UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM ATLACOMULCO

"Caracterización de especies en plasma frío mediante análisis de espectroscopia de emisión óptica por técnicas de Machine Learning"

T E S I S

Que para obtener el Grado Académico de:

Maestro en Ciencias de la Computación

Presenta:

Octavio Rosales Martínez

Director de Tesis:

Dr. Allan Antonio Flores Fuentes

Octubre de 2020

RESUMEN

La espectroscopía de emisión óptica es una técnica que permite la identificación de elementos químicos usando el espectro electromagnético que emite un plasma. Con base en la literatura. tiene aplicaciones diversas, por ejemplo: en la identificación de entes estelares, para determinar el punto final de los procesos de plasma en la fabricación de semiconductores o bien, específicamente en este trabajo, se tratan espectros para la determinación de elementos presentes en la degradación de compuestos recalcitrantes. En este documento se identifican automáticamente espectros de elementos tales como He, Ar, N, O, y Hg, en sus niveles de energía uno y dos, mediante técnicas de Machine Learning (ML).

En primer lugar, se descargan las líneas de elementos reportadas en el NIST (National Institute of Standards and Technology), después se preprocesan y unifican para los siguientes procesos: a) crear un generador de 84 espectros sintéticos implementado en Python y el módulo ipywidgets de Jupyter Notebook, con las posibilidades de elegir un elemento, nivel de energía, variar la temperatura, anchura a media altura, y normalizar el especto y, b) extraer las líneas para los elementos He, Ar, N, O y Hg en el rango de los 200 nm a 890 nm, posteriormente, se les aplica sobremuestreo para realizar la búsqueda de hiperparámetros para los algoritmos: Decision Tree, Bagging, Random Forest y Extremely Randomized Trees basándose en los principios del diseño de experimentos de aleatorización, replicación, bloqueo y estratificación.

En segundo lugar, se evalúa la métrica F1 de estos algoritmos con validación cruzada en 2, 3, 5, y 10 (cada una repetida 100 veces), después se buscan diferencias en sus distribuciones con la prueba paramétrica de ANOVA y no paramétrica de Friedman, así como su ranking con la prueba de Nemenyi. Posteriormente, estos algoritmos se ensamblan con votación suave, y se determinan el ajuste y la cantidad de datos de entrenamiento, con una curva de aprendizaje, para generar un modelo en el que se calcula el área bajo la curva de cada elemento mediante ROC (Receiver Operating Characteristics) y se generaran las fronteras de decisión para visualizar cómo el modelo final separa los elementos.

En tercer lugar, al modelo ensamblado se realiza: a) un análisis general que muestra el rendimiento del clasificador basado en valores empíricos con variación de paso en longitud de onda anchura a media altura, y temperatura, y b) un análisis específico que establece la exactitud de las predicciones con las características de paso en longitud de onda y anchura a media altura que tienen los espectros experimentales del ININ (Instituto Nacional de Investigaciones Nucleares), así como su tendencia al variar la temperatura.

Para finalizar, se implementa una interfaz de usuario con Python y QT5 que tiene como principales características: a) cargar los espectros experimentales del ININ, b) corregir el desplazamiento óptico por regresión lineal o polinomial, c) corregir el espectro de fondo continuo, d) caracterizar automáticamente las especies de elementos y e) estimar la temperatura del espectro y, f) generar reportes.

Se concluye que es posible realizar un ensamblado de algoritmos basados en árboles de decisión, para realizar identificación automática de especies de los espectros de átomos de He, Ar, N, O, y Hg, de niveles de energía uno y dos. En el análisis específico se alcanza una exactitud media para todas las clases con sus respectivos intervalos de confianza de 95%, dónde en un extremo se tiene la menor exactitud con una media de 0.93905 para Hg I con un paso de 0.028 nm, y en el otro extremo N I, N II y O I, con una exactitud media máxima de 1.0 para al menos un tamaño de paso. El conjunto de datos sintéticos conformado por 972 espectros permitió establecer una idea general del rendimiento del clasificador.

Finalmente, se integró la interfaz de usuario con el modelo de ML de manera exitosa. Es importante destacar que no se tienen antecedentes en la literatura de trabajos similares, por lo que este trabajo es un referente para futuras aplicaciones de ML en el área de física de plasmas mediante el análisis de espectros de emisión óptica.

ABSTRACT

Optical emission spectroscopy is a technique that allows the identification of chemical elements by using the electromagnetic spectrum emitted by a plasma. In agreement with literature, it has diverse applications, for instance: in the identification of stellar entities, to determining the endpoint of plasma processes in the manufacture of semiconductors, or specifically in this work, spectra are processed for the determination of elements present in the degradation of recalcitrant compounds. In this document, spectra of elements such as He, Ar, N, O, and Hg, at their energy levels one and two, are automatically identified by using Machine Learning (ML) techniques.

First, the lines of elements reported by the NIST (National Institute of Standards and Technology) are downloaded, preprocessed, and unified for the following processes: a) create a generator of 84 synthetic spectra implemented in Python and the ipywidgets module from Jupyter Notebook, with the possibilities to choose an element, energy level, vary the temperature, width at half height, and normalize the spectrum and, b) extract the lines for the elements He, Ar, N, O and Hg in the range of 200 nm to 890 nm, subsequently, they are oversampled to perform the hyperparameter search for the Decision Tree, Bagging, Random Forest and Extremely Randomized Trees algorithms based on the principles of experiment design: randomization, replication, blocking and stratification.

