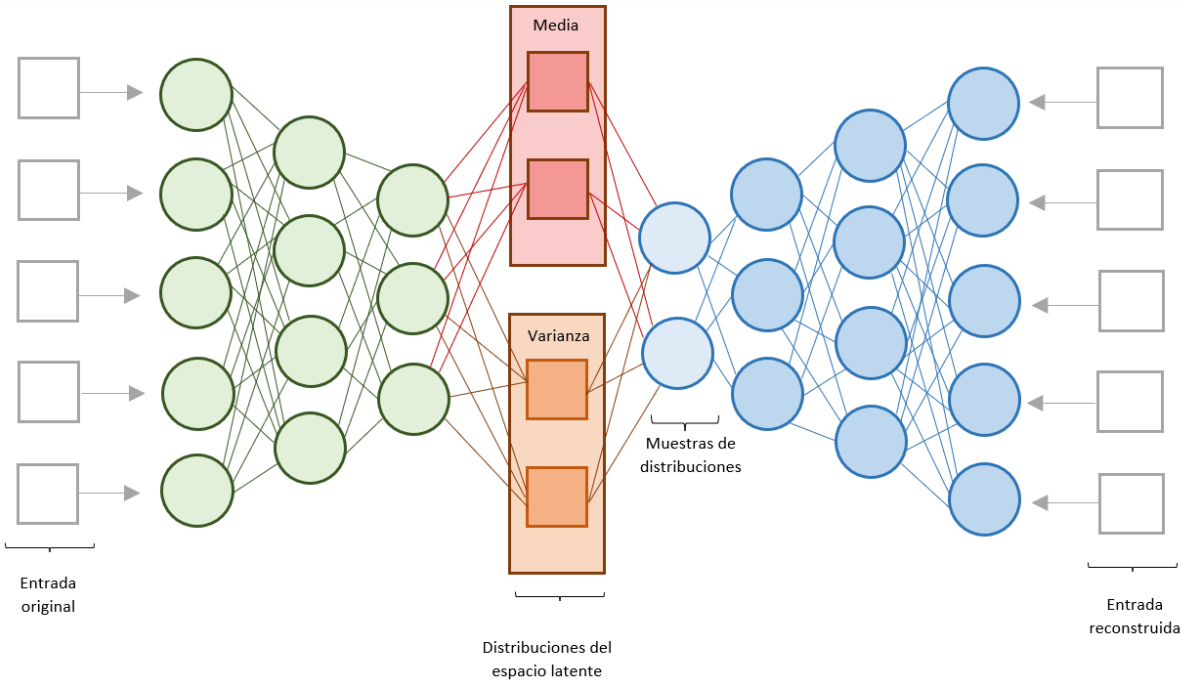


Ilustración 7: *Diagrama ejemplo de un VAE.*
Fuente: Elaboración propia.

Estas distribuciones codificadas se eligen como normales a fin de que el codificador pueda ser entrenado para devolver la media y la matriz de covarianza que describen estos gaussianos. Estos dos elementos son: $E(\mathbb{Z})$ y $V(\mathbb{Z})$, donde $\mathbb{Z}$ es la variable aleatoria latente que sigue una distribución Gaussiana con media $E(\mathbb{Z})$ y varianza $V(\mathbb{Z})$.

Así pues, la función de pérdida que se minimiza al entrenar un VAE se compone de un "término de reconstrucción" en la capa final, que tiende a hacer que el esquema de codificación-decodificación sea lo más eficaz posible, y un "término de regularización" en la capa latente, que tiende a regularizar la organización del espacio latente haciendo que las distribuciones devueltas por el codificador se aproximen a una distribución normal estándar. Ese término de regularización se expresa como la divergencia de *Kulback-Leibler* o divergencia KL [49] entre la distribución devuelta y una gaussiana estándar. La divergencia KL entre dos distribuciones gaussianas tiene una forma cerrada que puede expresarse directamente en términos de las medias y las matrices de covarianza de las dos distribuciones.

Siendo el término de reconstrucción para un problema de clasificación multiclase el siguiente:

$$BinaryCrossEntropy = \frac{1}{N} * \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log p_{ij} \qquad (2)$$

Y el termino de regularización como la divergencia KL:

$$KL(N(\mu, \Sigma_{diagonal}), N(0, I)) = \frac{1}{2} * \sum_{i=1}^{k} (\sigma_i^2 + \mu_i^2 - 1 - \ln(\sigma_i^2)) \qquad (3)$$

La función de perdida se expresa matemáticamente como:

$$loss = BinaryCrossEntropy(y, x) + \frac{1}{2} * \sum_{i=1}^{k} (\sigma_i^2 + \mu_i^2 - 1 - \ln(\sigma_i^2)) \qquad (4)$$

Cabe mencionar que pese a las ventajas que ofrece el VAE en lo referente a la generación sintética de datos en términos de estabilidad del entrenamiento, diversos estudios han señalado y estudiado alternativas a las carencias que presenta esta red neuronal.

- Pérdida de varianza: La elección de la distribución de inferencia [50], [51] parece ser la causa de la perdida de varianza en los datos generados por un VAE, dando lugar a la generación de imágenes borrosas. En este respecto, el uso de modelos de inferencia flexibles en la arquitectura ayuda a reducir este problema, y algunas alternativas propuestas

como VAE-GAN, VAE de múltiples etapas [52], VAE residual profundo o VAE jerárquicos parecen mejorar la calidad de la generación de imágenes.

En la función de pérdida de VAE, existe una tensión entre la divergencia KL, que regulariza el espacio latente, y la pérdida de reconstrucción, que afecta la calidad de la imagen. El VAE de dos etapas utiliza un factor de equilibrio que aprende durante el entrenamiento para balancear estos efectos [53].

- La representación desenredada en el ML se refiere a la separación de los factores generativos que forman los datos. En el contexto de los modelos generativos, facilita la comprensión y manipulación de la distribución de datos de entrada. Sin embargo, aprender estas representaciones de manera no supervisada es un desafío. Aunque existen variantes del VAE que intentan esto, como β-VAE [54], β-TCVAE [55] y FactorVAE [56], un estudio empírico de Google AI sugiere que aún no se ha logrado de manera no supervisada [57].

- Un problema común en la inferencia variacional es lo que en los VAEs se denomina como el "colapso posterior", por el cual las variables latentes no se utilizan en el modelo y la posterior se vuelve igual a la prior [58]. Existen modelos de VAE que intentan prevenir este colapso, como VAE de dos etapas [53], δ-VAE [59], VampPrior VAE [60], o eVAE [61] entre otros.

- Otro elemento que hay que tener en consideración es el efecto de gravedad del origen que afecta a los VAE en bajas dimensiones, donde el prior gaussiano tiende a empujar los centros de los clústeres del espacio latente hacia el origen. Este efecto puede ser explotado mediante modelos basados en GMM [62], como VADE o GMM-VAE [63], [64].

- Finalmente, los VAE sufren de la maldición de la dimensionalidad debido al uso de una norma $L2$ en la prior Gaussiana [65], lo que puede causar ineficiencias al muestrear en dimensiones altas. En tales casos, la literatura apunta a que el algoritmo de Monte Carlo Hamiltoniano [66] tiende a funcionar mejor.

3.3.2 Red Generativa Adversaria (GAN)

Las GANs se basan en un juego de suma cero formado por dos redes neuronales, la generadora y la discriminadora, que son, como su nombre indica, opuestas [24]. Esto significa que la ganancia o la pérdida de una red se equilibra con la ganancia o la pérdida de la otra red, es decir, las dos redes se compensan mutuamente.

- La red generativa se encarga de producir datos sintéticos. Para ello utiliza una función de activación no lineal, como la función de activación ReLU o la función de activación sigmoide, para transformar un vector de ruido aleatorio en una salida de datos sintéticos que imiten los datos reales de entrenamiento. La arquitectura más común de estas redes suele constar de capas densas *fully connected* o convolucionales, en función del tipo de datos que se estén generando [67].

- Por otro lado, la red discriminativa se encarga de evaluar si los datos son reales o sintéticos. Para ello también utiliza una función de activación no lineal para evaluar la autenticidad de los datos. El discriminador consta de capas densas o convolucionales que evalúan la calidad de la salida del generador y la comparan con los datos reales de entrenamiento.

Estas dos redes, generadora y discriminadora, se entrenan simultáneamente en un proceso de retroalimentación. En este proceso el generador recibe una entrada de ruido aleatorio y produce una salida sintética, mientras que el discriminador evalúa la salida del generador y la compara con los datos reales de entrenamiento. Si el discriminador detecta que la salida del generador es falsa, proporciona una señal de retroalimentación negativa al generador, para que este ajuste sus pesos y produzca una salida más realista [68]. Si el discriminador determina que la salida del generador es auténtica, proporciona una señal de retroalimentación positiva al generador, indicando que ha producido una salida de alta calidad. (ver Ilustración 8)

Para estas redes la función de pérdida es una combinación de dos funciones, una para la generadora y otra para la discriminadora. La función de pérdida del discriminador mide la precisión del discriminador al clasificar los datos como reales o sintéticos, mientras que la función de pérdida del generador mide la capacidad del generador para engañar al

discriminador produciendo datos sintéticos de alta calidad [69]. El objetivo de la GAN es minimizar la función de pérdida del discriminador y maximizar la función de pérdida del generador de forma simultánea.

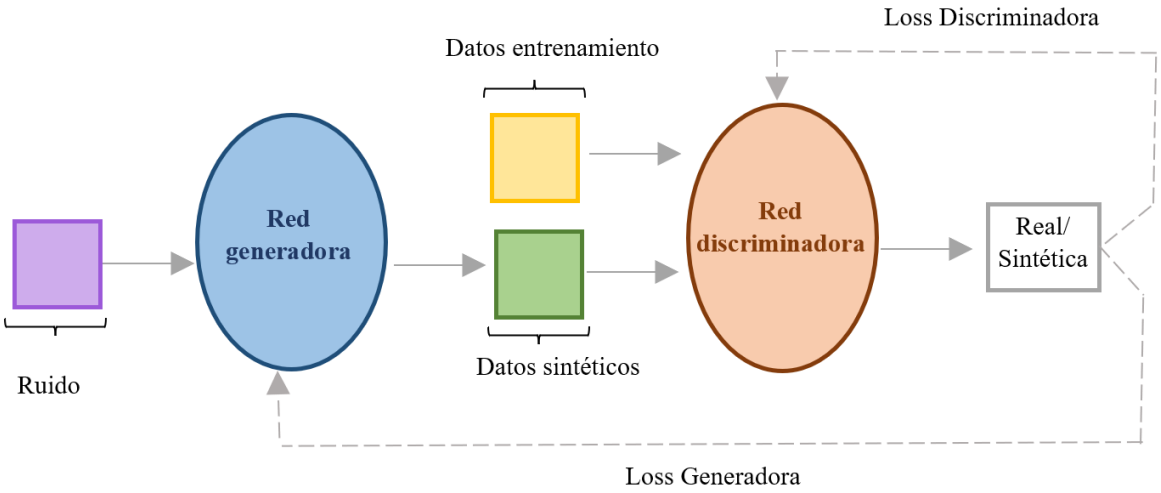


Ilustración 8: *Diagrama ejemplo de una GAN*
Fuente: Elaboración propia.

Las GANs han sido foco de gran interés desde su aparición gracias a que presentan una serie de ventajas sobre los VAE, el otro paradigma dominante hasta el momento en lo que a modelos generativos se refiere. Estas ventajas se presentan en términos de nitidez de las imágenes generadas, el tamaño de la variable aleatoria y la versatilidad en la elección de función generadora.

Sin embargo, esta arquitectura donde generador y discriminador se entrenan de manera simultánea es conocida por su inestabilidad en el entrenamiento. Una de las más conocidas, el colapso de modo, se presenta debido a la posible tendencia del generador de reproducir únicamente un modo específico que es capaz de burlar al discriminador [70]. Otra desventaja se encuentra en el desvanecimiento de gradientes que puede tener lugar cuando el discriminador se optimiza demasiado rápido en su función y los gradientes que propaga son demasiado bajos para asegurar la optimización del generador. Además, puede ocurrir que durante el entrenamiento los parámetros de ambas redes fluctúen sin encontrar un punto de

equilibrio y el generador tenga dificultades en encontrar un punto que genere imágenes de alta calidad [69].

Teniendo en cuenta las características descritas acerca de las arquitecturas de las redes GAN, durante los últimos años se han presentado arquitecturas derivadas de la original que mejoran su rendimiento en alguna de las desventajas mencionadas [71]. En primer lugar, la GAN semi-supervisada (SGAN) incluye una variación en el discriminador, añade un cabezal adicional para la predicción de la clase de pertenencia, que le permite aprovechar las ventajas de contar con datos supervisados [72]. Dependiendo de si se conoce dicha clase o no, se utiliza el cabezal *softmax* de predicción para optimizar al discriminador o la optimización vía clasificación binaria típica de la GAN convencional respectivamente. *Conditional GAN (cGAN)* introduce una entrada adicional tanto en el generador como en el discriminador que permite controlar las características de los mismos. Esta arquitectura ha sido usada en generación convolucional de caras, descripción natural de imágenes o traducción de imágenes entre otras aplicaciones. Otras arquitecturas como la *Deep Convolutional GAN* (DCGAN) o la *StyleGAN* [73] buscan mejorar la calidad y la resolución de las imágenes generadas; la primera utilizando capas convolucionales en el generador y el discriminador, y la segunda utilizando una arquitectura de generador basada en el mapeo latente.

CAPITULO 4. Métodos y Materiales

En este capítulo se detallan las herramientas, instrumentos y métodos utilizados en la realización de este estudio. Este capítulo tiene como propósito proporcionar una comprensión clara y completa de las herramientas y técnicas empleadas, facilitando así la replicabilidad y validación de nuestros resultados por parte de otros investigadores. Los instrumentos de medida como el VNA, el Sensor DR y el Lector Electrónico son presentados, seguido por una descripción del entorno de programación utilizado, que incluye Google Colaboratory y Visual Studio. Además, se discuten las distintas bibliotecas de Python utilizadas en este trabajo, como Numpy, Pandas, Scikit-learn, entre otras, con el fin de aclarar el papel que cada uno desempeña en el análisis y la interpretación de los datos.

4.1. Instrumentos de medida

Los conjuntos de datos que sirven de base para nuestro estudio están constituidos por las mediciones realizadas con dos instrumentos específicos: un VNA y un Lector Electrónico diseñado por el I.I.T.

4.1.1. Analizador vectorial de redes (VNA)

El VNA es un instrumento de medida electrónico que se utiliza para caracterizar dispositivos y componentes de radiofrecuencia (RF) y microondas en términos de su respuesta en frecuencia y sus parámetros de dispersión (parámetros S). Los VNAs son esenciales en el diseño, desarrollo y prueba de componentes y sistemas de comunicación, como antenas, componentes pasivos (filtros, divisores de potencia, etc.) y activos (amplificadores, mezcladores, etc.).

Para este proyecto se hará uso del modelo Anritsu MS46122B[1] de VNA, un modelo de alto rendimiento, diseñado para cubrir un amplio rango de frecuencias, desde 1 MHz hasta 8 GHz. Entre sus características principales, ofrece una excelente precisión, una alta velocidad de barrido y una gran estabilidad, lo que permite realizar medidas precisas y confiables de los dispositivos bajo prueba.

[1] https://www.anritsu.com/en-GB/test-measurement/support/downloads?model=MS46122B

El MS46122B destaca por su diseño compacto y su flexibilidad en la configuración, gracias a su arquitectura modular que permite adaptar el equipo a las necesidades específicas de cada aplicación. Además, este VNA cuenta con una interfaz gráfica de usuario intuitiva y fácil de usar, lo que facilita la configuración y el análisis de las medidas.

En cuanto a sus aplicaciones, el Anritsu MS46122B es ampliamente utilizado en el desarrollo y la fabricación de componentes y sistemas de RF y microondas, así como en la investigación y el diseño de dispositivos y tecnologías de comunicación inalámbrica. Entre los dispositivos que pueden ser caracterizados con este VNA se encuentran antenas, líneas de transmisión, conectores, amplificadores, filtros y dispositivos de control de fase.

4.1.2. Sensor DR

Como parte del estado del arte en el ámbito de los sensores MW, este sensor diseñado por el I.I.T. se describe con cierto detalle en la sección de Sensores de microondas.

El sensor DR empleado en el presente estudio se ha dilucidado previamente en investigaciones relativas a la caracterización de mezclas líquidas compuestas de agua y etanol [14]. Como se muestra en la Ilustración 9, el sensor DR tiene forma cilíndrica, con un diámetro de 34 mm (d_{dr}) y una dimensión vertical de 8 mm (h). El DR se fabrica a partir de zirconio, una sustancia cerámica que se distingue por una elevada permitividad relativa, $\varepsilon_r = 29$, junto con pérdidas de conducción insignificantes.

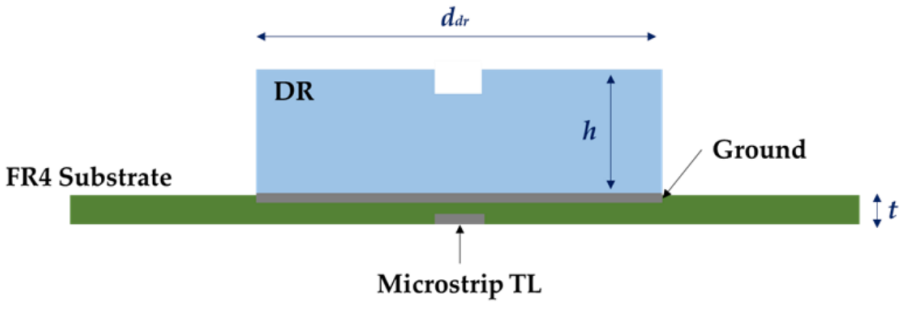


Ilustración 9: *Esquema del sensor DR.*
Imagen tomada de [13]

Encima del DR hay un portamuestras cilíndrico compacto, diseñado para alojar medidas de gotas. El funcionamiento del sensor se basa en una línea de transmisión *microstrip*, calibrada a 50 Ω y acoplada a un puerto SMA. En el plano de tierra se ha tallado una ranura rectangular, de forma y tamaño optimizados, para reforzar la transmisión de energía y garantizar una eficacia de acoplamiento superior.

Además, para mejorar la fiabilidad de las mediciones y garantizar un acoplamiento estable, se emplea una estructura de policarbonato para fijar el DR a un sustrato de fibra de vidrio FR4 (con ε_r = 4,4 y grosor t = 1,55 mm). El modo de resonancia clave dentro del DR, conocido como modo HEM110, se excita con una frecuencia de resonancia teórica que está inherentemente asociada con los atributos físicos y la permitividad del material utilizado.

Mediante una combinación de estudios de simulación y observaciones empíricas, se estableció que la frecuencia de resonancia del DR, cuando se expone al aire, es de 2,473 GHz. Este cambio observado en la frecuencia puede atribuirse a la interacción del campo electromagnético dentro del DR y la muestra líquida presente en el portamuestras. Este fenómeno se ajusta a los principios esbozados en la teoría de la perturbación.

4.1.3. Lector electrónico

El lector electrónico diseñado por el I.I.T. [13] es un dispositivo novedoso de bajo coste que permite la interrogación de un sensor DR como el previamente descrito. Este diseño, una iteración mejorada de otro ya establecido [74], incorpora tres componentes clave: un módulo de control, acondicionamiento y detección por MW.

En el núcleo del módulo de control se encuentra un Arduino MKR WiFi 1010, encargado de la generación y adquisición de señales. Esta unidad de control genera una señal triangular de barrido que se envía a un oscilador controlado por tensión (VCO) a través de un circuito de reacondicionamiento. El VCO, un Minicircuits ZX95-2536C-S+, se ajusta para modular la señal MW de salida dentro del rango de frecuencias de 2,25 a 2,55 GHz.

Esta señal de salida es dirigida por un circulador, un UiY CC2528A2400T2500SF, que gestiona la dirección de la señal MW, transmitiéndola al sensor DR a través del puerto SMA. Al

reflejarse en el sensor DR, la señal MW es recibida de vuelta por el circulador y dirigida a un circuito de amplificación para mejorar la resolución de la tensión.

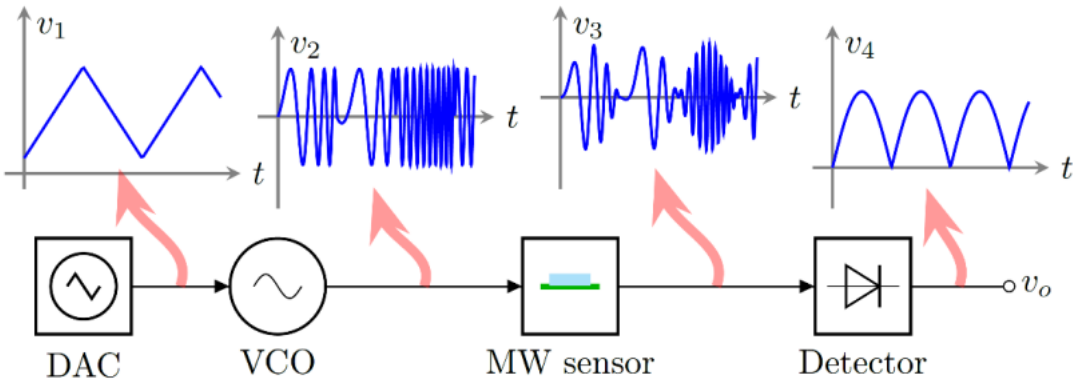


Ilustración 10: *Transformación de la señal a lo largo del flujo de trabajo del Lector electrónico:*
v1 señal triangular del DAC, v2 barrido de frecuencia desde el VCO, v3 señal reflejada por el sensor MW, v4 señal adquirida por el Arduino ADC.
Imagen tomada de [13]

Tras la amplificación, la señal se canaliza a un detector de potencia y se devuelve al Arduino. El detector de potencia traduce la potencia recibida a unidades de tensión y el Arduino digitaliza esta señal además de añadir sus correspondientes marcas de tiempo. Es importante señalar que este lector opera en el dominio del tiempo, registrando múltiples barridos de frecuencia por señal. Además, para la misma concentración de glicerina, cada señal registrada estará ligeramente desincronizada, introduciendo una varianza adicional en el conjunto de datos.

4.2. Entorno de programación

Para el desarrollo de este proyecto se han utilizado principalmente dos entornos de programación, por las herramientas que proporcionan cada uno. En lo que se refiere al entrenamiento de los modelos y aquellas tareas con alta carga computacional se ha optado por utilizar Google Colaboratory; y para el resto de las tareas y especialmente para el análisis exploratorio de los datos se ha optado por utilizar Visual Studio.

4.2.1. Google Colaboratory

Google Colaboratory[2], comúnmente conocido como Google Colab, es un entorno gratuito de Jupyter Notebook basado en la nube que permite a los usuarios escribir, ejecutar y compartir código en Python. Colab ofrece una plataforma accesible y fácil de usar para el aprendizaje, la investigación y el desarrollo de proyectos en ciencia de datos, ML e IA.

Algunas de las prestaciones más destacadas de este entorno son su integración con Google Drive, acceso gratuito a recursos computacionales como CPUs, GPUs y TPUs, lo que facilita la ejecución de tareas intensivas en recursos computacionales, como el entrenamiento de modelos de DL; y un entorno preconfigurado que permite comenzar a trabajar de inmediato sin tener que configurar un entorno local propio.

4.2.2. Visual Studio

Visual Studio[3] es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft, que proporciona una plataforma para desarrollar aplicaciones de software en diversos lenguajes de programación, como C#, C++, Visual Basic, JavaScript y Python entre otros. Visual Studio es ampliamente utilizado por desarrolladores de software en todo el mundo para crear aplicaciones web, de escritorio, móviles y en la nube.

Entre las características más destacadas de Visual Studio, se incluyen un editor de código avanzado, que facilita la escritura y organización de código; un depurador integrado, que permite depurar y solucionar problemas en el código mediante la ejecución paso a paso, puntos de interrupción y la inspección de variables y objetos; y un control de versiones integrado entre otras. Dentro de este entorno, para este proyecto se ha utilizado una versión específica de Visual Studio, Visual Studio Code, que consiste en un editor de código fuente ligero y multiplataforma.

[2] https://colab.research.google.com/
[3] https://visualstudio.microsoft.com/

4.3. Python

En lo que se refiere al lenguaje de programación utilizado, para este trabajo se ha optado por utilizar Python. Esta elección se debe principalmente a la existencia de múltiples librerías que se presentarán a continuación, además de las múltiples prestaciones que presenta este lenguaje.

Python[4] es un lenguaje de programación de alto nivel, de código abierto y de propósito general, que fue creado por Guido van Rossum en 1991 [75], y que desde su aparición ha experimentado continuos avances y versiones. Las características principales que caracterizan a este lenguaje son su legibilidad y simplicidad, lo que permite a los desarrolladores escribir código de manera eficiente y rápida. Además, cuenta con una amplia biblioteca estándar y una gran cantidad de paquetes de terceros que facilitan su uso en diferentes ámbitos.

Otras de las prestaciones más destacadas de Python es que es multiplataforma, lo que significa que se puede ejecutar en diferentes sistemas operativos, como Windows, macOS y Linux. También es un lenguaje escalable y fácil de aprender, lo que lo convierte en una opción popular para principiantes y expertos en programación.

Python es especialmente relevante en el campo de la ciencia de datos, gracias a la gran cantidad de bibliotecas y herramientas especializadas que ofrece. Entre ellas se encuentran NumPy, que permite el manejo de datos en forma de matrices y vectores; Pandas, que facilita el análisis y la manipulación de datos estructurados; y Matplotlib, que permite la visualización de datos de forma gráfica. Además, Python es ampliamente utilizado en el ML y la IA con bibliotecas como TensorFlow, Keras y Scikit-learn.

Dentro de las múltiples librerías que ofrece Python, para este trabajo se ha hecho uso principalmente de: Numpy, Pandas, Scikit-learn, Scikit-Optimize, SciPy, Matplotlib, Plotly, SDV, TensorFlow y Keras que se describirán a en las siguientes subsecciones.

[4] https://www.python.org/

4.3.1. Numpy

NumPy[5], que es un acrónimo de "*Numerical Python*", es una biblioteca de Python de código abierto esencial para cálculos numéricos y operaciones matriciales, que ofrece arreglos multidimensionales eficientes llamados "*ndarray*". Su capacidad para realizar operaciones vectorizadas permite un mejor rendimiento y un código más limpio en comparación con las listas de Python estándar [76]. NumPy es compatible con otras bibliotecas de análisis de datos como Pandas, Matplotlib y Scikit-learn, y es ampliamente utilizado en campos como análisis numérico, álgebra lineal, optimización, modelado estadístico, ML e IA.

4.3.2. Pandas

Pandas[6] es otra biblioteca de código abierto para Python que se utiliza ampliamente en el análisis y manipulación de datos. Es una de las alternativas más populares en el análisis de datos gracias a que proporciona estructuras de datos altamente eficientes y herramientas para procesar y analizar datos. Pandas permite cargar, manipular y filtrar fácilmente conjuntos de datos grandes, así como realizar tareas avanzadas de análisis de datos como análisis de series temporales, modelado estadístico y visualización de datos [77]. Las estructuras de datos principales de la biblioteca son *Series* (unidimensional) y *DataFrame* (bidimensional), que se pueden utilizar para representar datos en una forma tabular, similar a una hoja de cálculo. Esta librería también admite tareas avanzadas de manipulación de datos, como la fusión, el agrupamiento y el pivoteo, lo que la convierte en una herramienta versátil para el análisis de datos.

4.3.3. Scikit-learn

Scikit-learn[7] es una biblioteca de Python de código abierto enfocada en el ML y el análisis de datos. Proporciona una amplia variedad de algoritmos de clasificación, regresión, agrupamiento, reducción de dimensionalidad y selección de modelos, junto con herramientas para la evaluación y el ajuste de modelos. Entre sus funcionalidades más destacadas, Scikit-

[5] https://numpy.org/
[6] https://pandas.pydata.org/
[7] https://scikit-learn.org/stable/index.html

learn incluye máquinas de vectores de soporte, árboles de decisión, bosques aleatorios y regresión logística.

Scikit-learn destaca por su facilidad de uso, rendimiento y compatibilidad con otras bibliotecas de Python ya mencionadas, como NumPy y Pandas. Además, cuenta con una documentación extensa y una comunidad activa de desarrolladores y usuarios que contribuyen a su mejora y evolución. Esta biblioteca es ampliamente utilizada en la ciencia de datos, el ML y la IA para desarrollar, validar y ajustar modelos predictivos y de clasificación en diversos campos, como el análisis de imágenes, el procesamiento del lenguaje natural, la bioinformática y la detección de fraudes, entre otros.

4.3.4. Scikit-Optimize

Scikit-Optimize[8], también conocido como *skopt*, es una biblioteca de Python de código abierto diseñada para optimizar funciones objetivas costosas y ruidosas. Es una extensión de Scikit-learn que se enfoca en optimización global y local, ajuste de hiperparámetros y búsqueda de cuadrícula en la selección de modelos de ML.

Esta biblioteca ofrece una serie de algoritmos de optimización eficientes, como la optimización bayesiana, optimización de árboles de decisión y optimización basada en gradiente. Estos algoritmos permiten minimizar o maximizar una función objetivo desconocida y costosa en términos de tiempo de computación.

Scikit-Optimize es especialmente útil en la selección y ajuste de hiperparámetros en los modelos de ML, donde la evaluación de diferentes combinaciones de parámetros puede ser costosa en tiempo y recursos computacionales. La biblioteca se integra fácilmente con Scikit-learn y otras bibliotecas de ML, proporcionando una interfaz sencilla y coherente para la optimización y el ajuste de modelos.

[8] https://scikit-optimize.github.io/stable/

4.3.5. SciPy

SciPy[9] es una biblioteca de Python de código abierto centrada en la computación científica y técnica. Construida sobre NumPy, proporciona módulos adicionales para tareas como optimización, álgebra lineal, integración, interpolación, procesamiento de señales y estadísticas, entre otros. SciPy es ampliamente utilizado en la investigación científica y en aplicaciones de ingeniería para resolver problemas matemáticos y de modelado.

4.3.6. Matplotlib

Matplotlib[10] es una biblioteca de visualización de datos de Python de código abierto que permite crear gráficos y visualizaciones de alta calidad en una variedad de formatos y entornos. Ofrece una amplia gama de tipos de gráficos, como gráficos de líneas, barras, dispersión, histogramas y diagramas de caja. Matplotlib es altamente personalizable y compatible con otras bibliotecas de análisis de datos, como NumPy y Pandas, lo que facilita la visualización y el análisis de conjuntos de datos complejos en diversos campos, desde la ciencia de datos hasta la investigación científica.

4.3.7. Plotly

Plotly[11] también es una biblioteca de visualización de datos de Python de código abierto que permite crear gráficos interactivos y atractivos para la web. A diferencia de Matplotlib, que se centra en gráficos estáticos, Plotly se especializa en la creación de visualizaciones dinámicas e interactivas, lo que facilita la exploración y el análisis de datos en profundidad.

Plotly ofrece una amplia gama de tipos de gráficos similar a Matplotlib. Además, la biblioteca proporciona componentes interactivos, como paneles deslizantes, botones y menús desplegables, para crear visualizaciones personalizadas y fáciles de usar. Es destacable mencionar también su compatibilidad con otras bibliotecas de análisis de datos de Python lo que permite una integración sencilla en el flujo de trabajo de análisis de datos. Además, es

[9] https://scipy.org/
[10] https://matplotlib.org/
[11] https://plotly.com/python/

compatible con varios lenguajes de programación, como R y Julia, lo que amplía su alcance y versatilidad en la visualización de datos.

4.3.8. SDV

SDV[12], que significa Synthetic Data Vault, es una biblioteca de Python de código abierto desarrollada por el MIT *Data to AI Lab*[13] y el Grupo de Investigación de Inteligencia Artificial de Telefónica en 2016. A día de hoy, bajo el desarrollo de DataCebo[14], esta librería se ha convertido en el mayor ecosistema para la generación y evaluación de datos sintéticos. Su principal objetivo es generar datos sintéticos de alta calidad que preserven las características y las relaciones estadísticas presentes en los datos originales, mientras garantizan la privacidad de la información sensible.

Entre sus características principales, SDV ofrece una amplia gama de modelos para generar datos sintéticos para diferentes tipos de datos, como tabulares, temporales, de series temporales multivariantes y relacionales. Además, permite personalizar y ajustar los modelos según las necesidades específicas de cada proyecto, lo que facilita la creación de conjuntos de datos sintéticos realistas y precisos.

4.3.9. TensorFlow

TensorFlow[15] es una biblioteca de código abierto para computación numérica, creada y gestionada por Google Brain Team[16]. Desde su lanzamiento en 2015, TensorFlow ha liderado el campo de la IA y el DL debido a su versatilidad, escalabilidad y soporte comunitario.

TensorFlow se distingue por su arquitectura computacional simbólica, basada en el concepto de flujo de datos y grafos de cálculo. Esta arquitectura permite describir algoritmos complejos de ML en términos de operaciones matemáticas sobre tensores, que son generalizaciones multidimensionales de matrices. Las ventajas de esta biblioteca radican en su flexibilidad y

[12] https://docs.sdv.dev/sdv/
[13] https://dai.lids.mit.edu/projects/
[14] https://datacebo.com/
[15] https://www.tensorflow.org/?hl=es-419
[16] https://research.google/teams/brain/

portabilidad. La librería puede ejecutarse en diversas plataformas, desde CPU y GPU en máquinas locales hasta sistemas de cómputo en la nube. Además, permite la definición de modelos en Python y la ejecución eficiente en C++, ofreciendo así una combinación óptima entre expresividad y rendimiento.

Dentro de sus principales características, TensorFlow permite el diseño, entrenamiento y despliegue de modelos de DL, incluyendo redes neuronales convolucionales (CNN) y RNNs. Asimismo, ofrece la posibilidad de implementar algoritmos de ML tanto en CPU como en GPU, brindando una amplia flexibilidad en cuanto a la infraestructura de hardware. Esta librería es conocida por su robustez y facilidad de uso gracias a su interfaz de alto nivel Keras, que permite desarrollar prototipos de modelos de manera rápida y sencilla. Asimismo, su capacidad para trabajar con distintos formatos de datos y su amplia documentación lo hacen accesible tanto para profesionales experimentados como para aquellos que se están iniciando en el campo del ML.

4.3.10. Keras

Finalmente, Keras[17] es una API de Python diseñada para facilitar la experimentación con redes de DL. Funciona sobre TensorFlow, CNTK[18] o Theano[19], proporcionando módulos de construcción predefinidos que permiten desarrollar desde arquitecturas de red sencillas hasta complejas, incluyendo CNN y RNN. Su simplicidad y enfoque en la experiencia del usuario se destacan, permitiendo una alta modularidad y composición, lo que facilita el prototipado rápido. Todo código Keras es ejecutable tanto en CPU como en GPU sin alteraciones. Keras es ampliamente empleado en la academia y la industria para diversas tareas de DL. Su comunidad activa de usuarios y contribuyentes asegura su constante actualización y relevancia en el dinámico campo del aprendizaje profundo.

[17] https://keras.io/api/
[18] https://learn.microsoft.com/en-us/cognitive-toolkit/
[19] https://pypi.org/project/Theano/

CAPITULO 5. Modelos y discusión de resultados

5.1. Conjunto de datos

En este estudio se utilizaron dos conjuntos de datos definidos en el artículo publicado por el I.I.T. [13]. Estos conjuntos de datos son el resultado de experimentos realizados con aire y disoluciones de glicerina en concentraciones variables, que van del 0% al 80% en incrementos del 10% (como se indica en la Tabla 2). Se eligió el límite superior del 80%, ya que las concentraciones superiores, como el 90% o la glicerina pura (99%), resultaron demasiado viscosas para permitir medidas de volumen precisas con una micropipeta. Cada medida se realizó utilizando un volumen de muestra de 150 µl en la cavidad del resonador.

	ε_r Maxwell–Garnett Mixing Rule	**ε_r Literature**
Air	1	1
Gly80%	11.17	17.00
Gly70%	13.72	27.45
Gly60%	16.83	39.00
Gly50%	20.72	51.55
Gly40%	25.71	58.78
Gly30%	32.34	65.25
Gly20%	41.61	69.23
Gly10%	55.43	74.32
Water	78.30	78.30

Tabla 2: *Lista de disoluciones de glicerina probadas con sus valores de permitividad relativa a 20-21 ◦C.*
Tabla basada en [13]

El primer conjunto de datos, denominado conjunto de datos del VNA, incluye 100 espectros adquiridos para cada concentración de disolución de glicerina. Mientras que el segundo conjunto de datos, denominado conjunto de datos del Lector electrónico, contiene 180 señales para cada disolución obtenidas a partir de 35 muestras de gotas, con cinco a seis repeticiones

cada una y con un intervalo de seis segundos entre cada repetición. Todas las señales adquiridas están disponibles en un repositorio público de GitHub[20].

Los valores de permitividad relativa para las concentraciones de glicerina seleccionadas se extrajeron de la referencia [78], dentro del rango de frecuencias de 2,25 a 2,55 GHz a 21°C, fluctuando de 17 a 74,32. A modo de comparación, el valor de permitividad del agua pura a 20°C es de 78,3, según la referencia [79]. Para validar aún más estos resultados experimentales, se estimó la permitividad de cada disolución utilizando la regla de mezcla de Maxwell-Garnett [80].

Como se observa en la Tabla 2 existen ciertas discrepancias en los valores de permitividad, en este respecto la investigación realizada en el artículo del I.I.T evaluó qué valores de permitividad se ajustaban mejor a los modelos de ML utilizando tanto el VNA como el Lector electrónico de bajo coste y concluyó que la regla de mezcla de Maxwell-Garnett, que se utiliza habitualmente para estimar la permitividad de materiales compuestos con inclusiones [32], probablemente no es óptima para mezclas de líquidos como las que se utilizan para este estudio.