Second, the F1 metric of these cross-validation algorithms is evaluated in 2, 3, 5, and 10, (each one repeated 100 times) then differences are found for their distributions with the parametric ANOVA and non-parametric Friedman test, as well as their ranking with the Nemenyi test. Subsequently, these algorithms are assembled with soft-voting, and the fitness as well as the amount of training data are determined with a learning curve, to generate a model that calculates the area under the curve of each element with ROC (Receiver Operating Characteristics), and the decision borders will be generated to visualize how the final model separates the elements.

Third, to the assembled model is performed as follows: a) a general analysis that shows the performance of the classifier based on empirical values with step variation in wavelength, width at half-height, and temperature and, b) a specific analysis that establishes the accuracy of predictions with the characteristics of the step in wavelength

and width at a half-height that have the experimental spectra of the Plasma Physics Laboratory of ININ (National Institute of Nuclear Research), as well as the trend when the temperature is varied.

At the end, a user interface is implemented with Python and QT5 that has the following main characteristics: a) load the experimental spectra of the ININ, b) correct the optical displacement by linear or polynomial regression, c) correct the continuous background spectrum, d) the automatic characterization of the element species, e) to estimate the temperature of the spectrum and, f) generate reports.

It is concluded that it is possible to carry out an assembly of algorithms based on decision trees, to perform automatic identification of species from the spectra of He, Ar, N, O, and Hg atoms, of energy levels one and two. In the specific analysis, a mean accuracy is reached for all classes with their respective 95% confidence intervals, where at one extreme there is the lowest accuracy with a mean of 0.93905 for Hg I with a step of 0.028 nm, and at the other extreme NI, N II and OI, with a greatest mean accuracy of 1.0 for at least one step size. The synthetic data set consisting of 972 spectra, allowed to establish a state of the performance of the classifier.

Finally, the user interface was successfully integrated with the ML model. It is important to highlight that there are no antecedents in the literature about similar works, therefore this is a reference for future applications of ML in the area of plasma physics specifically in the analysis of optical emission spectroscopy.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La espectroscopia de emisión óptica en plasma no térmico es una técnica no invasiva en la que se identifican especies de elementos en gases ionizados utilizando su espectro [1], en el Instituto Nacional de Investigaciones Nucleares se utiliza para determinar los elementos presentes en la degradación de compuestos recalcitrantes [2], [3].

Hasta el momento se encontraron en la literatura trabajos afines dónde se utilizan espectros y redes neuronales para la identificación de radioisótopos de rayos gamma, así como la identificación de elementos químicos en aerosoles. Por otra parte, se encuentra el uso de técnicas de Machine Learning para la clasificación de entes estelares y galaxias. Sin embargo, no se encontraron trabajos que identifiquen especies de elementos provenientes de un espectro de emisión óptica en plasma frio. El poder encontrar una solución a esta problemática es lo que motiva el desarrollo de este trabajo.

Dentro de las alternativas para resolver este tema, se siguió por el camino de las técnicas de Machine Learning en lugar de las Redes Neuronales por tratarse de un área poco explorada en temas de espectroscopia, es así como se parte de una hipótesis de trabajo que plantea el poder caracterizar especies generadas en un plasma frío mediante espectros de emisión haciendo uso de la espectroscopia de emisión óptica en un intervalo de longitud de onda acorde con el espectrómetro utilizado en el Instituto Nacional de Investigaciones Nucleares que comprende de los 200 nm a 890 nm con técnicas de Machine Learning, se plantea una exactitud mayor o igual al 70% porque no se tienen referentes previos.

La estructura de esta tesis es la siguiente:

En el Capítulo 1 se define el problema y se establecen los objetivos, justificación, impactos e hipótesis. También se establece el uso de los lenguajes de programación Python y QT5 y las bibliotecas de funciones que se emplearon para el procesamiento, almacenamiento, predicción y graficación de datos.

El Capítulo 2 corresponde al estado del arte donde se encuentra una introducción a la espectroscopia de emisión óptica y a las técnicas de caracterización basadas en Machine Learning. También se integra una matriz con las referencias consultadas con su discusión.

El Capítulo 3 es metodología, que abarca desde la creación del generador de espectros sintéticos en Jupyter, la caracterización automática de especies y estimación de la temperatura electrónica integrados en una interfaz gráfica implementada en QT5.

El Capítulo 4 contiene la experimentación basada en los principios de aleatorización, replicación, blocking y estratificación; se establece la configuración del experimento con validación cruzada y repetición, y se usan técnicas estadísticas de ANOVA, Friedman y Nemenyi, así como curvas de validación, curvas de Características Operativas del Receptor y fronteras de decisión.

En el Capítulo 5 se realiza un análisis general y específico de la exactitud en las predicciones con la variación de ***Paso***, ***Anchura*** y ***Temperatura*** en los espectros sintéticos, y se valida la hipótesis con un intervalo de confianza del 95% que las predicciones en la caracterización automática de especies se encuentran en un rango de exactitud que va del 0.93905 a 1.0.