Las medidas del VNA se obtuvieron utilizando el VNA descrito previamente (pág. 29), que se calibró en la gama de frecuencias de 2,25-2,55 GHz. El VNA se configuró en modo de 1 puerto para obtener el coeficiente de reflexión $|s_{11}|$ y los valores de frecuencia (Ilustración 11). El buffer de muestreo utilizado fue de 5000 puntos, lo que permitió un alto nivel de precisión en las medidas. Las medidas del VNA proporcionaron una línea base de resultados para comparar con el Lector electrónico portátil diseñado y validar su rendimiento.

Por otro lado, la señal del Lector electrónico se obtiene mediante un barrido temporal. De forma similar a la adquisición del VNA, el sensor DR funciona por reflexión: el circulador recibe de vuelta la señal MW reflejada y, a continuación, transmite la señal al circuito de amplificación para aumentar la resolución de la tensión. Por último, la señal medida se transmite al detector de potencia que genera una señal de tensión en función de la potencia recibida y luego al Arduino, que digitaliza dicha señal y le añade las marcas de tiempo correspondientes. Es importante señalar que el VNA adquiere un único barrido de frecuencia, mientras que el Lector

[20] https: //github.com/MigMH/VNA_ER_GlycerinSolutions

electrónico registra una señal con varios barridos en el dominio temporal. Esto significa que las medidas del VNA proporcionan información sobre el coeficiente de reflexión a diferentes frecuencias, mientras que el Lector electrónico proporciona información sobre el comportamiento temporal de la señal en varios periodos (Ilustración 12).

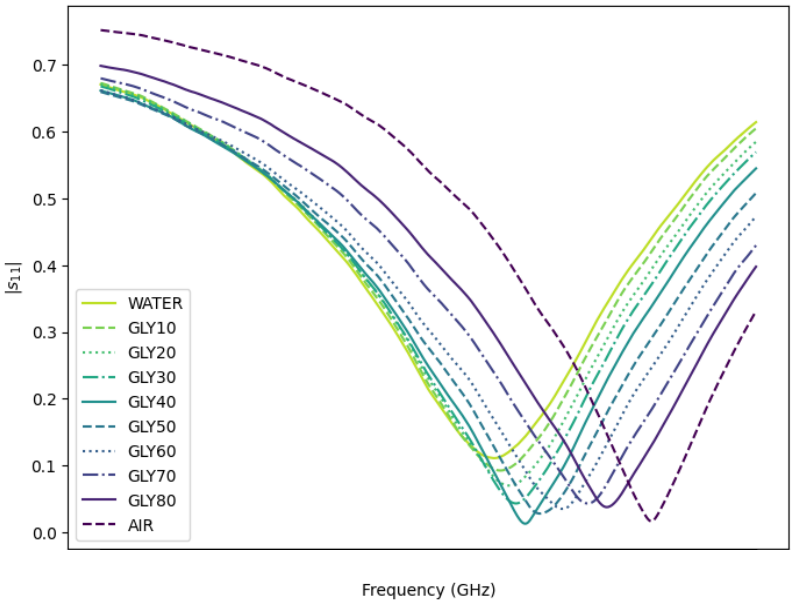


Ilustración 11: *Señales promediadas del VNA para cada concentración de glicerina.*

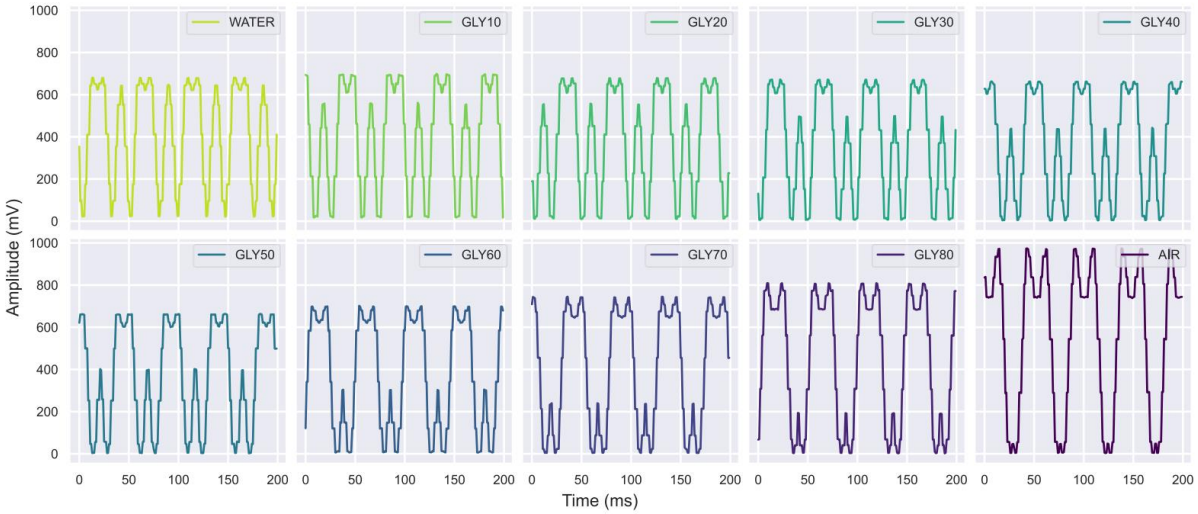


Ilustración 12: *Ejemplos de las señales del Lector electrónico para cada concentración de glicerina y aire.*

En esta primera etapa del estudio y debido a la alta dimensionalidad de los datos espectrales recopilados, se requiere el uso de métodos que reduzcan la cantidad de características. En este contexto, se utiliza el PCA, un procedimiento estadístico que detecta las direcciones en las que los datos espectrales muestran la máxima variabilidad. Este algoritmo genera un nuevo espacio matemático en el que cada espectro se proyecta, manteniendo la información más relevante. Así, cada espectro se representa mediante un vector en este espacio de componentes principales generado por el PCA.

En lo que respecta a los datos del espectro del VNA solo se requirieron tres componentes principales para que el PCA condensara el 99% de la información. Como el PCA proyecta las señales en un espacio abstracto, su explicabilidad es generalmente baja. Sin embargo, en este caso, el gráfico de PCA para el VNA revela grupos discretos y separados en lo que parece ser una curva de concentración (Ilustración 13).

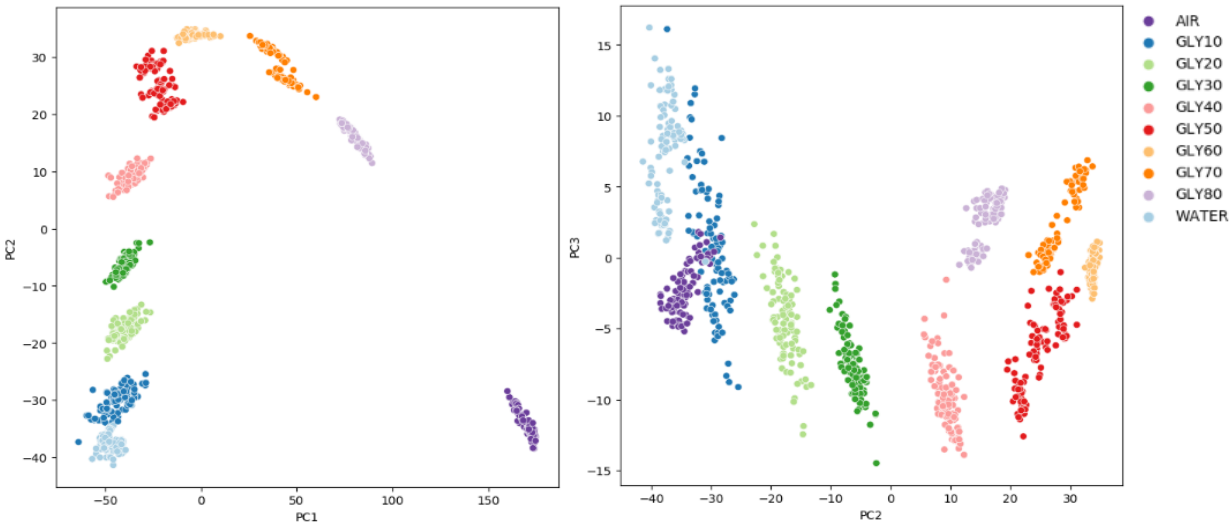


Ilustración 13: *Gráfico de dispersión de PCA del espectro del VNA.*

En contraposición a los componentes principales requeridos por el VNA, en el caso del Lector electrónico se requirieron siete componentes principales para alcanzar el 97% de la varianza explicada. Las proyecciones del VNA muestran grupos claros y distintos (Ilustración 13), mientras que el Lector electrónico proyecta patrones circulares (Ilustración 14). En particular, el gráfico PC1-PC2 muestra una proyección agrupada en círculos concéntricos, cada uno

correspondiente a una concentración de glicerina, en lo que parece ser una graduación de permitividad. Debe notarse que la separabilidad de clases alcanza su máximo cuando se consideran todas las dimensiones a la vez. Por ejemplo, el gráfico PC1-PC7 extiende la graduación en otra dimensión donde las diferencias de disolución se incrementan. En resumen, ambos métodos de adquisición muestran una buena separabilidad de clases en los gráficos de PCA que parecían anticipar buenos resultados en la aplicación de técnicas de SDG.

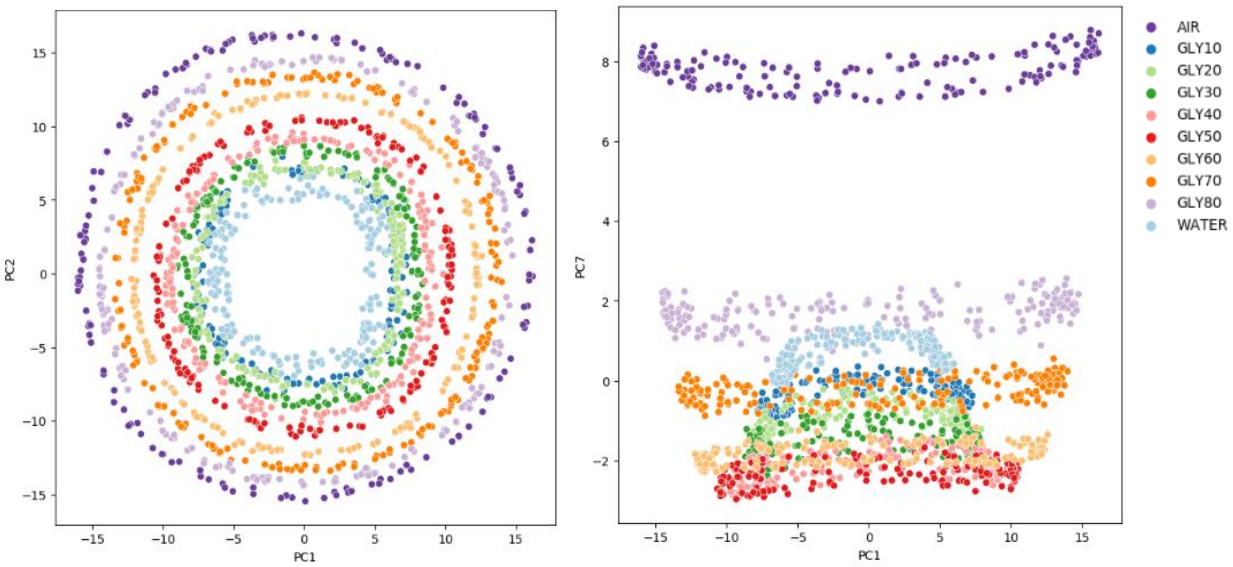


Ilustración 14: *Gráfico de dispersión de PCA de las señales del Lector electrónico.*

Tras aplicar el PCA se obtuvieron para esta etapa del estudio dos conjuntos de datos tabulares de las siguientes dimensiones:

1. **VNA:** matriz de datos multidimensional compuesta de 4 columnas (columnas con los valores de PC1 a PC3 y la etiqueta que identifica a la disolución de glicerina) y 998 filas correspondientes a las distintas muestras tomadas.
2. **Lector electrónico:** matriz de datos multidimensional compuesta de 8 columnas (columnas con los valores de PC1 a PC7 y la etiqueta que identifica a la disolución de glicerina) y 1814 filas correspondientes a las distintas muestras tomadas.

De cara a los modelos generativos que se emplearían en etapas posteriores ambos conjuntos de datos fueron diseñados de tal manera que se mantuviera un correcto balanceo de todas las

clases. Tal y como se describe en las secciones acerca de estos modelos (pág. 22-26), los modelos generativos como los VAEs y GANs aprenden la distribución de los datos de entrada. Si las clases no están balanceadas, esto puede provocar que estas redes aprendan de manera incorrecta la distribución de las clases minoritarias. De tal manera que los VAEs podrían generar menos ejemplos de la clase minoritaria, mientras que los GANs podrían "sobregenerar" la clase mayoritaria y tener dificultades en reconocer la clase minoritaria. Por ello, se consideró importante balancear correctamente las clases y tomar un número similar de muestras de todas estas.

5.2. Estudio de técnicas de generación de datos sintéticos

Tal y como se presenta en la sección acerca del estado del arte de las técnicas SDG (pág. 17), actualmente existe un gran abanico de métodos, algoritmos y modelos aplicados a la generación de datos sintéticos. Para este estudio se realizó una búsqueda bibliográfica de diversos métodos y se consideró que los modelos proporcionados por la biblioteca de Python bajo el desarrollo actual de DataCebo, SDV, era un buen punto de partida para el estudio. Tal y como se desarrolla en la correspondiente sección (pág. 38), esta librería se ha convertido en el mayor ecosistema para la generación y evaluación de datos sintéticos y ofrece una amplia gama de modelos para generar datos sintéticos para diferentes tipos de datos, como tabulares, temporales, de series temporales multivariantes y relacionales.

En un principio para este estudio se trabajó con los cuatro modelos principales que ofrece esta librería: GaussianCopula[21], CopulaGAN[22], CTGAN[23] y TVAE[24]; y se llevó a cabo un análisis en profundidad del desempeño de estos modelos en la tarea estudiada.

Como se ha expuesto anteriormente, el propósito principal de este estudio consiste en explorar la factibilidad de utilizar técnicas de generación de datos sintéticos para la optimización de modelos de caracterización de disoluciones de compuestos orgánicos. En este contexto, se efectuó un análisis en profundidad de las técnicas mencionadas y de los resultados que estas

[21] https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/gaussiancopulasynthesizer
[22] https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/copulagansynthesizer
[23] https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/ctgansynthesizer
[24] https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/tvaesynthesizer

generaban, como se detalla en las siguientes secciones. En base a la revisión y comparación de estos resultados, se constató que los modelos GaussianCopula y CopulaGAN presentaban un rendimiento insatisfactorio en la tarea mencionada en comparación con los modelos TVAE y CTGAN. Por consiguiente, se decidió eliminar estos modelos del estudio con el fin de proporcionar una representación más clara de la comparativa entre los modelos CTGAN y TVAE.

1. **CTGAN (Conditional Tabular GAN):** Este es un modelo GAN diseñado específicamente para la generación de datos tabulares [81]. CTGAN supera algunas de las limitaciones de los GAN tradicionales al incorporar técnicas de muestreo estratificado y modificación de la función de pérdida para manejar mejor las variables categóricas y la discreción de los datos. Para este estudio este modelo es especialmente interesante puesto que permite generar datos condicionados a una etiqueta o variable de entrada. CTGAN puede capturar y replicar mejor las correlaciones entre las variables en comparación con otros modelos. Sin embargo, al igual que otros modelos GAN, puede ser difícil de entrenar.

2. **TVAE (Tabular Variational Autoencoder):** TVAE es un modelo basado en un VAE diseñado específicamente para datos tabulares [81]. Los VAEs aprenden una representación de baja dimensión de los datos y luego generan nuevos datos muestreando de esta representación. TVAE adapta este enfoque a los datos tabulares al manejar de manera específica las variables categóricas y continuas. Comparado con los modelos de GANs, TVAE puede ser más estable y fácil de entrenar, pero puede que no capture las dependencias de los datos tan bien como un modelo GAN bien entrenado. Al igual que el modelo anterior, este modelo es de gran interés debido a su diseño que permite generar datos condicionados a una etiqueta o variable de entrada.

Tal y como indica la literatura acerca de la evaluación de las técnicas de SDG, esta se llevaría a cabo en función de múltiples aspectos (privacidad, similitud, utilidad y dimensiones de rendimiento). A excepción de la primera dimensión de evaluación que concierne a aspectos de

privacidad, el resto de las dimensiones serían de interés de cara a evaluar la viabilidad del uso de estas técnicas como herramienta para mejorar la caracterización de disoluciones de compuestos orgánicos.

Con el fin de llevar a cabo una comparativa en profundidad acerca del desempeño de estos dos modelos, en este estudio se han evaluado los resultados teniendo en cuenta dos escenarios adicionales: la estrategia de entrenamiento y la optimización de los hiperparámetros de cada modelo. Además, esta evaluación se realizó de manera simultánea utilizando los dos conjuntos de datos con los que se trabaja. Se consideró interesante observar si existía alguna diferencia en la comparativa de los distintos métodos en función del conjunto de datos utilizados.

5.2.1. Estrategias de entrenamiento

En primer lugar, se llevaron a cabo evaluaciones de diferentes estrategias de entrenamiento para ambos modelos. En particular, se comparó el rendimiento de cada modelo al ser entrenado con todas las disoluciones de manera simultánea, en contraste con el entrenamiento de un modelo individual para cada disolución.

La estrategia de entrenamiento que involucra todas las disoluciones busca capturar la distribución global de los datos, incluidas las posibles correlaciones y patrones existentes entre las diferentes disoluciones. El objetivo es que el modelo capture estructuras de datos más complejas, generando así datos sintéticos de mayor calidad. No obstante, como se detallará a continuación, esta estrategia enfrentó ciertas dificultades debido a las diferencias considerables en los patrones y las correlaciones entre las disoluciones estudiadas.

Por otro lado, el entrenamiento de un modelo individual para cada disolución se propuso como una forma de permitir una adaptación más específica a los patrones y estructuras de datos de cada una de estas. Esto podría resultar en una mejor calidad de los datos sintéticos para cada disolución en particular, aunque a expensas de no capturar posibles correlaciones entre las diferentes disoluciones.

La metodología seguida en esta etapa del estudio implicó diseñar un flujo de trabajo en el cual, para cada conjunto de datos utilizado (VNA y Lector electrónico), se realizó una división en

función de la disolución de las muestras. Cada subconjunto de datos correspondiente a cada disolución se empleó para entrenar un modelo y, posteriormente, generar datos sintéticos. Tras generar los datos sintéticos para cada disolución, se combinaron de nuevo en un conjunto de datos completo para llevar a cabo la evaluación (ver Ilustración 15).

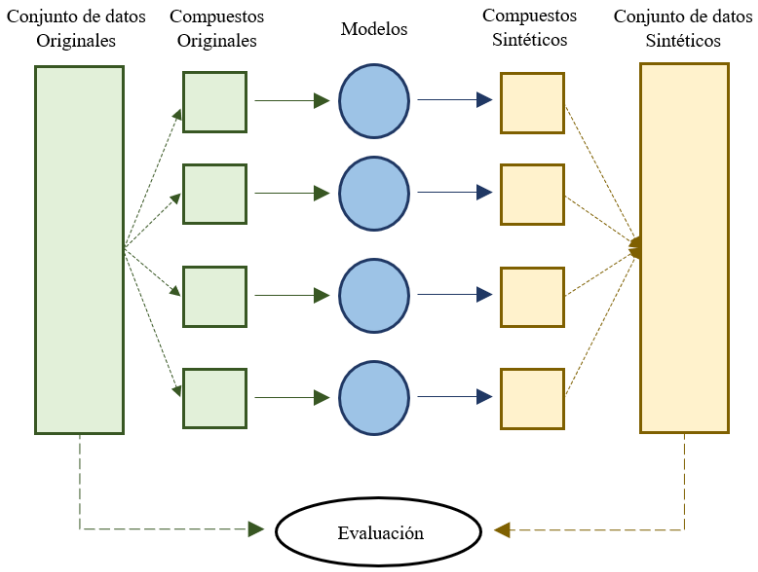


Ilustración 15: *Esquema de entrenamiento individual por disolución.*

5.2.2. Afinación de hiperparámetros

Una vez se hubieron desarrollado las dos estrategias propuestas, el siguiente paso fue evaluar la mejora en rendimiento y desempeño de ambos modelos una vez se hubiera aplicado la optimización de los hiperparámetros de cada uno. Este estudio busca evaluar las posibles diferencias en el desempeño de ambos modelos usando los parámetros predefinidos por la biblioteca y una vez se hubiera aplicado la optimización de los hiperparámetros.

El método utilizado para la optimización de los hiperparámetros fue una optimización bayesiana. De manera resumida, el enfoque de este método se basa en el teorema de Bayes para actualizar la creencia sobre la función objetivo; y fue especialmente interesante para el estudio ya que a diferencia de la búsqueda en cuadrícula o la búsqueda aleatoria, la optimización bayesiana tiene en cuenta la información de las evaluaciones anteriores para decidir qué hiperparámetros probar a continuación. Para llevar a cabo esta tarea se definió por un lado una

función objetivo común para ambos modelos y en segundo lugar el espacio de hiperparámetros para cada uno de ellos.

Para lo primero, se definió una función objetivo de tal manera que se maximizara el error de un clasificador que intenta distinguir entre datos reales y sintéticos. En términos técnicos, se puede conceptualizar como la "confusión" del clasificador, que es precisamente lo que se busca optimizar.

Los hiperparámetros son un conjunto de variables de alto nivel que influyen directamente en la capacidad del modelo para aprender y generalizar a partir de los datos. A continuación, se describirán los distintos hiperparámetros ajustados y su impacto en el entrenamiento de cada uno de los modelos.

En primer lugar, el parámetro de las épocas, representa el número total de cantidad de iteraciones completas del conjunto de datos de entrenamiento a través de las redes neuronales que componen el modelo. De cara a definir el rango de búsqueda se tomó en consideración que un número insuficientemente pequeño de épocas puede resultar en un aprendizaje deficiente, mientras que un número excesivo puede llevar al modelo a un estado de sobreajuste u *overfitting*, donde se ajusta demasiado a los datos de entrenamiento y falla en generalizar para los nuevos datos.

Por otro lado, el *batch_size* o tamaño del lote se refiere al número de muestras de entrenamiento utilizadas en una iteración del modelo (es decir, una actualización del gradiente). Un tamaño de lote más pequeño puede llevar a una actualización de los parámetros más frecuente y un entrenamiento más rápido, pero también puede resultar en un entrenamiento inestable y en una convergencia más difícil.

A partir de aquí existen distintos parámetros para cada uno de los modelos, en primer lugar, para el TVAE se tiene el *loss_factor*, un factor de ponderación que determina cuánto énfasis se coloca en minimizar la función de pérdida durante el entrenamiento. Aquí, un valor más alto significa dar mayor relevancia a la reducción de la función de pérdida, potencialmente mejorando la precisión del modelo, pero también aumentando el riesgo de *overfitting*. Un factor

de pérdida más bajo podría, por el contrario, robustecer el modelo al dar menos importancia a la función de pérdida, aunque podría limitar su precisión.

Otro parámetro importante es el *l2scale*, que se refiere al factor de escala para la regularización L2, utilizada para mitigar el *overfitting*. Un valor de *l2scale* más alto puede prevenir eficazmente el *overfitting*, pero también puede restringir la capacidad del modelo para aprender. Un valor menor, por otro lado, podría facilitar un mejor aprendizaje, aunque con un riesgo incrementado de *overfitting*.

El modelo CTGAN por otro lado cuenta con los parámetros de *generator_lr* y *discriminator_lr*. Estas son las tasas de aprendizaje para el generador y el discriminador, respectivamente. La tasa de aprendizaje controla cuánto se ajustan los pesos del modelo en respuesta a la pérdida estimada en cada paso de la actualización. De cara a determinar el valor de este parámetro hay que tener en cuenta que, una tasa de aprendizaje más alta puede resultar en aprendizaje más rápido, pero también puede resultar en saltarse el mínimo global.

En contraposición, este modelo también cuenta con los parámetros de *generator_decay* y *discriminator_decay*. Estos son los términos de decadencia para el generador y el discriminador, respectivamente; y se utilizan para disminuir la tasa de aprendizaje a lo largo del tiempo, lo que puede ayudar a mejorar la convergencia del modelo.

Otro parámetro que considerar es *discriminator_steps*, que determina el número de pasos que se toman para entrenar el discriminador por cada paso que se toma para entrenar el generador. La elección de un valor mayor provoca un mayor enfoque en el entrenamiento del discriminador, lo que puede ser útil si el generador fuera demasiado fuerte y el discriminador no pudiera mantenerse al mismo ritmo.

Posteriormente, se abordó los parámetros de *embedding_dim, dim1, dim2, ndim1, ndim2, generator_dim* y *discriminator_dim* que representan el tamaño del vector de incrustación, las dimensiones y número de las capas de compresión y descompresión en el *autoencoder*, y las dimensiones de las capas internas del generador y discriminador respectivamente. Un tamaño de incrustación mayor y dimensiones ocultas más grandes pueden permitir que los modelos

capturen relaciones más complejas en los datos. Sin embargo, estos valores también incrementan la carga computacional y el riesgo de *overfitting*.

En resumen, la selección de los hiperparámetros es una tarea de equilibrio entre precisión, robustez, generalización y eficiencia computacional. Por ello, este trabajo adopta un enfoque sistemático para la optimización de estos hiperparámetros, con el fin de obtener el rendimiento óptimo de ambos modelos.

5.3. Análisis de rendimiento en computación

Una vez presentados los escenarios que componen esta evaluación, la primera etapa de esta comparativa consistió en analizar el rendimiento en computación de ambos modelos, en función del conjunto de datos utilizados y la estrategia de entrenamiento utilizada.

El rendimiento de los modelos de generación de datos sintéticos, CTGAN y TVAE, de la librería de SDV, se puede calificar a partir de diferentes parámetros: el uso de memoria, el tiempo de generación y los recursos computacionales requeridos. Estas variables son esenciales para determinar la practicidad y eficiencia de un modelo en un contexto de aplicación real.

Estas métricas se obtuvieron utilizando librerías estándar de Python que incorporan métodos de *profiling*. En nuestra evaluación, los modelos demostraron comportamientos distintos, ofreciendo una perspectiva interesante acerca de su rendimiento según el conjunto de datos en uso. Es importante destacar que los resultados obtenidos del *profiling* dependen del hardware específico y las condiciones del sistema en el que se ejecuta el código, es por esto por lo que en este estudio la evaluación del rendimiento se realizará como una comparativa entre ambos modelos.

En primer lugar, si se analizan las métricas obtenidas por ambos modelos para el conjunto de datos provenientes del VNA se puede observar que los resultados obtenidos por el modelo TVAE parecen superar a los del modelo CTGAN en todos los escenarios estudiados. En términos de uso de memoria, el modelo CTGAN presentó en todos los escenarios evaluados un valor significativamente mayor, aproximadamente el doble que el modelo TVAE. Este

comportamiento puede ser atribuido a la complejidad intrínseca de los modelos GANs, lo que podría limitar su uso en contextos con restricciones de memoria.

Con respecto al tiempo de generación de datos sintéticos, el modelo TVAE, aunque con tiempos de generación superiores en comparación con otros modelos, demostró una eficiencia notable, superando en rendimiento al modelo CTGAN, especialmente en términos de la velocidad de generación de datos. Esta diferencia en los tiempos de generación de datos sintéticos se vio considerablemente incrementada al comparar los resultados de la estrategia de entrenamiento individual entre ambos modelos. En este aspecto, es importante mencionar que en lo que concierne al rendimiento todos los modelos experimentaron un aumento significativo en los recursos requeridos cuando se trabajaba con los modelos optimizados frente a los modelos con los hiperparámetros estándar.

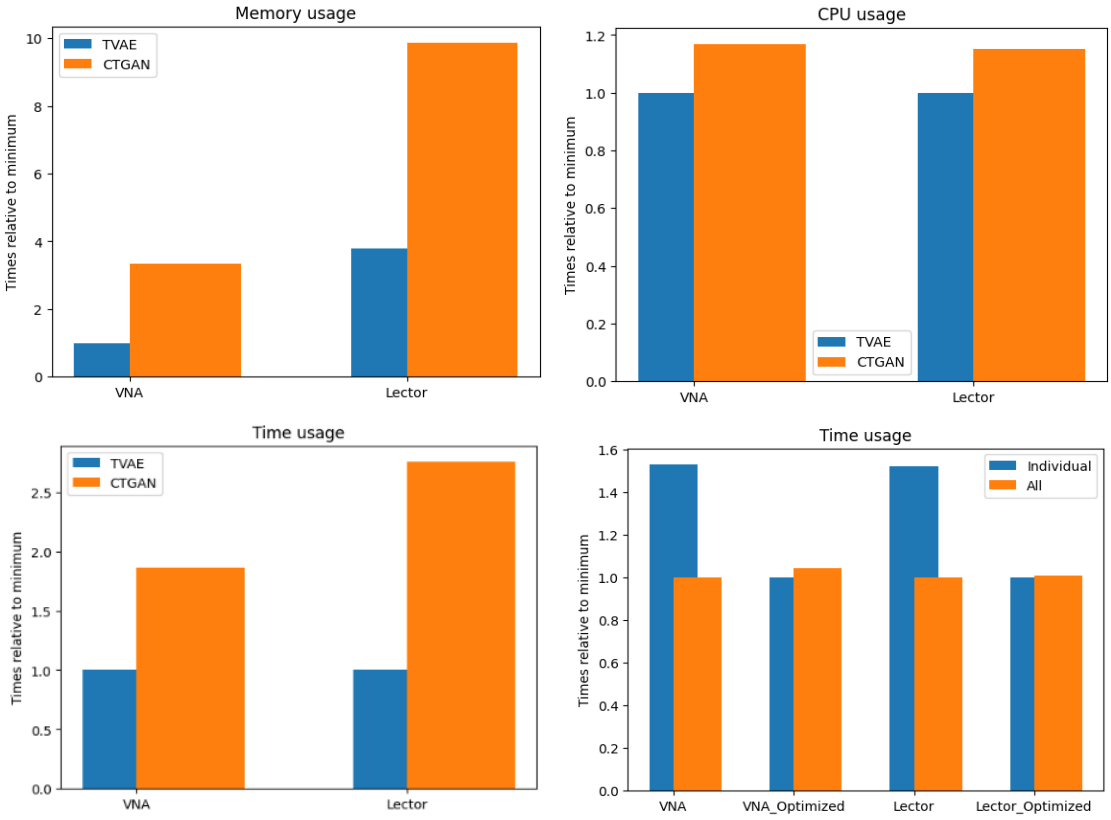


Ilustración 16: *Comparativa de métricas de rendimiento en computación.*

En cuanto al uso de recursos computacionales, ambos modelos mostraron requerimientos parecidos. Esta métrica, medida en segundos de CPU durante la generación de datos sintéticos, revela que tanto los modelos basados en GANs como el modelo TVAE pueden requerir una inversión más alta de recursos computacionales. A pesar de esto, en un entorno donde las limitaciones de recursos no son un factor determinante, como es el caso de nuestra investigación, este aspecto puede no resultar crítico.

Si se pone el foco la comparativa general de las dos estrategias planteadas en términos de rendimiento se puede observar que el tiempo requerido para el entrenamiento individual supera al del entrenamiento simultaneo para ambos modelos. El entrenamiento y la generación de datos sintéticos son tareas computacionalmente intensivas y, dentro de este marco, el entrenamiento del modelo con todas las disoluciones de manera simultánea se presenta como una opción más eficiente en términos de recursos. La principal ventaja del modelado simultaneo de las disoluciones en este aspecto está asociada al hecho de que, al entrenar un solo modelo en todo el conjunto de datos, se puede aprovechar el paralelismo para acelerar estas tareas, ya que los diferentes núcleos de la CPU o las diferentes unidades de procesamiento en una GPU pueden trabajar en distintas partes del conjunto de datos simultáneamente. En contraste, en el entrenamiento de muchos modelos individuales, cada uno de estos modelos requiere su propio tiempo de CPU, lo que resulta en un tiempo de computación total más largo al sumarlo todo. Otra razón radica en lo que se conoce como *overhead* de entrenamiento. Cada vez que se inicia el entrenamiento de un nuevo modelo, existe un cierto *overhead* o coste de tiempo asociado con la inicialización del modelo, la preparación de los datos, etc. En el caso del entrenamiento individual de los modelos, este *overhead* se incurre muchas veces, lo que también contribuye a un tiempo de computación total más largo.

Los resultados obtenidos de hacer este mismo análisis a partir de los datos del Lector electrónico no son muy distintos. A pesar de un ligero incremento en todos los valores obtenidos, debido principalmente al incremento en complejidad que supone el uso de este conjunto de datos, las métricas obtenidas en términos de uso de memoria, tiempo de generación de datos sintéticos y uso de CPU continúan presentando al modelo TVAE como una alternativa más eficiente que el CTGAN en esta tarea. La Ilustración 16 muestra la comparativa entre

ambos modelos en términos de memoria utilizada, consumo de CPU y tiempo de generación de datos sintéticos. De este último parámetro, la Ilustración 16 presenta una gráfica donde se observa la comparativa realizada en función de la estrategia de entrenamiento utilizada. De estos resultados se puede extraer que las diferencias observadas entre el modelo CTGAN y TVAE a excepción del uso de CPU se ven incrementadas en gran medida para los datos del Lector electrónico. Se observa además que para los modelos optimizados el tiempo de generación de datos sintéticos no parece diferir apenas entre ambas estrategias, comparado con los modelos sin optimizar donde si se observan las diferencias mencionadas previamente. Estos resultados apuntan a que el factor más importante en lo que al tiempo de entrenamiento de los modelos se refiere es el número de épocas, numero de lotes y las dimensiones internas de las capas definidas para cada uno de los modelos; es por esta razón que para los modelos optimizados se puede constatar que la comparativa entre las dos estrategias planteadas no muestra diferencia (Ilustración 16).

A la vista de los resultados, se puede concluir que ambos modelos, aunque con diferencias marcadas en términos de uso de memoria y tiempo de generación, (Ilustración 16) pueden considerarse viables para la generación de datos sintéticos en la caracterización de disoluciones de compuestos orgánicos, dado su potencial para capturar dependencias complejas y generar datos de alta calidad. Cada uno presenta ventajas y desventajas que deben ser ponderadas cuidadosamente según las necesidades específicas del problema de investigación.

5.4.Dimensiones de similitud

En el marco de la evaluación de técnicas SDG, el concepto de similitud juega un papel clave. La similitud, en este contexto, puede entenderse como la medida en la que los datos sintéticos reflejan las propiedades estadísticas, la estructura y las relaciones inherentes de los datos originales. La evaluación de esta similitud es crucial para garantizar la utilidad y la eficacia de los datos generados. Hay múltiples factores a considerar al evaluar la similitud, que se pueden dividir en tres categorías principales: similitud univariada, similitud multivariada y similitud en términos de dependencias de datos.

Cada una de estas categorías proporciona una lente diferente para evaluar la similitud entre los conjuntos de datos originales y sintéticos. La elección de las métricas apropiadas para evaluar la similitud dependerá del contexto específico y los objetivos de la generación de datos sintéticos. En el marco del estudio que aquí se desarrolla, la caracterización de disoluciones de compuestos biológicos, se considera especialmente relevante considerar métricas que reflejen la preservación de las relaciones entre las variables de interés. En este caso, se pondrá el foco en evaluar las métricas de similitud univariada y multivariada.

En este estudio las métricas elegidas para la evaluación de la similitud entre los datos sintéticos generados por los modelos expuestos y los conjuntos de datos reales, serán principalmente las métricas de *KSComplement* y *CorrelationSimilarity*. Estas métricas provistas por la librería SDV proporcionan una herramienta para estudiar tanto la similitud univariada como la similitud multivariada entre nuestros datos. Es importante tener en cuenta que la similitud perfecta en todas las dimensiones no es siempre el objetivo, especialmente si los datos sintéticos que se buscan generar están destinados a ser utilizados para mejorar la generalización del modelo de caracterización o para introducir variabilidad en los datos.

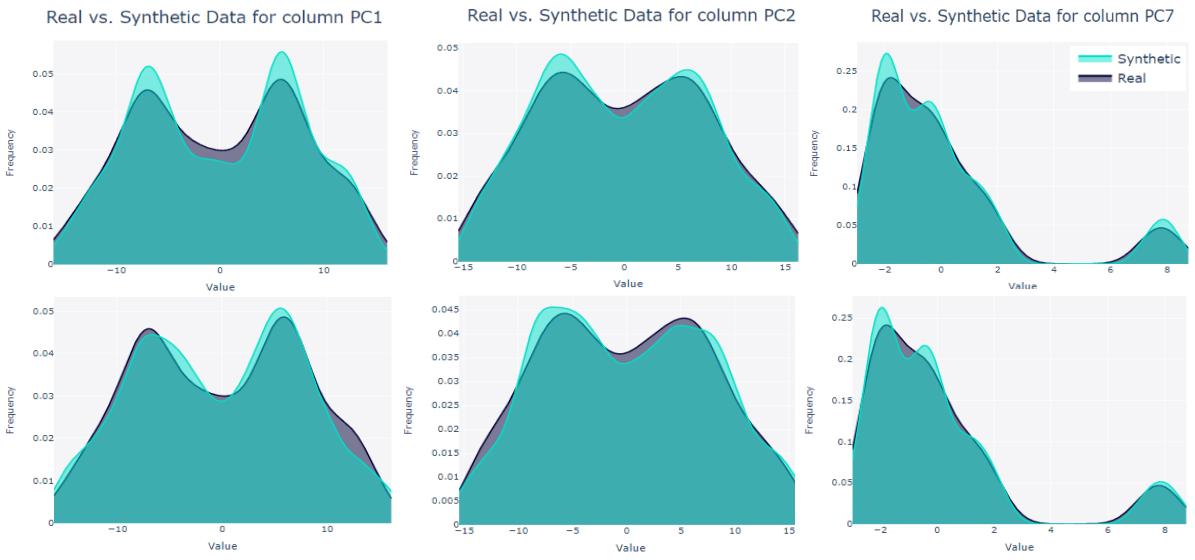


Ilustración 17: *Comparativa de KSComplement entre CTGAN (arriba) y TVAE (abajo)*

En primer lugar, la similitud univariada se refiere a la comparación de distribuciones marginales de variables individuales en los conjuntos de datos originales y sintéticos. Las métricas utilizadas para evaluar la similitud univariada pueden incluir medidas estadísticas

clásicas, como la media, la mediana, el rango intercuartílico y la desviación estándar. Esta categoría también incluye pruebas estadísticas, como la prueba de Kolmogórov-Smirnov, para comparar las distribuciones.