Para concretar este trabajo se contó con la orientación por parte del Laboratorio de Física de Plasmas del Instituto Nacional de Investigaciones Nucleares, quienes a su vez proporcionaron espectros de una lámpara de calibración que justificó la necesidad de realizar correcciones como el desplazamiento óptico y el espectro de fondo continuo, así como la validación de la construcción de un generador de espectros sintéticos en Jupyter Notebook, y la implementación de una interfaz gráfica en QT5 para la caracterización automática de especies basado en los requerimientos de usuario.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Definición del problema.

En el Laboratorio de Física de Plasmas del Instituto Nacional de Investigaciones Nucleares (ININ) se realizan descargas de Barrera Dieléctrica para generar plasma no térmico a presión atmosférica. Este tipo de descargas se realiza con gases tales como: Helio (He), Argón (Ar), Nitrógeno (N), Oxigeno (O), mezclados o no con algunos otros compuestos para su degradación. Cabe resaltar que N y O no existen como tal en estado natural, sino como N_2 y O_2, y para fines prácticos en este documento se indican como N y O.

Actualmente, para caracterizar las descargas generadas, se utilizan técnicas eléctricas, químicas y ópticas; esta última mediante espectroscopia de emisión. Esta técnica consiste en primer lugar en usar una fibra óptica para captar y transportar la emisión luminosa del plasma hacia la rejilla de entrada de un espectrómetro que descompondrá por difracción, la luz en sus diversas longitudes de onda que la componen para ser amplificadas por una cámara de dispositivo de carga acoplada (CCD por sus siglas en inglés). Ésta proporciona datos que son capturados con el software WinSpec32™ que posteriormente, después de ser transformados en el formato adecuado, serán graficados en software como Origin™, entre otros.

El objetivo de este procedimiento es analizarlos, identificar las líneas o bandas espectrales, obtener el área bajo la curva de una línea seleccionada y con esta información realizar un cálculo manual utilizando expresiones matemáticas como la ecuación de Boltzmann. Para cada línea seleccionada, existen datos de la transición y excitación, estos son colectados en una base de datos del Instituto Nacional de Estándar y Tecnología (NIST por sus siglas en inglés). Para el análisis, graficación y calculo es deseable y necesario disponer de una herramienta por medio de una interfaz gráfica desarrollada a la medida que permita llevar a cabo el tratamiento de datos a partir de los archivos con extensión, CSV (Comma Separated Values por su significado en inglés), los cuales son generados por WinSpec32™ de la forma más sistemática, ordenada y controlada, sin perder la rigurosidad de análisis.

Por lo anteriormente expuesto, se plantea la siguiente pregunta: ¿es posible caracterizar especies de plasma frío mediante análisis de espectros de emisión colectados por espectroscopia de emisión óptica con técnicas de Machine Learning? Para responderla se propone realizar una interfaz de usuario en el lenguaje de programación Python y QT5, para el Sistema Operativo Windows 8 de 64 bits, la caracterización automática de elementos provenientes de espectroscopia de emisión óptica se realiza con algoritmos no lineales de aprendizaje supervisado, y la corrección del desplazamiento óptico con algoritmos lineales de aprendizaje supervisado. Esto se implementa integrando tecnologías de software libre como: *pandas* por sus estructuras de datos, *numpy* y su manejo de arreglos n-dimensionales, *scikit-learn* porque está orientado al Machine Learning, *scipy* y sus bibliotecas de funciones para cómputo científico, *matplotlib* por sus métodos para graficar en 2D/3D, y *pickle* para almacenar objetos optimizados para usarse como repositorio de datos local de las especies de elementos reportadas en el NIST.

1.2 Objetivos de investigación.

1.2.1 Objetivo general.

Implementar técnicas y algoritmos de Machine Learning para determinar la posibilidad de caracterizar automáticamente de líneas de átomos excitados resultantes de la espectroscopia óptica de emisión de un plasma frío generado por descarga de barrera dieléctrica en un gas puro o en la degradación de compuestos recalcitrantes con datos del Laboratorio de Física de Plasmas del ININ.

1.2.2 Objetivos específicos.

1) Implementar un repositorio local con líneas espectrales de especies reportadas en el NIST.
2) Implementar técnicas de Machine Learning para la caracterización automática de las siguientes especies: He, N, O, Ar y Hg, resultantes de la espectroscopia óptica de emisión de plasma frío en sus niveles I y II.
3) Identificar la exactitud de las predicciones en la caracterización automática de especies provenientes de un plasma frio obtenidos por espectroscopia de emisión óptica.

4) Usar las líneas de especies reportadas en NIST como datos de entrenamiento para los algoritmos de Machine Learning.
5) Medir la exactitud de predicciones en espectros con distintas condiciones de paso, anchura y temperatura.
6) Corregir desplazamiento óptico de los espectros experimentales con referencia a espectros obtenidos de la lámpara de calibración HG-1 de la empresa Ocean Optics™.
7) Ajustar los hiperparámetros de los algoritmos elegidos con base a metodologías de Machine Learnig.
8) Validar y probar el modelo generado de Machine Learning.
9) Implementar una interfaz gráfica de usuario con las funcionalidades de carga de espectros para su graficación, caracterización y generación de reporte.
10) Estimar la temperatura de excitación electrónica de los elementos caracterizados.