Dentro de esta categoría la métrica elegida, *KSComplement***,** evalúa la similitud de las distribuciones marginales unidimensionales de los datos reales y sintéticos basándose en la estadística de Kolmogórov-Smirnov, una medida no paramétrica de la discrepancia entre dos distribuciones de probabilidad. La estadística de Kolmogórov-Smirnov se obtiene de trazar la distribución acumulativa (CDF) de dos muestras e identificar el punto en el que estas dos funciones se alejan más. Esta métrica es aplicable a datos continuos y numéricos. Su rango es [0,1], siendo 1 la similitud perfecta.

Considerando los resultados obtenidos en términos de similitud univariante, los datos sintéticos generados por los modelos CTGAN y TVAE demuestran un rendimiento destacado. Concretamente, CTGAN y TVAE exhiben valores superiores en casi todas las variables a 0.7 en la métrica de *KSComplement*, siendo el TVAE el que presenta el valor más alto. Esta métrica refleja la capacidad de los modelos para reproducir las distribuciones marginales y las dependencias entre ellas, que son aspectos críticos en la generación de datos sintéticos. En la Ilustración 17, se pueden observar los resultados obtenidos de evaluar tanto para el modelo optimizado de TVAE como para el de CTGAN, el *KSComplement* de las variables PC1, PC2, PC7 entre los datos sintéticos y reales del conjunto de datos del Lector electrónico. En lo relativo al *KSComplement*, los resultados obtenidos por ambos modelos para los distintos conjuntos de datos son ciertamente similares. Las métricas proporcionadas por esta librería muestran que los datos sintéticos generados por ambos modelos superan el 90% de cobertura de rangos numéricos, categorías y límites mínimos y máximos presentes en los datos reales de ambos conjuntos de datos.

En segundo lugar, la similitud multivariada se refiere a la comparación de las relaciones entre las variables en los conjuntos de datos originales y sintéticos. Esto implica el uso de medidas de correlación, como el coeficiente de correlación de Pearson, Spearman o Kendall, para comparar las correlaciones entre pares de variables en ambos conjuntos de datos. Dentro de

esta categoría, también se incluye el uso de métodos, como el PCA, para comparar la estructura multivariada de los datos.

Dentro de las métricas definidas dentro de esta categoría, la métrica de *CorrelationSimilarity* evalúa la similitud de las correlaciones entre pares de columnas en los datos reales y sintéticos utilizando los coeficientes de correlación de Pearson o Spearman, dos medidas estadísticas de la dependencia entre dos variables. El coeficiente de Pearson mide la correlación lineal, mientras que el coeficiente de Spearman mide la correlación monótona. La métrica *CorrelationSimilarity* calcula estos coeficientes para cada par de columnas en los datos reales y sintéticos y luego devuelve un puntaje de similitud. Esta métrica es aplicable a datos continuos y numéricos. Su rango es [0,1], siendo 1 la correlación perfecta.

En el análisis de la similitud multivariante se encuentra que los valores de las métricas obtenidas de los datos sintéticos generados por el modelo CTGAN son inferiores a los de TVAE para todos los escenarios estudiados. Como se ha definido la similitud multivariada describe el grado de relaciones no lineales presentes en los datos reales ha conseguido captar nuestro modelo, en este sentido se observa que el modelo CTGAN experimenta variaciones apreciables en esta métrica en función de las características del entrenamiento, a diferencia del TVAE que parece ser más estable en este respecto.

Si se toman en consideración ambas métricas se puede observar que los resultados en términos de similitud entre los conjuntos de datos originales y los generados sintéticamente varían de manera significativa en función de la estrategia de entrenamiento utilizada y la optimización del modelo. En la Tabla 3 se puede observar una tabla comparativa entre ambos modelos donde se encuentran los valores alcanzados en *Column Shapes* y *Column Pair Trends*, unas métricas agregadas de *KSComplement* y *CorrelationSimilarity* respectivamente para todas las variables de cada conjunto de datos. En dicha tabla se observa que, en primer lugar, existe una mejora apreciable en los resultados obtenidos de los modelos optimizados frente a los modelos estándares, esta diferencia es especialmente notoria en el modelo CTGAN para las métricas de similitud multivariada obtenidas.

Una vez evaluadas las métricas de similitud obtenidas, la siguiente fase del análisis consistió en evaluar si los datos sintéticos generados para cada uno de los conjuntos de datos presentaban los patrones observados en los datos originales, y descritos en epígrafes anteriores. En primer lugar, el PCA de los datos provenientes del VNA presentaban las disoluciones estudiadas como grupos discretos y separados en lo que parecía ser una curva de concentración (Ilustración 13).

			VNA		Lector electrónico	
			Column Shapes	Column Pair Trends	Column Shapes	Column Pair Trends
Simultaneo	TVAE	Estándar	0.9306	0.8568	0.9510	0.9280
		Optimizado	0.9659	0.9331	0.9563	0.9407
	CTGAN	Estándar	0.7688	0.6064	0.7912	0.8677
		Optimizado	0.8361	0.6849	0.8999	0.8904
Individual	TVAE	Estándar	0.9794	0.9592	0.9634	0.9453
		Optimizado	**0.9885**	**0.9824**	**0.9816**	**0.9716**
	CTGAN	Estándar	0.9580	0.9146	0.9372	0.9316
		Optimizado	0.9811	0.9722	0.9609	0.9549

Tabla 3: *Comparativa de métricas de similitud*

En la Ilustración 18 se pueden observar los datos sintéticos generados por los modelos TVAE y CTGAN optimizados para las dos estrategias planteadas, entrenamiento individual y entrenamiento simultaneo. En lo que respecta a los modelos entrenados con las disoluciones de manera simultánea, se puede apreciar que los datos sintéticos generados por el modelo CTGAN no presentan los patrones esperados; la separabilidad de las disoluciones que se observaba en los datos originales ya no está en absoluto presente en los datos sintéticos generados por este modelo. Para el modelo TVAE con esta misma estrategia, sin embargo, sí que se observan los patrones de los datos originales y existe separabilidad entre las disoluciones. A pesar de esto, y comparándolo con los datos originales se puede observar que

existen diversas muestras mal generadas en los datos sintéticos. La gráfica de dispersión muestra diversos puntos de varias disoluciones en los rangos de valores PC1-PC2 que corresponden a otras disoluciones, disminuyendo de esta manera la separabilidad de clases entre estos y, de cara al uso de modelos para caracterizar estas mismas disoluciones, introduciendo ruido en la muestra que puede empeorar el desempeño de los mismos.

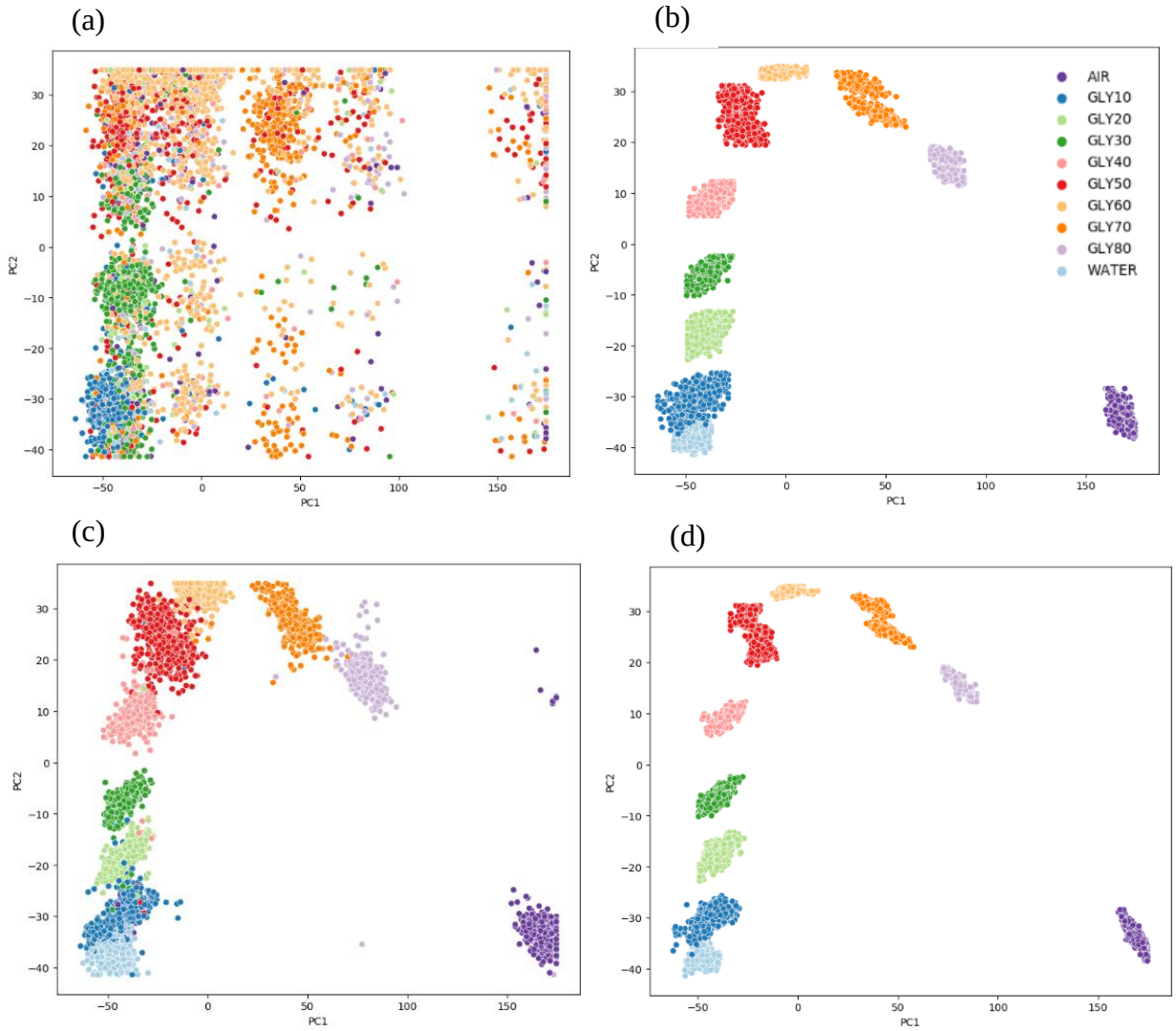


Ilustración 18: *Gráficas de dispersión de los datos sintéticos generados para el VNA. (a) Modelo CTGAN con entrenamiento simultáneo. (b) Modelo CTGAN con entrenamiento individual. (c) Modelo TVAE con entrenamiento simultáneo. (d) Modelo TVAE con entrenamiento individual.*

La estrategia de modelado individual ofrece ventajas significativas en términos de la calidad de los datos sintéticos generados. Para ambos modelos, los resultados obtenidos parecen indicar que la estrategia de modelar cada disolución por separado permite que cada modelo identifique las características propias de cada disolución (patrones, distribución, correlaciones entre las variables, …). Ambos modelos presentan para esta estrategia una separabilidad clara de todas las disoluciones y presentan unos patrones muy similares a los de los datos originales. Con respecto a los resultados obtenidos de la estrategia de entrenamiento simultaneo, los resultados obtenidos de esta segunda estrategia presentan una mejora especialmente observable en los datos sintéticos generados por el modelo CTGAN.

Al entrenar un modelo individual para cada disolución, se permite que cada modelo aprenda y reproduzca estos patrones específicos con mayor fidelidad, lo que resulta en datos sintéticos de mayor calidad en términos de similitud. En contraste, cuando se entrena un solo modelo en todo el conjunto de datos, este modelo aprende a captar los patrones y correlaciones en los datos de todas las disoluciones lo que puede provocar una "sobregeneralización" o *underfitting*. Este fenómeno se produce cuando el modelo aprende patrones demasiado amplios que no representan con precisión las características de ninguna de las disoluciones individuales.

Este riesgo de *underfitting* se reduce al entrenar modelos individuales, ya que cada modelo solo necesita captar las características de la disolución para la que se está entrenando. Otro aspecto que considerar en este análisis es la posible presencia de *outliers* o valores atípicos en alguna de las disoluciones utilizadas, en este respecto a diferencia del modelado simultaneo que se puede ver afectado en la calidad total del mismo por la presencia de estos valores atípicos; el modelado individual de las disoluciones permite que estos valores atípicos solo afecten a la o las disoluciones correspondientes.

Si se comparan los resultados obtenidos por ambos modelos para la estrategia de modelado individual, se puede observar que la varianza presente en los datos sintéticos generados por el TVAE es inferior a la de los datos generados por el modelo CTGAN. Para este conjunto de datos este aspecto toma especial importancia para las disoluciones de WATER y GLY10, cuya separabilidad de clases en PC2 puede verse afectada y dificultar la caracterización de estas dos sustancias en los modelos de clasificación.

Para el conjunto de datos proveniente del Lector electrónico, el PCA realizado sobre los datos originales presentaba patrones circulares (Ilustración 14). En particular, el gráfico PC1-PC2 mostraba una proyección agrupada en círculos concéntricos, cada uno correspondiente a una concentración de disolución de glicerina, en lo que parecía una graduación de permitividad. Además, cuando se tomaban en consideración el resto de las dimensiones, se podía observar que la separabilidad de clases alcanzaba su máximo. Un ejemplo de esto se observaba en el gráfico PC1-PC7 que extendía la graduación en otra dimensión donde las diferencias de disolución se incrementan.

En este marco, en la Ilustración 19, Ilustración 20 e Ilustración 21 se observa que existen diferencias notables en los datos sintéticos generados por ambos modelos en función del escenario estudiado. En primer lugar, si se observan los datos sintéticos generados por los modelos TVAE y CTGAN optimizados con un entrenamiento individual de cada una de las disoluciones (Ilustración 19) se pueden apreciar diferencias significativas entre los resultados de ambos modelos. La grafica de dispersión PC1-PC2 de los datos sintéticos generados por el modelo TVAE presenta los patrones circulares observados en los datos originales; estos patrones, aunque observables en los datos sintéticos no presentan la separabilidad de clases que tenían los primeros. Similar a lo que ocurría para los datos sintéticos generados por el modelo TVAE con entrenamiento simultaneo en el conjunto de datos del VNA, existen diversas muestras mal generadas en los datos sintéticos. La gráfica de dispersión muestra diversos puntos de varias disoluciones en los rangos de valores PC1-PC2 que corresponden a otras disoluciones, disminuyendo de esta manera la separabilidad de clases entre estos. Esto sin embargo no parece observarse en la gráfica de PC1-PC7 donde la separabilidad de las clases parece mantenerse prácticamente idéntica a la de los datos originales.

En contraposición, los datos sintéticos generados por el modelo CTGAN muestran unos patrones distintos a los de los datos originales y a los observados para el modelo TVAE optimizado. En este caso, si se observa la gráfica PC1-PC2 de la Ilustración 19 se puede constatar que en vez de presentar patrones circulares como se podía esperar, los datos sintéticos generados por el modelo CTGAN presentan unos patrones cuadrados. A pesar, de presentar estos patrones cuadrados, los datos sintéticos generados por este modelo parecen mantener una

separabilidad de las clases ciertamente similar al observado para el modelo TVAE, con ciertas muestras fuera de los rangos esperados, pero manteniendo en cierta manera la separabilidad de las clases. Si se observa la gráfica PC1-PC7 para este modelo se puede constatar que presenta mucha más semejanza a los datos originales; mientras que los patrones circulares presentes en PC1-PC2 no parecen haber sido captados por el modelo a la hora de generar los datos, las diferencias entre las disoluciones en PC7 sí que se mantienen. Los patrones observados para ambos modelos en PC1-PC7 son similares, pero mientras los datos sintéticos generados por el modelo CTGAN presentan una forma más cuadrada, los del TVAE se asemejan más a los de los datos originales presentando una mayor curvatura en estas dimensiones de las disoluciones.

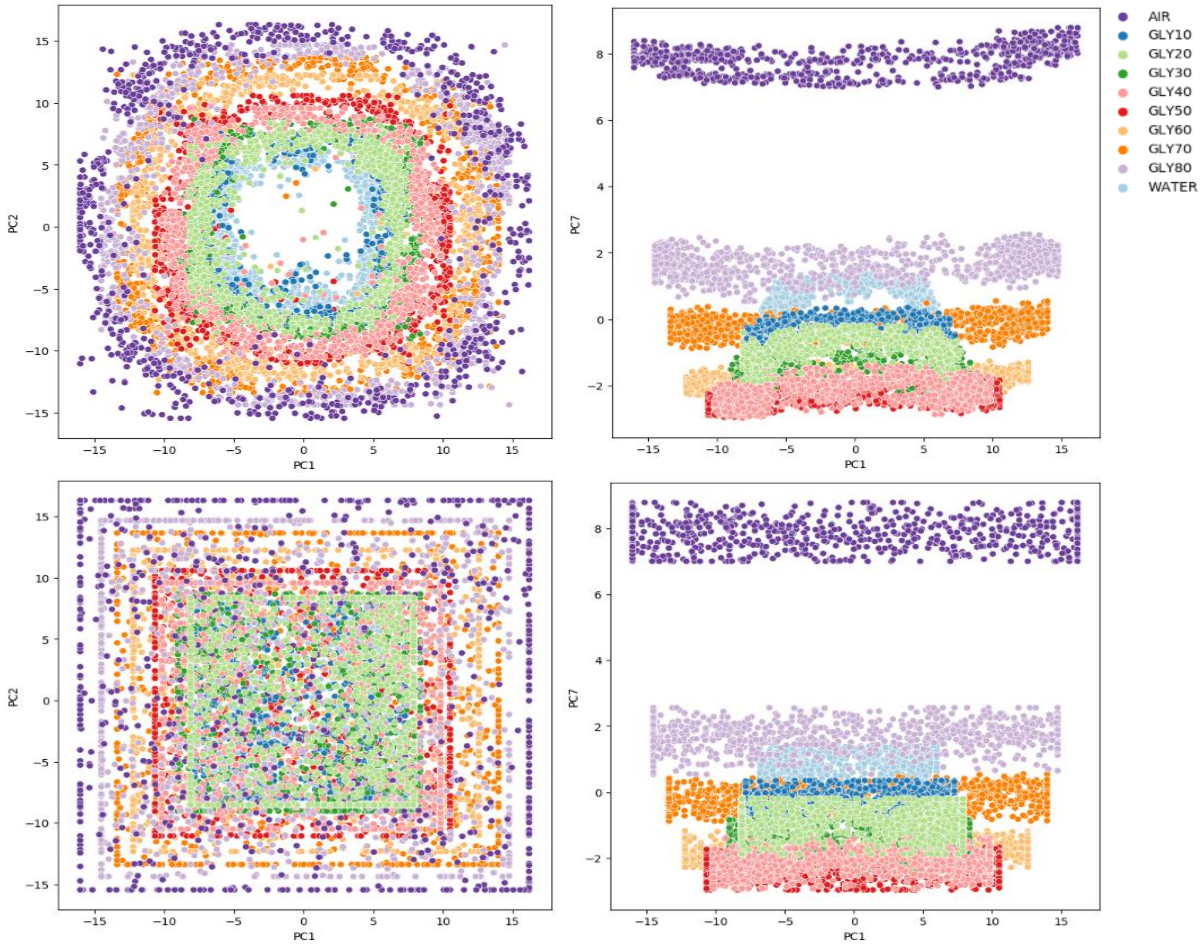


Ilustración 19: *Gráficos de dispersión de datos sintéticos generados para el Lector Electrónico (TVAE arriba y CTGAN abajo).*

Tal y como se observaba en las métricas de similitud, existe un impacto relativamente grande de la optimización de los modelos de cara a la calidad de los datos sintéticos generados. La optimización de los modelos supone, en gran medida, un aumento considerable de recursos computacionales y tiempo de entrenamiento. Este incremento se debe principalmente al valor de los parametros de épocas, numero de lotes y las dimensiones internas de las capas de cada uno de los modelos. Sin embargo, a pesar de este incremento en los recursos computaciones necesarios y el tiempo requerido, los resultados obtenidos presentan una mejora considerable con respecto a la de los modelos sin esta optimización.