1.2.3 Preguntas de investigación.

1) ¿Qué técnica permite extraer datos de las líneas espectrales de especias reportadas en el NIST?
2) ¿Cuáles son las técnicas de Machine Learning que permiten la caracterización automática de especies resultantes de la espectroscopía óptica de emisión de plasma frío en sus niveles I y II?
3) ¿Qué métricas se pueden emplear para medir la exactitud de las predicciones en la caracterización de especies provenientes de un plasma frio obtenidos por espectroscopia de emisión óptica?
4) ¿De qué manera se pueden emplear las líneas de especies reportadas en el NIST para entrenar los algoritmos de Machine Learning?
5) ¿Cómo se pueden crear espectros sintéticos de He, N, O, Ar y Hg en distintas condiciones de paso, anchura y temperatura para medir la exactitud del modelo predictor?
6) ¿Con qué procedimiento se puede corregir un desplazamiento óptico?
7) ¿Cómo se realiza la optimización de hiperparámetros?
8) ¿De qué manera se puede validar y probar el modelo generado de Machine Learning?

9) ¿Cómo se implementa una interfaz gráfica para los algoritmos de Machine Learning, mostrar las gráficas del espectro y generar un reporte?
10) ¿Cuál es el procedimiento para estimar la temperatura de excitación electrónica de un espectro de emisión óptica?

1.3 Justificación.

el Laboratorio de Física de Plasmas del ININ realiza experimentos con plasma frio generado mediante descargas eléctricas de barrera dieléctrica para la degradación de compuestos recalcitrantes, con el objetivo de realizar aportes que contribuyen a la mejora del medio ambiente. El tratamiento de los espectros para realizar cálculos para estimar propiedades puede llevar desde horas hasta días, ya que durante la degradación de compuestos recalcitrantes se capturan datos resultantes de la espectroscopía de emisión óptica, mismos que deben ser procesados y analizados. Esta etapa de análisis requiere de cálculos manuales que contribuyen a alargar los tiempos de investigación.

Actualmente, la implementación de técnicas de Machine Learning, ha mostrado ser eficiente en la automatización del reconocimiento de patrones, algunos trabajos afines, desarrollados y presentados en el estado del arte muestran resultados satisfactorios. Sin embargo, la motivación de este trabajo de investigación se centra específicamente en el tratamiento de datos provenientes de espectroscopia de emisión óptica para la caracterización automática de especies, no se han identificado trabajos de Machine Learning en este campo en particular, por lo que la aportación de este trabajo se llevará en esta área de conocimiento.

Es así como se propone el uso de algoritmos de Machine Learning para la caracterización de especies con un repositorio local de los datos de las especies reportadas por el NIST, además de una interfaz gráfica de usuario, para agilizar las investigaciones recortando los tiempos al realizar consultas locales. El rendimiento del modelo generado de Machine Learning como herramienta de identificación automática, dependerá en gran medida del; a) preprocesamiento de datos, b) balanceo de las líneas de las especies consideradas, c) cantidad de datos empleados en las etapas de entrenamiento, d) búsqueda de hiperparámetros (variable de configuración del modelo cuyo valor no puede ser estimado

a partir de los datos), e) el tamaño del paso, anchura y temperatura de los espectros sintéticos generados, f) corrección del desplazamiento óptico del espectro experimental, y g) de las características del espectrómetro utilizado.

Como característica adicional de la interfaz gráfica, en este trabajo se presenta un primer acercamiento para la estimación automática de temperatura, esta requiere de un conjunto de especies identificadas en los extremos del espectro que cuenten con todos los parámetros reportados en el NIST para su substitución en la ecuaciones que forman parte de la estimación de la temperatura de excitación electrónica (3.17), (3.18), (3.19), (3.20) y (3.21) y (ver sección 3.3.2). La elección de estas especies requiere de la pericia y experiencia del usuario, por esto se implementa la posibilidad de generar un reporte de las especies detectadas para facilitar su cálculo manual en caso de que el software diseñado no realice la estimación correctamente.

1.4 Impactos.

Científico: Generar una metodología para la caracterización automática de especies, en espectroscopia de emisión óptica, ya sea en una descarga de gas puro o en la degradación de compuestos recalcitrantes, implementando técnicas de Machine Learning.

Tecnológico: Se aplican tecnologías emergentes como Machine Learning, para la implementación de un software científico con interfaz de usuario para agilizar la selección, preprocesamiento y análisis de datos provenientes de espectros de emisión óptica en el intervalo de 200 nm a 890 nm que corresponde al ultravioleta (UV) y parte del visible (Vis) mediante la caracterización automática de especies.

1.5 Hipótesis.

Mediante la incorporación de técnicas de Machine Learning en una interfaz de usuario se podrán caracterizar automáticamente especies de plasma frío en espectroscopia de emisión óptica con una exactitud mayor o igual al 70%, en un intervalo de longitud de onda de 200 nm a 890 nm.

2 ESTADO DEL ARTE

En esta sección del documento se presentan trabajos de investigación que exponen el estado actual de las investigaciones para evaluar la caracterización de diferentes tipos de especies de manera automática, implementando técnicas de Machine Learning. Se presentan los conceptos fundamentales teóricos no sólo del contexto de Machine Learninig, sino también del cómputo científico aplicado al área de espectroscopía de emisión óptica y aplicaciones.

2.1 Fondo continuo, línea base y balanceo de datos.

En el Laboratorio de Física de Plasmas del ININ se llevan a cabo experimentos de espectroscopía de emisión óptica, a partir de estos experimentos se obtienen datos de espectros que requieren un análisis por parte del usuario, en los que invierte desde horas hasta semanas. Un análisis previo de los espectros proporcionados por el Laboratorio de Física de Plasmas del ININ mostró que los espectros están montados sobre un fondo continuo, en la literatura, se presenta información dónde se calcula una línea base (equivalente al espectro de fondo continuo) con una distorsión mínima de los picos. Se trata de un método corrección de la línea base por el método de suavizado de mínimos cuadrados asimétricos, el cual es rápido, flexible y ajustable, además converge en un rango de 5 a 7 iteraciones. Este método requiere del juicio humano por inspección visual para determinar la validez de la línea base generada [4].

En la Figura 2.1.1 se muestra en color azul el espectro experimental de la lámpara de calibración HG-1 sin corrección alguna y en color anaranjado el espectro de fondo continuo a una altura con intensidad promedio de 613 u.a.

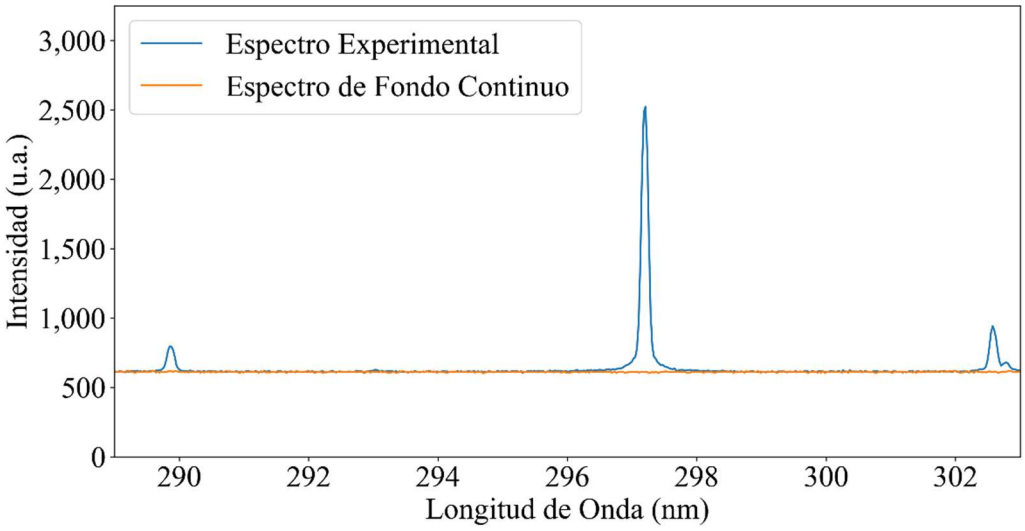


Figura 2.1.1. Espectro experimental y espectro de fondo continuo.

Por otra parte, se observó que los datos obtenidos del NIST están desbalanceados, es decir, hay especies de elementos que superan en cantidad a otros, lo que ocasiona que el modelo de ajuste generado realice predicciones incorrectas en las especies minoritarias (tratado con detalle en sección 3.2.2). En la búsqueda de una métrica apropiada para resolver esta problemática se encontró un estudio que mide la efectividad de las predicciones mediante la métrica F1, en la cual todos los atributos tienen el mismo peso y cuyo cálculo parte de una matriz de confusión (tratado con detalle en sección 4.3). Esta métrica tiene la ventaja de mostrar cuando una clase minoritaria no es predicha correctamente al disminuir la métrica F1 macro, situación que no ocurre cuando sólo se calcula el promedio [5].

2.2 Espectroscopía óptica.

La espectroscopia de emisión óptica es una técnica no invasiva; es decir, no cambia las características del plasma que se usa para la caracterización de especies a través de la luz que emiten gases excitados o ionizados [1], también es posible caracterizar especies sólidas evaporando su superficie y analizando el espectro resultante [6]. En [7] esta técnica se utiliza para estimar la temperatura de excitación en las diferentes líneas espectrales de un plasma de argón y neón.

En el Laboratorio de Física de Plasmas del ININ se captura la radiación luminosa emitida por un plasma utilizando fibra óptica, espectrómetro, equipo de cómputo y software, generando un archivo de datos representativo del espectro que deberá ser analizado manualmente para determinar especies de elementos, temperatura electrónica, vibracional y rotacional, y su densidad.

Otros aportes en esta área permiten determinar la temperatura de excitación y la densidad del aire en flamas premezcladas de acetileno y aire, la determinación de la temperatura electrónica se hace con la ecuación de Boltzmann, a partir de una línea espectral y un software que soporta la regresión lineal llamado SigmaPlot. La regresión lineal se emplea para verificar que la temperatura encontrada corresponda a la especie de estudio [8].