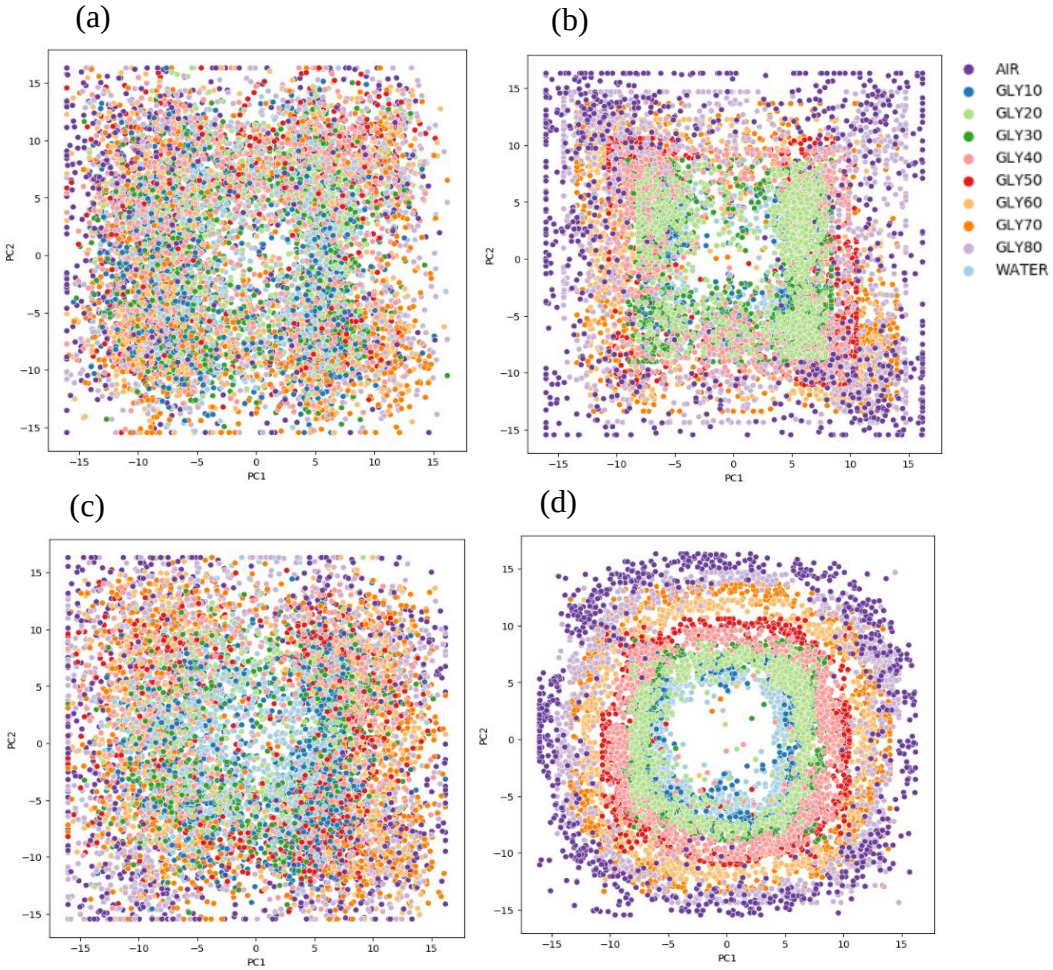


Ilustración 20: *Gráficos de dispersión de datos sintéticos generados por TVAE para el Lector electrónico.*

(a) TVAE sin optimizar y con entrenamiento simultáneo. (b) TVAE optimizado y con entrenamiento simultáneo. (c) TVAE sin optimizar y con entrenamiento individual. (d) TVAE optimizado y con entrenamiento individual.

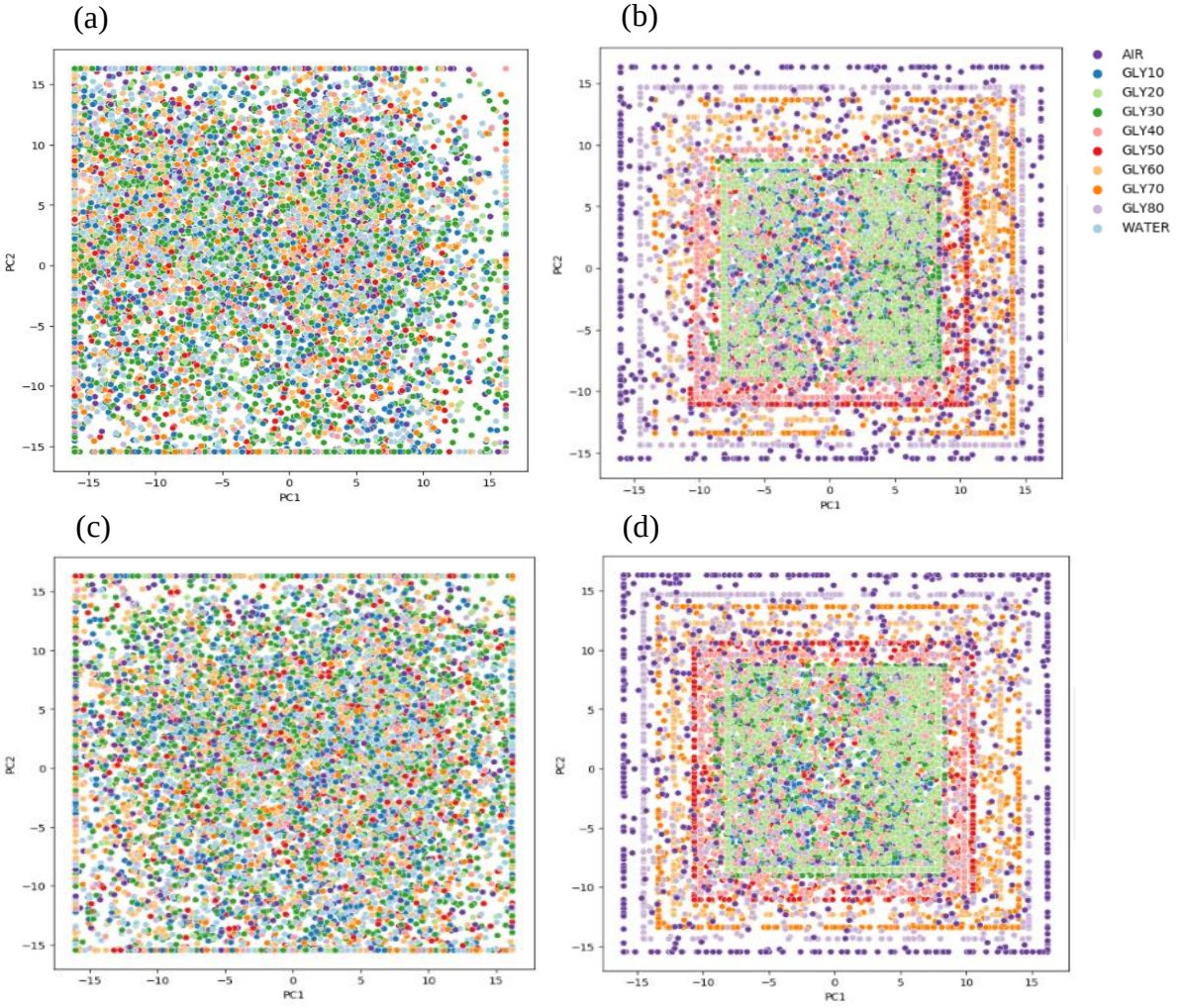


Ilustración 21: *Gráficos de dispersión de datos sintéticos generados por CTGAN para el Lector electrónico.*

(a) CTGAN sin optimizar con entrenamiento simultáneo. (b) CTGAN optimizado con entrenamiento simultáneo. (c) CTGAN sin optimizar con entrenamiento individual. (d) CTGAN optimizado con entrenamiento individual.

En la Ilustración 20 se pueden observar las gráficas de dispersión de los datos sintéticos del Lector electrónico generados por el modelo TVAE optimizado y sin optimizar para las dos estrategias de entrenamiento utilizadas. En esta, se aprecia en los gráficos de la izquierda (a y c) que sin optimización el modelo no parece ser capaz de capturar las relaciones no lineales entre las variables y la distribución y patrones presentes en los datos sintéticos generados no se asemejan a los de los datos originales. Si se observan los resultados obtenidos para el modelo optimizado se puede observar que los datos sintéticos generados sí que parecen presentar los

patrones previamente descritos. Los datos sintéticos generados por el modelo optimizado para la estrategia de entrenamiento con todas las disoluciones de manera simultánea, no presenta los patrones circulares de los datos originales, sin embargo, se puede observar que la estructura cuadrada de los datos se asemeja en cierta manera a la observada en los resultados obtenidos por el modelo CTGAN optimizado.

De la misma manera, la Ilustración 21 presenta las gráficas de dispersión de los datos sintéticos del Lector electrónico generados por el modelo CTGAN optimizado y sin optimizar para las dos estrategias de entrenamiento utilizadas. En este caso, al igual que se observaba para TVAE los datos sintéticos generados por el modelo sin optimizar no presentan ninguno de los patrones esperados. Sin embargo, mientras que en los resultados del TVAE se podía observar una cierta graduación de permitividad entre las disoluciones; para el CTGAN los datos sintéticos generados por el modelo sin optimizar se asemejan a una nube de puntos sin una estructura clara. Si se observan los resultados para el escenario con el modelo optimizado, se puede comprobar que los datos sintéticos generados por ambas estrategias presentan patrones cuadrados para las distintas disoluciones.

5.5. Dimensiones de utilidad

La evaluación de la utilidad de los datos sintéticos generados reviste una complejidad inherente dada la variedad de factores que pueden influir en su calidad. En este contexto, para este estudio se pondrá el foco en evaluar lo que se conocen como las “medidas de utilidad basadas en tareas” (Task-Based Utility Measures, TBU). Las TBU son una serie de métricas que permiten evaluar la utilidad de los datos sintéticos en función de las tareas específicas para las que se pretenden utilizar.

Este estudio se enmarca en la línea de investigación desarrollada por el I.I.T [13]. Tal y como se presenta en la sección Sensores de microondas, en este artículo los investigadores proponen el uso de un sensor DR para determinar la concentración de disolución de glicerina y usan como punto de referencia los resultados obtenidos en un VNA comercial como método de evaluar el rendimiento y validar los resultados. En este estudio se utilizó una Máquina de Vectores de Soporte (SVM) sobre el PCA para alcanzar una precisión sobresaliente (98-100%)

y se lograron valores bajos de Error Cuadrático Medio (RMSE, por sus siglas en inglés) entre 0.6 y 1.2 utilizando el Regresor de Vectores de Soporte (SVR).

El modelo SVM se utilizó para clasificar las disoluciones de glicerina en diferentes concentraciones, mientras que el modelo SVR se utilizó para estimar la permitividad de la disolución de glicerina en lugar de la concentración. Partiendo de los resultados presentados en dicha investigación, para este estudio utilizaremos como fuente de permitividad los valores adquiridos por espectroscopía dieléctrica de la literatura (Tabla 2).

La metodología seguida en este epígrafe será la de tomar como punto de referencia los resultados obtenidos de entrenar los modelos mencionados haciendo uso de los datos originales y comparar estos resultados con los obtenidos de utilizar en el entrenamiento los datos sintéticos provistos por los distintos modelos. El objetivo perseguido es evaluar la calidad de los datos sintéticos generados por cada modelo en términos de su capacidad para mejorar el rendimiento de los modelos mencionados. Las métricas obtenidas se definieron de tal manera que los modelos habrían sido entrenados con los datos sintéticos y un 10% de los datos reales, y se llevaría a cabo la predicción sobre el 90% restante de los datos reales. Con el objetivo de realizar un análisis más preciso, las métricas evaluadas serán un promedio de múltiples métricas obtenidas por cada uno de los modelos de clasificación y regresión respectivamente.

Como se ha hecho hasta ahora en este estudio, se empezará por analizar los resultados obtenidos para el conjunto de datos provenientes del VNA. En este respecto, las métricas obtenidas para los datos sintéticos generados por ambos modelos presentan resultados acordes con las observaciones realizadas hasta el momento. En primer lugar, existe una mejora significativa en las métricas obtenidas para todos escenarios estudiados donde se utilizan los modelos optimizados y el entrenamiento individual por disolución (Tabla 4). Los modelos de clasificación y regresión entrenados con los datos sintéticos de los modelos entrenados de manera simultánea y sin optimizar presentan unas métricas peores que las obtenidas en un inicio. En este escenario, la optimización de los modelos TVAE y CTGAN parece provocar una mejora significativa en los resultados obtenidos, especialmente en el caso del CTGAN. Ambos modelos mejoran su desempeño, sin embargo, mientras que el modelo TVAE experimenta una pequeña mejora obteniendo unas métricas superiores a las obtenidas para los

modelos originales (100% de precisión y 0.9 de RMSE); el modelo CTGAN experimenta una mejora muy significativa pasando de un 33% de precisión a un 68% y de un valor de 17 en RMSE a 6.33. Estos últimos resultados muestran que, a pesar de la evidente mejora de los datos sintéticos generados por modelo CTGAN optimizado, el entrenamiento simultáneo no se presenta como una alternativa viable para este modelo en esta tarea.

En lo que concierne a las métricas obtenidas con la estrategia de entrenamiento individual, se puede observar que existe una mejora significativa con respecto a las del entrenamiento simultaneo. En este escenario se puede apreciar que existe una menor diferencia entre los modelos TVAE y CTGAN en términos del valor de las métricas obtenidas. Es importante destacar que para el modelo TVAE optimizado, las métricas obtenidas además de presentar una precisión perfecta para el modelo de clasificación; también muestran un valor de RMSE bastante prometedor (0.827).

			VNA		Lector electrónico	
			RMSE	Precisión	RMSE	Precisión
Original			0.957	0.999	2.11	0.907
Simultaneo	TVAE	Estándar	1.054	0.991	8.174	0.773
		Optimizado	0.913	1	6.407	0.801
	CTGAN	Estándar	17.008	0.336	26.450	0.211
		Optimizado	6.33	0.685	23.828	0.254
Individual	TVAE	Estándar	0.955	0.997	5.696	0.898
		Optimizado	**0.827**	**1**	**2.075**	**0.958**
	CTGAN	Estándar	1.145	0.991	9.562	0.748
		Optimizado	0.905	0.997	8.654	0.874

Tabla 4: *Comparativa de métricas de utilidad*

Para el conjunto de datos del Lector electrónico los resultados son menos prometedores. Esto se puede entender por la mayor complejidad en los datos, posiblemente relacionada con la mayor dimensionalidad o las relaciones intrincadas reflejadas en las componentes principales,

puede desafiar la capacidad de los modelos. Los datos sintéticos generados por ambos modelos para este conjunto de datos con entrenamiento simultáneo son inferiores a las métricas obtenidas para los datos originales tanto con los modelos optimizados como sin optimizar. Para las métricas de los modelos optimizados utilizando el entrenamiento individual se puede observar que la precisión y el RMSE de TVAE son superiores a los de los datos originales. Tal y como se observa en la Tabla 4 los valores de estas métricas parecen reflejar que este modelo optimizado ha sido capaz de capturar de manera efectiva la distribución general de los datos originales y las dependencias no lineales en los mismos. En contraposición, el modelo CTGAN no parece haber sido capaz de capturar la estructura y las interdependencias en el conjunto de datos del Lector electrónico con la misma precisión, provocando que los datos sintéticos generados por este modelo introduzcan un cierto ruido en los datos de entrenamiento de los modelos de clasificación y regresión, y por tanto aumentando el RMSE y disminuyendo la precisión.

Como se mencionaba al principio de esta sección, la separabilidad de las clases que se observa en el PCA de los dos conjuntos de datos originales es un aspecto clave de cara al desempeño posterior de los modelos de clasificación y regresión entrenados. En este aspecto, los resultados obtenidos en términos de utilidad están estrechamente relacionados con las conclusiones extraídas de la evaluación de la similitud.

La Ilustración 22 muestra las matrices de confusión para los resultados obtenidos a partir del modelo CTGAN y TVAE. En estas gráficas se puede observar que los modelos de clasificación entrenados con los datos sintéticos de los modelos CTGAN y TVAE presentan un alto porcentaje de acierto en la caracterización de las disoluciones, aunque parecen clasificar incorrectamente disoluciones similares como GLY20 y GLY30. El incremento de varianza en los datos sintéticos de disoluciones generadas por estos modelos en comparación con los originales, parece provocar que disminuya esta separabilidad y en el caso de CTGAN especialmente empeora el desempeño de los modelos de clasificación y regresión para esta tarea.

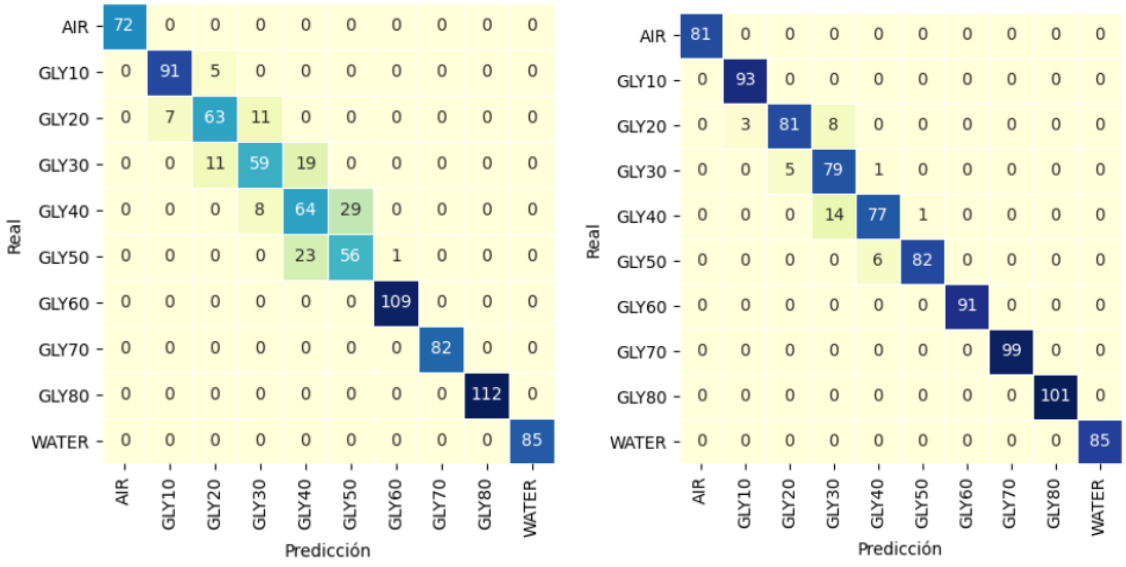


Ilustración 22: *Matriz de Confusión para el modelo de clasificación SVM para los datos sintéticos generados del Lector Electrónico por el modelo CTGAN (izq.) y el modelo TVAE (dcha.).*

5.6. Resultados

El análisis comparativo realizado acerca del rendimiento de los modelos TVAE y CTGAN en la tarea de generar datos sintéticos y su impacto en la mejora del desempeño de modelos de caracterización de disoluciones de compuestos orgánicos, nos ha permitido extraer diversas conclusiones significativas.

En primer lugar, se ha podido constatar que la estrategia de entrenamiento individual de ambos modelos generativos para cada una de las disoluciones presenta unos resultados significativamente superiores al entrenamiento simultáneo en términos de la calidad de los datos sintéticos generados. En lo que concierne a la diferencia de rendimiento en computación, para los modelos optimizados ambas estrategias presentan métricas muy similares reforzando la idea de que el entrenamiento de los modelos generativos para todas las disoluciones de manera simultánea no es una alternativa viable para esta tarea.

La evaluación realizada para el conjunto de datos del VNA ha proporcionado una línea base de resultados para comparar y validar el rendimiento de ambos modelos de cara a su posterior aplicación en el conjunto de datos del Lector electrónico. La aplicación de ambos modelos para

este conjunto de datos nos ha permitido observar la viabilidad del uso de los mismos para la mejora de los modelos de caracterización de las disoluciones estudiadas. Las métricas obtenidas para ambos modelos en términos de similitud y utilidad muestran que los datos generados para cuentan con una alta calidad y permiten mejorar los modelos de caracterización utilizados. Sin embargo, y pese a la semejanza en las métricas de ambos modelos, el modelo TVAE presenta resultados superiores al modelo CTGAN.

Estas diferencias entre ambos modelos se acrecientan en su aplicación al conjunto de datos del Lector electrónico. La separabilidad de las clases para este conjunto de datos es algo inferior a la del conjunto de datos del VNA y presenta además una serie de patrones circulares que resultan más difíciles de captar para los modelos generativos profundos como el TVAE y el CTGAN.

En lo que concierne al rendimiento en computación, el modelo TVAE se presenta como una alternativa mucho más atractiva que el modelo CTGAN. El modelo TVAE presenta unos valores de memoria utilizada y tiempo de entrenamiento y generación de los datos sintéticos muy inferiores al modelo CTGAN (Tabla 5).

Rendimiento en computación			
	Memoria	CPU	Tiempo
TVAE	1	1	1
CTGAN	2.5	1.2	3

Tabla 5: *Comparativa de modelos en rendimiento en computación*

En cuanto a la similitud de los datos sintéticos generados por ambos modelos, las métricas de calidad reflejan una superioridad clara del modelo TVAE sobre el CTGAN. Los datos sintéticos generados por el TVAE alcanzan valores superiores en la forma de las columnas y las tendencias de pares de columnas (Tabla 6). La diferencia observada entre ambos modelos en lo que respecta a estos valores (~2%) se puede ver reflejada en los patrones de los datos sintéticos generados por ambos modelos (Ilustración 19), donde los datos sintéticos generados por el CTGAN, a pesar de presentar una separabilidad de las clases adecuada, mostraban unos patrones cuadrados para cada disolución en vez de los patrones circulares presentes en los datos

originales. Esto sugiere que TVAE es capaz de capturar las dependencias en los datos de manera más efectiva, reflejándose en una calidad y similitud univariante y multivariante superiores.

Similitud		
	Columns Shapes	Column Pair Trends
TVAE	98.16%	97.16%
CTGAN	96.09%	95.49%

Tabla 6: Comparativa de modelos en similitud

Finalmente, en términos de utilidad, el TVAE también presenta unas métricas bastante superiores al CTGAN en lo que respecta a la precisión de clasificación obtenida por el modelo SVM y el RMSE obtenido para el modelo SVR. Los datos sintéticos generados por TVAE demostraron ser más útiles para las tareas de clasificación y predicción, ayudando a mejorar la precisión del modelo de clasificación y a disminuir el error del modelo SVR para ambos conjuntos de datos. En contraposición, aunque la precisión de clasificación y el RMSE del CTGAN supusieron una ligera mejora con respecto a los datos originales del VNA, se observó una caída significativa en el caso del Lector electrónico, presentando en este caso unas métricas muy inferiores a las obtenidas originalmente.

Utilidad		
	Precisión SVM	RMSE SVR
Original	90.7%	2.11
TVAE	95.8%	2.075
CTGAN	87.4%	8.654

Tabla 7: *Comparativa de modelos en utilidad*

CAPITULO 6. Conclusiones y Trabajos Futuros

6.1. Conclusiones

Este estudio ha abordado el desafío de mejorar la caracterización de disoluciones de compuestos orgánicos utilizando técnicas de DL y en específico, la generación de datos sintéticos a través de modelos generativos profundos como las GANs y los VAEs. Concretamente, este trabajo se ha enfocado en estudiar la viabilidad del desarrollo de un método de generación de datos sintéticos para la creación de un conjunto de datos más robusto y representativo de un conjunto de datos compuesto de muestras de diferentes concentraciones de disoluciones de glicerina.

En este respecto, en este estudio se ha logrado cumplir con los objetivos presentados en la sección Objetivos:

1. El objetivo principal de este estudio ha sido explorar y aplicar técnicas de SDG mediante el uso de modelos generativos profundos para crear un conjunto de datos más robusto y fiable que represente los distintos conjuntos de datos estudiados. Este objetivo se ha alcanzado tal y como se muestra a lo largo del Capítulo 5 (pág. 40-71).

Este objetivo descrito se puede dividir en otra serie de hitos que se han ido alcanzando a lo largo de este estudio.

2. Se ha conseguido desarrollar un análisis en profundidad de las técnicas de DL, con particular énfasis en los modelos TVAE y CTGAN, y su aplicabilidad en la creación de datos sintéticos. Para ello se utilizaron los conjuntos de datos adquiridos a través del VNA y el Lector electrónico, y se plantearon diversos escenarios para llevar a cabo un análisis comparativo de ambos modelos y su desempeño (pág. 40-51)

3. En lo que se refiere a la evaluación y análisis de la efectividad y eficiencia de las técnicas de SDG, concretamente los modelos TVAE y CTGAN utilizados, en este estudio se ha conseguido realizar un análisis comparativo de rendimiento en computación de ambos modelos (pág. 51) y se ha concluido que el modelo TVAE presenta una mayor eficiencia. Los

resultados obtenidos en este respecto muestran que el modelo TVAE requiere la mitad de memoria y un tercio del tiempo de entrenamiento y generación de datos comparado con el modelo CTGAN para los conjuntos de datos utilizados y los escenarios planteados (Ilustración 16).

4. En lo que se refiere a construir y perfeccionar un modelo de DL que sea capaz de producir datos sintéticos que sean indiferenciables de los datos reales recopilados a través de sensores MW basados en resonadores dieléctricos. Este objetivo también se ha alcanzado, habiendo realizado un análisis de similitud para los modelos estudiados (pág. 54) y habiendo obtenido un modelo TVAE con unas métricas de formas de columnas y tendencias de pares de columnas de 98% y 97% respectivamente. Además, tal y como se observa en la Ilustración 18 e Ilustración 19 los datos sintéticos generados por dicho modelo presentan la separabilidad de las clases observadas en los datos originales y los distintos patrones observados para cada disolución de glicerina.

5. Finalmente, en lo referente a comprobar la aplicabilidad de los datos sintéticos generados en la identificación y categorización de disoluciones de compuestos orgánicos, haciendo uso de modelos de clasificación y regresión ya existentes; los resultados obtenidos en el análisis realizado (pág. 65) muestran que los datos sintéticos generados por el modelo TVAE permiten mejorar, por un lado, la precisión del modelo SVM utilizado, llegando a alcanzar un 95% de precisión en la clasificación de las disoluciones para el conjunto de datos del Lector Electrónico frente al 90% obtenido del entrenamiento con los datos originales. Y por otro lado disminuir el error del modelo SVR para predecir la permitividad de las distintas disoluciones, alcanzando un RMSE de 2.075 frente al 2.1 obtenido con los datos originales.

Los modelos generativos profundos como las GANs y los VAEs han emergido como tecnologías altamente prometedoras para la generación sintética de datos. En nuestro estudio, se ha explorado en profundidad el desempeño y la aplicabilidad de estas dos técnicas de DL en el contexto de la generación de datos sintéticos para la caracterización de disoluciones de compuestos orgánicos. Se ha observado que ambas tecnologías presentan un gran potencial en términos de su capacidad para producir datos sintéticos de alta calidad, lo que puede contribuir significativamente a la mejora de la precisión y la eficiencia de los métodos de caracterización.

En resumen, el análisis de los modelos generativos profundos como los VAEs y las GANs ha demostrado ser una línea de investigación con un gran potencial. Este estudio ha permitido entender mejor el uso de estas técnicas de DL y sus aplicaciones en el ámbito de los sensores RD, proporcionando una base sólida para futuras investigaciones y mejoras en el campo.

6.2. Línea de investigación futura: SDG con Keras

Tal y como se ha observado a lo largo del estudio realizado, los resultados obtenidos del modelo TVAE se presentaban especialmente prometedores en términos de la calidad de los datos sintéticos generados. Sin embargo, tal y como se describe en la sección de Conjunto de datos (pág. 40), este estudio se ha realizado utilizando como datos de entrenamiento los vectores en el espacio de componentes principales generados por el PCA para cada espectro de los datos tomados del VNA y cada señal temporal del Lector electrónico.

La alta dimensionalidad de los datos espectrales recopilados, requería del uso de métodos que reduzcan la cantidad de características y es por esta razón que se decidió utilizar este procedimiento sobre los datos originales. Sin embargo, de cara a profundizar más en detalle acerca del desempeño y la viabilidad de utilizar técnicas de SDG en esta tarea se llevaron a cabo una serie de pruebas experimentales en las que se utilizaron los datos originales (pág. 40) de ambos métodos de adquisición para entrenar los modelos VAE.

El uso de PCA, como técnica que proyecta los datos en un subespacio de dimensiones inferiores, previo al entrenamiento de un modelo VAE puede introducir sesgo o dificultar la generación de datos sintéticos de calidad. Esto se debe a que el PCA asume que los datos siguen una estructura lineal y puede no captar relaciones complejas entre variables.

Por esta razón, la metodología que se planteó para esta siguiente línea de investigación fue: entrenar un modelo VAE para cada conjunto de datos originales provenientes respectivamente del VNA y del Lector electrónico. Una vez generados los datos sintéticos se le aplicaría al conjunto entero de los datos, tanto sintéticos como reales, un PCA de tal manera que se pudieran observar los resultados de una manera mucho más interpretable y se pudieran extraer conclusiones valiosas para el estudio.

Debido a la alta dimensionalidad de los datos de entrenamiento se optó por utilizar una técnica a más bajo nivel para la generación de datos sintéticos. Los modelos presentados en SDV, aunque ajustados para datos tabulares, no presentaban un desempeño óptimo en términos de recursos computacionales requeridos y tiempo de entrenamiento necesario. Es por esta razón que para esta etapa se optó por utilizar la biblioteca TensorFlow y más concretamente su interfaz de alto nivel Keras para diseñar un VAE con el que se trabajaría.

Tal y como se describe en la subsección de Keras , esta API se presenta como una alternativa muy prometedora para el objetivo perseguido gracias a su alta modularidad y composición que facilitan un prototipado rápido. Esta API funciona sobre la librería de TensorFlow, la cual al ejecutarse en C++ y trabajar con matemáticas tensoriales presenta un rendimiento muy superior al que se pudo observar para SDV en esta nueva etapa del estudio.

En términos generales, Keras brinda más flexibilidad para diseñar la arquitectura del modelo frente a las opciones ofrecidas por SDV, aunque también es mucho menos intuitivo y requiere un mayor conocimiento técnico para explotar todo su potencial.

Las pruebas preliminares realizadas haciendo uso de distintos modelos VAEs diseñados usando Keras se mostraron ciertamente prometedores. Los datos sintéticos generados presentaban una calidad muy similar a la obtenida por el mejor modelo TVAE entrenado a lo largo del estudio. En términos de similitud y utilidad los datos sintéticos generados presentaron unas métricas muy similares a la obtenidas por este modelo, sin embargo, es importante destacar que se aprecia una mejora muy considerable en las métricas asociadas al tiempo y recursos de computación.

6.3.Trabajos futuros

Mirando hacia el futuro, existen varias líneas de trabajo que podrían seguirse para expandir y mejorar la investigación llevada a cabo en este proyecto:

- **Optimización de VAE en Keras.** En primer lugar, existe la posibilidad de seguir profundizando en el estudio de los VAE diseñados en Keras. Los resultados obtenidos de las pruebas preliminares realizadas muestran que el uso de Keras para diseñar un

VAE se presenta como una alternativa muy interesante para continuar mejorando los modelos ya definidos a lo largo de este estudio. Para este estudio se diseñaron múltiples versiones de un VAE definiendo la función de perdida, aplicando re-parametrización y modificando el número y las características las capas utilizadas en el *encoder* y el *decoder* entre otras cosas. Keras ofrece muchas herramientas para realizar arquitecturas mucho más a medida, por lo que es posible continuar profundizando en el estudio realizado y buscando una mejor arquitectura para la tarea planteada.

- **Estudio de otras técnicas SDG.** Otra línea de investigación sería la de explorar la utilización de otras técnicas de SDG para comparar su rendimiento y eficacia con las GANs y los VAEs utilizados en este estudio. Este estudio se ha enfocado en estudiar estos dos modelos, pero tal y como se muestra en la sección de Técnicas de generación de datos sintéticos (SDG) existe un gran abanico de técnicas en este ámbito, por lo que sería interesante estudiar algunas otras de las presentadas en dicha sección. Esto permitiría una mejor comprensión del espectro de técnicas disponibles y cómo estas podrían ser utilizadas de manera efectiva en la generación de datos sintéticos.

- **Entrenamiento con otros compuestos.** Para este estudio se ha hecho uso de los conjuntos de datos obtenidos por el I.I.T. provenientes de los espectros y señales de muestras de distintas concentraciones de disoluciones de glicerina. Para futuras investigaciones sería interesante investigar más a fondo el uso de estas técnicas de DL para la generación de datos sintéticos haciendo uso de otra serie de compuestos. Las disoluciones de glicerina presentaban una buena separabilidad de las clases tras el PCA, por lo que sería interesante estudiar otras disoluciones con el objetivo de observar si existen cambios significativos en el desempeño de los modelos para disoluciones de compuestos con patrones menos distinguibles. También sería interesante observar si existe una transferencia de aprendizaje en los modelos desarrollados cuando se aplican para varias muestras de disoluciones de compuestos distintos. El objetivo final sería mejorar la caracterización de una mayor gama de disoluciones de compuestos orgánicos.

- **Estudio de utilidad más en profundidad.** Se podría realizar un estudio más detallado de la utilidad de los datos sintéticos en la tarea de caracterización y clasificación de disoluciones de compuestos orgánicos, evaluando su rendimiento en diferentes contextos y utilizando diferentes modelos de clasificación. En este estudio se han utilizado los modelos de SVM y SVR porque presentan ciertas ventajas a la hora de trabajar con conjuntos de datos pequeños como el utilizado. Sin embargo, podría ser interesante estudiar el desempeño de modelos más complejos como XGBoost o Random Forest. Esto entre otras cosas permitiría observar la calidad de los datos sintéticos generados en este respecto en función del modelo utilizado.

En conclusión, a pesar de los desafíos encontrados en el camino, este proyecto ha demostrado la viabilidad y la potencialidad de las técnicas de DL en la generación de datos sintéticos para la caracterización de disoluciones de compuestos orgánicos. Sin embargo, se reconoce que aún hay mucho espacio para la investigación y mejora en este ámbito.

El uso de datos sintéticos generados por modelos como los VAEs abre nuevas oportunidades para superar las limitaciones actuales en la recolección de datos y puede conducir a una mejora significativa en la caracterización de disoluciones de compuestos orgánicos. A su vez, esto puede tener un impacto significativo en la eficiencia y eficacia de los sensores DR, que son de vital importancia en diversas áreas, desde la industria química hasta la medicina y la biotecnología.

Finalmente, se espera que este trabajo sirva como un punto de partida para futuras investigaciones en el campo de la generación de datos sintéticos y su aplicación en el ámbito de los biosensores. Creemos firmemente que esta línea de trabajo tiene el potencial de conducir a avances significativos en la eficacia y eficiencia de la caracterización y clasificación de disoluciones de compuestos orgánicos.

CAPITULO 7. Referencias

[1] F. Cui, Y. Yue, Y. Zhang, Z. Zhang, y H. S. Zhou, «Advancing Biosensors with Machine Learning», *ACS Sens*, vol. 5, n.º 11, pp. 3346-3364, nov. 2020, doi: 10.1021/acssensors.0c01424.

[2] M. Monteagudo Honrubia, J. Matanza Domingo, F. J. Herraiz-Martínez, y R. Giannetti, «Low-Cost Electronics for Automatic Classification and Permittivity Estimation of Glycerin Solutions Using a Dielectric Resonator Sensor and Machine Learning Techniques», *Sensors*, vol. 23, n.º 8, p. 3940, abr. 2023, doi: 10.3390/s23083940.

[3] K. E. Schackart y J.-Y. Yoon, «Machine Learning Enhances the Performance of Bioreceptor-Free Biosensors», *Sensors*, vol. 21, n.º 16, p. 5519, ago. 2021, doi: 10.3390/s21165519.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, y W. P. Kegelmeyer, «SMOTE: Synthetic Minority Over-sampling Technique», *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, jun. 2002, doi: 10.1613/jair.953.

[5] A. Singh *et al.*, «Recent Advances in Electrochemical Biosensors: Applications, Challenges, and Future Scope», *Biosensors (Basel)*, vol. 11, n.º 9, p. 336, sep. 2021, doi: 10.3390/bios11090336.

[6] M. G. Mayani, F. J. Herraiz-Martinez, J. M. Domingo, y R. Giannetti, «Resonator-Based Microwave Metamaterial Sensors for Instrumentation: Survey, Classification, and Performance Comparison», *IEEE Trans Instrum Meas*, vol. 70, 2020, doi: 10.1109/TIM.2020.3040484.

[7] S. Keyrouz y D. Caratelli, «Dielectric Resonator Antennas: Basic Concepts, Design Guidelines, and Recent Developments at Millimeter-Wave Frequencies», *Int J Antennas Propag*, vol. 2016, 2016, doi: 10.1155/2016/6075680.

[8] S. K. K. Dash, T. Khan, y Y. M. M. Antar, «A state-of-art review on performance improvement of dielectric resonator antennas», *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 28, n.º 6, p. e21270, ago. 2018, doi: 10.1002/MMCE.21270.