En otro trabajo reportado, se obtienen la emisión de espectros de OH desde una descarga corona pulsada de alto voltaje con una mezcla de gas N_2 y aire húmedo H_2O, se usa la descarga corona pulsada porque produce bajas temperaturas de gas y alta temperatura de electrones. Se estudian especies con fuerte reactividad como OH, O, H, N, HO_2, N^+, N_2^+, O_3 porque pueden remover gases ácidos de gases de combustión, eliminan compuestos orgánicos del agua, realizan descontaminación bacteriana y destruyen o descomponen otros componentes tóxicos; se presta especial atención al radical OH por su papel principal en la oxidación de muchos procesos fisicoquímicos. Se describe detalladamente el hardware empleado y se muestra el esquemático del experimento (ver Figura 2.2.1) [9].

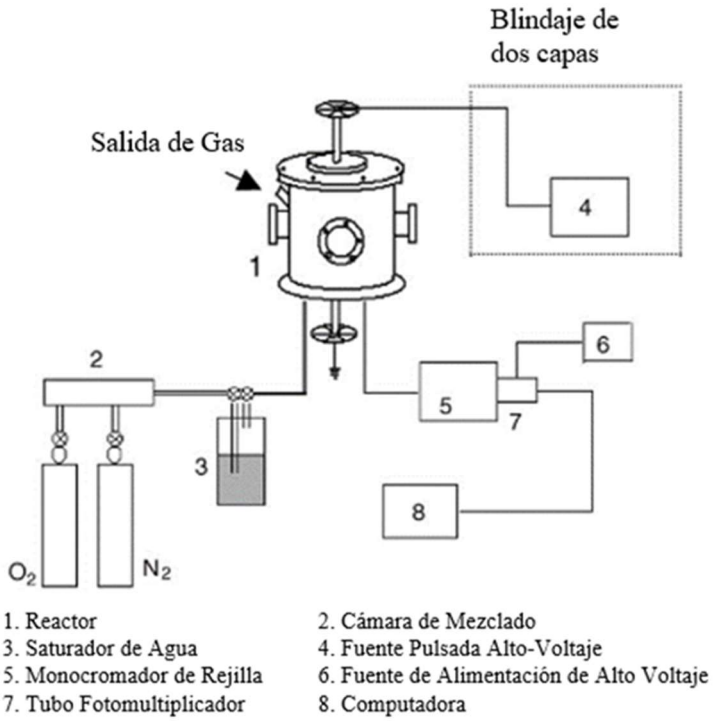


Figura 2.2.1. Esquema de la Configuración del Experimento [9].

Otras aplicaciones en plasma, por ejemplo, en un jet de plasma generado a presión atmosférica se utilizan mezclas de Ar y H_2O al 0.05%, en este, se emplean tubos capilares de 1.3mm y 3mm a una frecuencia de 71 kHz, un voltaje pico entre 12.2 y 17 kV_{p-p} y una fuente de potencia de 12.8 W. Este estudio concluye que la temperatura del gas se incrementa con la adición de H_2O al Ar, desde 265 K hasta 1,125 K, con una intensidad máxima de radicales OH [10].

En el trabajo presentado en [11], se estima la temperatura electrónica del Plasma por medio de la ecuación de Boltzmann y se consulta la base de datos de especies de elementos del NIST para ser almacenada en disco duro y posteriormente traducida y almacenada en la base de datos local no relacional con MySQL. El sistema es capaz de estimar la temperatura electrónica en el pico de una especie de elemento con la intervención del usuario mediante la elección de tres puntos: un punto máximo equivalente a la longitud de onda observada y otros dos puntos mínimos como base. Para lograrlo, se hace uso de la biblioteca JFreeChart y se limita a leer archivos de Excel con la biblioteca JExcelApi. Para calcular el área bajo la curva usa el método del trapecio y para suavizar la curva usa opcionalmente el filtro media-móvil y el tamaño de la ventana lo determina el usuario,

finalmente interpola como mínimo dos veces la estimación de la temperatura electrónica con el fin de reducir el error de estimación.

2.3 Técnicas de caracterización basadas en Machine Learning.

Machine Learning o Aprendizaje Automático es un área de conocimiento de la Inteligencia Artificial (IA), que a través de un proceso de inferencia algorítmica y estadística se puede encontrar el patrón de un conjunto de datos. Para este proceso se utilizan datos de entrenamiento como entrada de una instancia en un algoritmo, posteriormente se crea un modelo de ajuste para los datos y finalmente se compara la salida del modelo con los datos de prueba para evaluar su rendimiento.

En Machine Learning se encuentran las categorías de algoritmos: regresión, clustering, reducción de la dimensionalidad y clasificación. Los algoritmos de regresión se centran en variables continuas conocidas, aquí destacan algoritmos como: lineal, lasso, ridge y support vector regression. Por otra parte, en los algoritmos de clustering no se tiene una clase, pero se conoce a priori la cantidad de grupos de datos, algunos algoritmos son: k-means y DBSCAN. También se encuentran los algoritmos de reducción de dimensionalidad, donde el número de características se reduce para mejorar los tiempos y las predicciones, se mencionan en la literatura algoritmos como: manifold learning, factor analysis y principal component analysis. Finalmente se encuentran los algoritmos de clasificación, estos se distinguen en el uso de datos cuya clase es conocida y discreta o categórica, sobresalen algoritmos como: decision trees, random forest, naive bayes y logistic regression [12], [13].

En otro contexto, se implementan técnicas de inteligencia artificial con redes neuronales artificiales de propagación hacia atrás para la identificación de 28 radioisótopos en rayos gamma que permiten realizar la tarea de caracterización automática de especies. La identificación de los picos en el espectro se obtiene con la segunda derivada y solo se consideraron picos de energía mayores a 1.5 keV. La estructura de la red neuronal (Figura 2.3.1) tiene 47 neuronas de entrada, 52 neuronas para la capa oculta (determinadas por ensayo y error) y 28 neuronas de salida (cada una corresponde a un radioisótopo), esta red fue entrenada con 409 conjuntos de datos de entrada y con un número total de iteraciones

de 500,000. Cada neurona de salida tiene un peso que oscila entre 0 y 1, solo se consideraron aquellas especies que excedían un peso ($W^{n}_{i,j}$) mayor o igual a 0.8, donde i es la neurona de la capa anterior y j es la neurona de la capa superior, y n es la capa de la red neuronal [14].

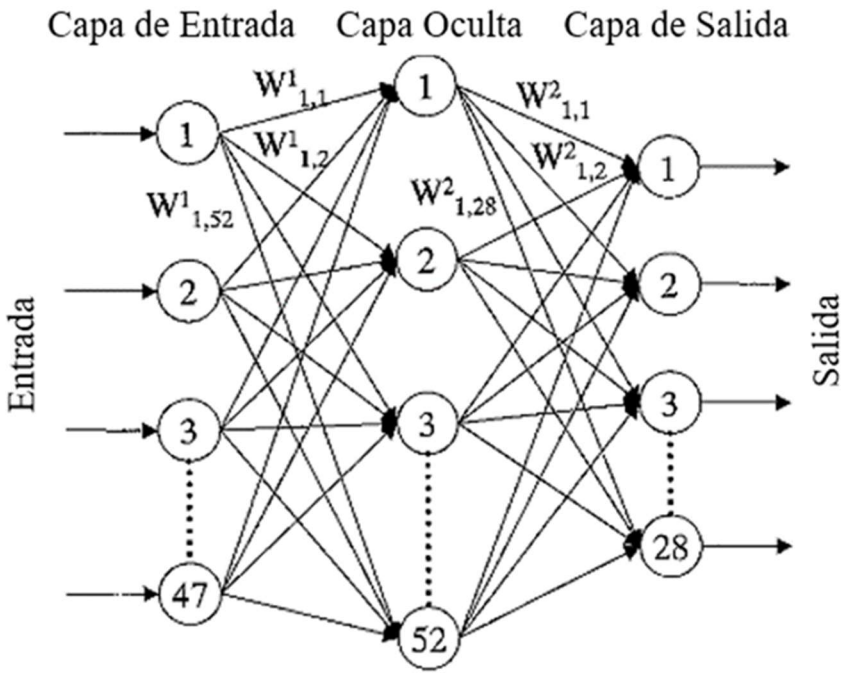


Figura 2.3.1. Estructura de la red neuronal usada en el análisis de rayos gamma [14].

Esta red neuronal tuvo problemas con el ruido al identificar radioisótopos con precisiones en la identificación de especies que oscilaban desde el 21% hasta el 99%, en la Figura 2.3.2 se muestra dos espectros de uranio dentro y fuera de la zona del reactor.

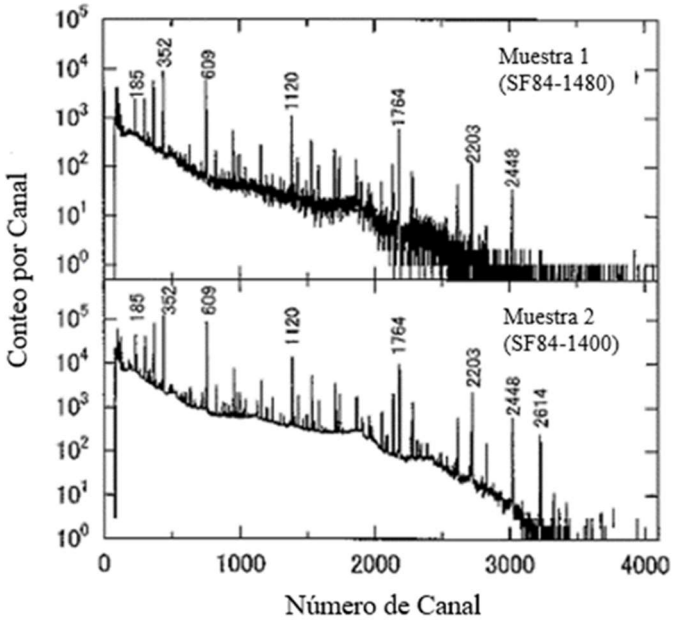


Figura 2.3.2. Espectros de rayos gamma para uranio (Muestra 1: Zona del Reactor; Muestra II: fuera de la zona del reactor) [14].