[9] P. Mehrotra, B. Chatterjee, y S. Sen, «EM-Wave Biosensors: A Review of RF, Microwave, mm-Wave and Optical Sensing», *Sensors 2019, Vol. 19, Page 1013*, vol. 19, n.º 5, p. 1013, feb. 2019, doi: 10.3390/S19051013.

[10] J. Muñoz-Enano, P. Vélez, M. Gil, y F. Martín, «Planar Microwave Resonant Sensors: A Review and Recent Developments», *Applied Sciences 2020, Vol. 10, Page 2615*, vol. 10, n.º 7, p. 2615, abr. 2020, doi: 10.3390/APP10072615.

[11] R. A. Alahnomi *et al.*, «Review of Recent Microwave Planar Resonator-Based Sensors: Techniques of Complex Permittivity Extraction, Applications, Open Challenges and Future Research Directions», *Sensors 2021, Vol. 21, Page 2267*, vol. 21, n.º 7, p. 2267, mar. 2021, doi: 10.3390/S21072267.

[12] M. G. Mayani, F. J. Herraiz-Martinez, J. M. Domingo, y R. Giannetti, «Resonator-Based Microwave Metamaterial Sensors for Instrumentation: Survey, Classification, and Performance Comparison», *IEEE Trans Instrum Meas*, vol. 70, pp. 1-14, 2021, doi: 10.1109/TIM.2020.3040484.

[13] M. Monteagudo Honrubia, T. Ul Haq, B. Ali Fraea Esmail, J. Matanza Domingo, F. Javier Herraiz-Martínez, y R. Giannetti, «Low-Cost Electronics for Automatic Classification and Permittivity Estimation of Glycerin Solutions Using a Dielectric Resonator Sensor and Machine Learning Techniques», *Sensors 2023, Vol. 23, Page 3940*, vol. 23, n.º 8, p. 3940, abr. 2023, doi: 10.3390/S23083940.

[14] M. G. Mayani, F. J. Herraiz-Martinez, J. M. Domingo, R. Giannetti, y C. R.-M. Garcia, «A Novel Dielectric Resonator-Based Passive Sensor for Drop-Volume Binary Mixtures Classification», *IEEE Sens J*, vol. 21, n.º 18, pp. 20156-20164, sep. 2021, doi: 10.1109/JSEN.2021.3094904.

[15] S. Tanwar, A. Nayyar, y R. Rameshwar, *Machine Learning in Signal Processing: Applications, Challenges, and The Road Ahead*. CRC Press; Chapman & Hall, 2022. Accedido: 15 de mayo de 2023. [En línea]. Disponible en: https://www.routledge.com/Machine-Learning-in-Signal-Processing-Applications-Challenges-and-the/Tanwar-Nayyar-Rameshwar/p/book/9780367618902

[16] K. Chan *et al.*, «Low-cost electronic sensors for environmental research: Pitfalls and opportunities», *Prog Phys Geogr*, vol. 45, n.º 3, pp. 305-338, jun. 2021, doi: 10.1177/0309133320956567/ASSET/IMAGES/LARGE/10.1177_0309133320956567-FIG14.JPEG.

[17] B. Heidt *et al.*, «Point of Care Diagnostics in Resource-Limited Settings: A Review of the Present and Future of PoC in Its Most Needed Environment», *Biosensors 2020, Vol. 10, Page 133*, vol. 10, n.º 10, p. 133, sep. 2020, doi: 10.3390/BIOS10100133.

[18] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, y S. Vappangi, «A systematic literature review on prototyping with Arduino:

Applications, challenges, advantages, and limitations», *Comput. Sci. Rev.*, vol. 40, may 2021, doi: 10.1016/J.COSREV.2021.100364.

[19] N. P. Pai, C. Vadnais, C. Denkinger, N. Engel, y M. Pai, «Point-of-Care Testing for Infectious Diseases: Diversity, Complexity, and Barriers in Low- And Middle-Income Countries», *PLoS Med*, vol. 9, n.º 9, p. e1001306, sep. 2012, doi: 10.1371/JOURNAL.PMED.1001306.

[20] E. M. Rojas, «Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo», en *Revista Ibérica de Sistemas e Tecnologías de Informação*, 2021, pp. 586-589.

[21] H. H. Rashidi, N. K. Tran, E. V. Betts, L. P. Howell, y R. Green, «Artificial intelligence and machine learning in pathology: the present landscape of supervised methods», *Acad Pathol*, vol. 6, p. 2374289519873088, 2019.

[22] J. P. Lévy Mangin, J. M. Fernández Fernández, y R. Flórez López, *Las redes neuronales artificiales.* Netbiblo, S.L., 2008.

[23] D. G. García Murillo, «Redes Neuronales: Conceptos Básicos y Aplicaciones. », *https://www.academia.edu*, marzo de 2001.

[24] R. Salas, «Redes neuronales artificiales», Universidad de Valparaıso, 2004.

[25] Y. Ito, «Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory», *Neural Networks*, vol. 4, n.º 3, pp. 385-394, ene. 1991, doi: 10.1016/0893-6080(91)90075-G.

[26] S. Hochreiter, «The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions», *https://doi.org/10.1142/S0218488598000094*, vol. 6, n.º 2, pp. 107-116, nov. 2011, doi: 10.1142/S0218488598000094.

[27] A. F. Agarap, «Deep Learning using Rectified Linear Units (ReLU)», mar. 2018.

[28] M. Wasef y N. Rafla, «Hardware implementation of multi-Rate input SoftMax activation function», en *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE, ago. 2021, pp. 783-786. doi: 10.1109/MWSCAS47672.2021.9531761.

[29] I. Kouretas y V. Paliouras, «Simplified Hardware Implementation of the Softmax Activation Function», en *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, IEEE, may 2019, pp. 1-4. doi: 10.1109/MOCAST.2019.8741677.

[30] S.-C. Li, B.-C. Tai, y Y. Huang, «Evaluating Variational Autoencoder as a Private Data Release Mechanism for Tabular Data», en *2019 IEEE 24th Pacific Rim*

International Symposium on Dependable Computing (PRDC), IEEE, dic. 2019, pp. 198-1988. doi: 10.1109/PRDC47002.2019.00050.

[31] A. Dandekar, R. A. M. Zen, y S. Bressan, «A Comparative Study of Synthetic Dataset Generation Techniques», 2018, pp. 387-395. doi: 10.1007/978-3-319-98812-2_35.

[32] M. Hernandez, G. Epelde, A. Alberdi, R. Cilla, y D. Rankin, «Synthetic data generation for tabular health records: A systematic review», *Neurocomputing*, vol. 493, pp. 28-45, 2022, doi: https://doi.org/10.1016/j.neucom.2022.04.053.

[33] J. Jordon, A. Wilson, y M. van der Schaar, «Synthetic Data: Opening the data floodgates to enable faster, more directed development of machine learning methods», dic. 2020.

[34] T. E. Raghunathan, «Synthetic Data», *Annu Rev Stat Appl*, vol. 8, n.º 1, pp. 129-140, 2021, doi: 10.1146/annurev-statistics-040720-031848.

[35] T. Hesterberg, «Bootstrap», *Wiley Interdiscip Rev Comput Stat*, vol. 3, n.º 6, pp. 497-526, nov. 2011, doi: 10.1002/wics.182.

[36] M. G. Hall y D. C. Alexander, «Convergence and Parameter Choice for Monte-Carlo Simulations of Diffusion MRI», *IEEE Trans Med Imaging*, vol. 28, n.º 9, pp. 1354-1364, sep. 2009, doi: 10.1109/TMI.2009.2015756.

[37] A. Navarro, L. F. Ochoa, y D. Randles, «Monte Carlo-based assessment of PV impacts on real UK low voltage networks», en *2013 IEEE Power & Energy Society General Meeting*, IEEE, 2013, pp. 1-5. doi: 10.1109/PESMG.2013.6672620.

[38] C. Ji, T. Wang, y L. Yin, «Monte-Carlo Methods in Financial Modeling», 2017, pp. 285-317. doi: 10.1007/978-981-10-3307-0_14.

[39] Y. Sun, A. Cuesta-Infante, y K. Veeramachaneni, «Learning Vine Copula Models For Synthetic Data Generation», dic. 2018.

[40] R. K. Yew Low, R. Faff, y K. Aas, «Enhancing mean–variance portfolio selection by modeling distributional asymmetries», *J Econ Bus*, vol. 85, pp. 49-72, 2016, doi: https://doi.org/10.1016/j.jeconbus.2016.01.003.

[41] A. Jindal, I. Shakhat, J. Cardoso, M. Gerndt, y V. Podolskiy, «IAD: Indirect Anomalous VMMs Detection in the Cloud-Based Environment», 2022, pp. 190-201. doi: 10.1007/978-3-031-14135-5_15.

[42] I. Goodfellow, «NIPS 2016 Tutorial: Generative Adversarial Networks», dic. 2016.

[43] K. Bajaj, D. K. Singh, y Mohd. A. Ansari, «Autoencoders Based Deep Learner for Image Denoising», *Procedia Comput Sci*, vol. 171, pp. 1535-1541, 2020, doi: 10.1016/j.procs.2020.04.164.

[44] E. Hosseini-Asl, J. M. Zurada, y O. Nasraoui, «Deep Learning of Part-Based Representation of Data Using Sparse Autoencoders With Nonnegativity Constraints», *IEEE Trans Neural Netw Learn Syst*, vol. 27, n.º 12, pp. 2486-2498, dic. 2016, doi: 10.1109/TNNLS.2015.2479223.

[45] X. Guo, X. Liu, E. Zhu, y J. Yin, «Deep Clustering with Convolutional Autoencoders», 2017, pp. 373-382. doi: 10.1007/978-3-319-70096-0_39.

[46] V. Wan, Y. Agiomyrgiannakis, H. Silen, y J. Vit, «Google's Next-Generation Real-Time Unit-Selection Synthesizer Using Sequence-to-Sequence LSTM-Based Autoencoders», *INTERSPEECH*, pp. 1143-1147, ago. 2017.

[47] D. P. Kingma y M. Welling, «Auto-Encoding Variational Bayes», dic. 2013.

[48] C. Doersch, «Tutorial on Variational Autoencoders», jun. 2016.

[49] S. Kullback y R. A. Leibler, «On Information and Sufficiency», *The Annals of Mathematical Statistics*, vol. 22, n.º 1, pp. 79-86, mar. 1951, doi: 10.1214/aoms/1177729694.

[50] A. Asperti, «Variance Loss in Variational Autoencoders», feb. 2020.

[51] S. Zhao, J. Song, y S. Ermon, «Towards Deeper Understanding of Variational Autoencoding Models», feb. 2017.

[52] L. Cai, H. Gao, y S. Ji, «Multi-Stage Variational Auto-Encoders for Coarse-to-Fine Image Generation», en *Proceedings of the 2019 SIAM International Conference on Data Mining*, Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019, pp. 630-638. doi: 10.1137/1.9781611975673.71.

[53] A. Asperti y M. Trentin, «Balancing Reconstruction Error and Kullback-Leibler Divergence in Variational Autoencoders», *IEEE Access*, vol. 8, pp. 199440-199448, 2020, doi: 10.1109/ACCESS.2020.3034828.

[54] C. P. Burgess *et al.*, «Understanding disentangling in β-VAE», abr. 2018.

[55] R. T. Q. Chen, X. Li, R. B. Grosse, y D. K. Duvenaud, «Isolating Sources of Disentanglement in Variational Autoencoders», en *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, y R. Garnett, Eds., Curran Associates, Inc., 2018. [En línea]. Disponible en: https://proceedings.neurips.cc/paper_files/paper/2018/file/1ee3dfcd8a0645a25a35977997223d22-Paper.pdf

[56] H. Kim y A. Mnih, «Disentangling by Factorising», feb. 2018.

[57] F. Locatello *et al.*, «Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations», nov. 2018.

[58] Y. Wang, D. Blei, y J. P. Cunningham, «Posterior Collapse and Latent Variable Non-identifiability», en *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, y J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 5443-5455. [En línea]. Disponible en: https://proceedings.neurips.cc/paper_files/paper/2021/file/2b6921f2c64dee16ba21ebf17f3c2c92-Paper.pdf

[59] A. Razavi, A. van den Oord, B. Poole, y O. Vinyals, «Preventing Posterior Collapse with delta-VAEs», ene. 2019.

[60] J. Tomczak y M. Welling, «VAE with a VampPrior», en *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, A. Storkey y F. Perez-Cruz, Eds., en Proceedings of Machine Learning Research, vol. 84. PMLR, mar. 2018, pp. 1214-1223. [En línea]. Disponible en: https://proceedings.mlr.press/v84/tomczak18a.html

[61] S. Yeung, A. Kannan, Y. Dauphin, y L. Fei-Fei, «Tackling Over-pruning in Variational Autoencoders», jun. 2017.

[62] Y. Feigin, H. Spitzer, y R. Giryes, «GMM-Based Generative Adversarial Encoder Learning», dic. 2020.

[63] Z. Jiang, Y. Zheng, H. Tan, B. Tang, y H. Zhou, «Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering», nov. 2016.

[64] N. Dilokthanakul *et al.*, «Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders», nov. 2016.

[65] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, y J. M. Tomczak, «Hyperspherical Variational Auto-Encoders», abr. 2018.

[66] M. J. Betancourt y M. Girolami, «Hamiltonian Monte Carlo for Hierarchical Models», dic. 2013.

[67] M. M. Moghadam *et al.*, «Game of GANs: Game-Theoretical Models for Generative Adversarial Networks», jun. 2021.

[68] I. J. Goodfellow *et al.*, «Generative Adversarial Networks», jun. 2014.

[69] J. de la Torre, «Redes Generativas Adversarias (GAN) Fundamentos Teoricos y Aplicaciones», feb. 2023.

[70] L. Metz, B. Poole, D. Pfau, y J. Sohl-Dickstein, «Unrolled Generative Adversarial Networks», nov. 2016.

[71] J. Gui, Z. Sun, Y. Wen, D. Tao, y J. Ye, «A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications», ene. 2020.

[72] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, y X. Chen, «Improved Techniques for Training GANs», jun. 2016.

[73] T. Karras, S. Laine, y T. Aila, «A Style-Based Generator Architecture for Generative Adversarial Networks», dic. 2018.

[74] G. Galindo-Romera, J. Carnerero-Cano, J. J. Martínez-Martínez, y F. J. Herraiz-Martínez, «An IoT Reader for Wireless Passive Electromagnetic Sensors», *Sensors 2017, Vol. 17, Page 693*, vol. 17, n.º 4, p. 693, mar. 2017, doi: 10.3390/S17040693.

[75] G. Van Rossum y F. L. Drake Jr, «Python tutorial», *Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands*, 1995.

[76] C. R. Harris *et al.*, «Array programming with NumPy», *Nature*, vol. 585, n.º 7825, pp. 357-362, sep. 2020, doi: 10.1038/s41586-020-2649-2.

[77] W. Mckinney, «pandas: a Foundational Python Library for Data Analysis and Statistics», *Python High Performance Science Computer*, abr. 2011.

[78] P. M. Meaney, C. J. Fox, S. D. Geimer, y K. D. Paulsen, «Electrical Characterization of Glycerin: Water Mixtures: Implications for Use as a Coupling Medium in Microwave Tomography», *IEEE Trans Microw Theory Tech*, vol. 65, n.º 5, pp. 1471-1478, may 2017, doi: 10.1109/TMTT.2016.2638423.

[79] W. J. Ellison, K. Lamkaouchi, y J. M. Moreau, «Water: a dielectric reference», *J Mol Liq*, vol. 68, n.º 2-3, pp. 171-279, abr. 1996, doi: 10.1016/0167-7322(96)00926-9.

[80] A. Sihvola, «Mixing Rules with Complex Dielectric Coefficients», *Subsurface Sensing Technologies and Applications 2000 1:4*, vol. 1, n.º 4, pp. 393-415, 2000, doi: 10.1023/A:1026511515005.

[81] L. Xu, M. Skoularidou, A. Cuesta-Infante, y K. Veeramachaneni, «Modeling Tabular Data using Conditional GAN», *33rd Conference on Neural Information Processing Systems*, pp. 3-6, oct. 2019, Accedido: 15 de mayo de 2023. [En línea]. Disponible en: https://github.com/DAI-Lab/CTGAN

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los Objetivos de Desarrollo Sostenible (ODS), aprobados por las Naciones Unidas en 2015, representan un plan integral para abordar desafíos globales clave y orientar la construcción de un mundo más justo y sostenible. De los 17 objetivos planteados, este proyecto se alinea principalmente con el Objetivo 9: Industria, Innovación e Infraestructura y el Objetivo 10: Reducción de las Desigualdades.

Ilustración 23: *Objetivos de desarrollo sostenible de las Naciones Unidas.*

Fuente: Cortesía de Naciones Unidas[25]

Objetivo 9: Industria, Innovación e Infraestructura

El proyecto se propone avanzar en la innovación tecnológica al aplicar redes generativas para la clasificación de disoluciones de compuestos orgánicos, una contribución significativa en el

[25] https://www.un.org/sustainabledevelopment/es/news/communications-material/

desarrollo de herramientas de análisis en sectores industriales como la farmacéutica y la cosmética. Se trata de un paso clave hacia la promoción de una infraestructura resiliente, la promoción de una industrialización inclusiva y sostenible y el fomento de la innovación.

El uso de estos modelos promete impulsar la creación de biosensores más económicos y portables. Estos avances pueden facilitar una mayor adaptabilidad en diversos entornos, permitiendo una implementación más sencilla y eficaz de herramientas analíticas. Así, se estaría contribuyendo directamente a la construcción de una industria más sostenible e innovadora que pueda responder a los desafíos del siglo XXI.

Objetivo 10: Reducción de las Desigualdades

Este proyecto también tiene un impacto significativo en el Objetivo 10 al buscar reducir el coste del proceso de entrenamiento de los modelos de clasificación de disoluciones de compuestos orgánicos. Este esfuerzo facilita el acceso a métodos avanzados de detección de biomoléculas en entornos que anteriormente podrían haber tenido barreras financieras para adquirir instrumentación y equipamiento sofisticado.

Al hacer estos métodos más accesibles, se está democratizando el acceso a la tecnología y contribuyendo a reducir las desigualdades en términos de recursos y capacidades técnicas. Este es un paso clave hacia el empoderamiento y la inclusión de todos, independientemente de su estatus económico, garantizando igualdad de oportunidades y reduciendo las disparidades de resultados.

En conclusión, este proyecto se alinea directamente con los Objetivos de Desarrollo Sostenible, impulsando la innovación en la industria y trabajando para reducir las desigualdades en el acceso a la tecnología. En este sentido, refleja el compromiso con el fomento de un desarrollo equitativo, inclusivo y sostenible.

ANEXO II: Repositorio de GitHub

El código implementado en este proyecto está disponible en un repositorio de GitHub, que alberga tanto el Jupyter Notebook desarrollado en VSCode como los conjuntos de datos empleados en el estudio. Este repositorio proporciona una visión detallada y accesible del proceso y técnicas empleadas. Se puede acceder al repositorio en el siguiente enlace: https://github.com/villacampaporta/Modelo-Spectral.