Otros estudios emplean una red neuronal con propagación hacia adelante y entrenamiento de retro propagación del error. Un conjunto de datos de 22 espectros de muestras orgánicas (18 para entrenamiento y 4 para la prueba) y 37 espectros de aerosoles (29 para el entrenamiento y 8 para la prueba), resalta que se desarrollan tantas Redes Neuronales por cada elemento químico de interés [15]. La metodología para la obtención de resultados, consiste en emplear las Emisiones de Rayos X Inducidas por Protones (Proton Induced X-ray Emissions, PIXE), y disparar un haz de protones de 3.5MeV producido por un generador de Van de Graff que pasan por un imán deflector que lo dirigirá a una muestra, la cual al interactuar con el haz de protones emitirá Rayos X y se arrancarán electrones (detectados con un sensor de Na(I) o uno de Si(Li)) de las capas más ligadas a los átomos de los diferentes elementos, generando vacancias, espacios vacíos en dichas capas) que posteriormente se ocupan por electrones de capas superiores. Durante el proceso de transición entre un electrón arrancado y otro electrón que ocupa su lugar se produce una emisión X característica de un átomo excitado. Finalmente, los protones que atraviesan la muestra se recopilan con una caja de Faraday como se observa en la Figura 2.3.3 y las emisiones de rayos X capturadas por el detector son caracterizadas por la red neuronal.

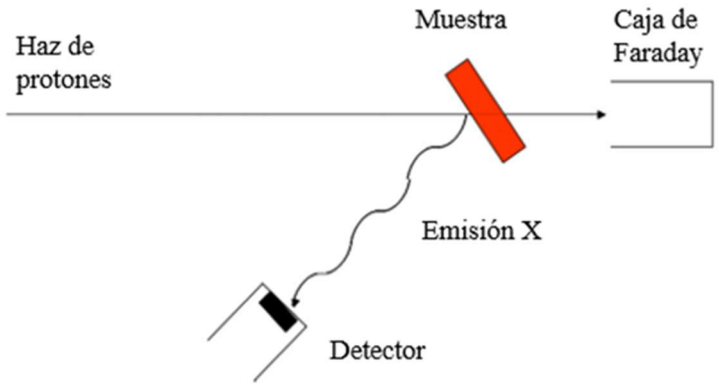


Figura 2.3.3. Diagrama del proceso del gas iónico con una muestra [15].

Otros estudios referentes al análisis de espectros con diferentes campos de aplicación que utilizan algoritmos de Machine Learning se basan en árboles de decisión.

En el estudio presentado por O. Miettinen, la clasificación de objetos estelares con 8 algoritmos de Machine Learning, muestra que los porcentajes de predicción más altos se alcanzan utilizando la técnica nombrada Random Forest con 81% y Gradient Boosting con 82%, esto se realiza con 80% de los datos para entrenamiento y el 20% restante para prueba con el uso de *10 fold cross validation* en datos no balanceados cuyo rango de proporción en la clase a predecir oscila desde el 1% hasta el 37.95%. Es de destacar la influencia del número reducido de características para minimizar sobreajuste y aumentar precisión en la predicción [16].

En otro trabajo afín, se implementan 18 algoritmos de Machine Learning, y destacan Random Forest y Coarse Gaussian SVM (Support Vector Machine, por sus siglas en inglés) en la clasificación de galaxias mediante sus datos espectroscópicos, esto se logra con 4.4 millones de registros para alcanzar una precisión de predicción del 99.2%. Un aspecto para resaltar en cuanto a tiempos es que Random Forest tarda horas en comparación con Coarse Gaussian SVM que tarda una semana. En este estudio también se concluye que aquellos espectros que son similares dificultan la diferenciación espectral [17].

El algoritmo Random Forest mencionado en los trabajos anteriores [16], [17], se basa en árboles de decisión. En primer lugar se destaca la elección de buenas características y su influencia en la ganancia de información, así como la velocidad y precisión de los árboles

de decisión, y menciona las 3 dimensiones que conforman un sistema de Machine Learning para que pueda realizar tareas de clasificación, las cuales son: estrategia usada, representación del conocimiento y el dominio de aplicación del sistema [18]. En segundo lugar, se muestra un ejemplo de clasificación binaria para salir un sábado por la noche considerando como características predictoras: *outlook*, *temperature*, *humidity* y *windy* [18].

Además, el algoritmo Random Forest consiste en construir árboles que toman características y muestras al azar de los datos de entrenamiento, el resultado de la predicción es por votación, es decir, la clase ganadora es aquella que obtuvo la frecuencia más alta en el bosque generado. Por lo tanto, en este tipo de algoritmo el sobreajuste no es un problema porque cada árbol es independiente del resto, y por ley de los grandes números se llega a una buena predicción. Finalmente, otra ventaja del Random Forest es la ponderación interna de cada característica de entrenamiento, lo que permite crear árboles más reducidos y genéricos. Se concluye que, el incremento en el número de características en los datos de entrenamiento induce más error, de la misma manera que lo hacen las características que están altamente correlacionadas por su redundancia de información [19].

En lo que respecta a la calidad y tamaño de los datos de entrenamiento y su influencia en el rendimiento de un árbol de decisión y algoritmos afines, se menciona que las técnicas de poda controlan el tamaño del árbol para combatir el sobreajuste y que la selección aleatoria de datos de entrenamiento influye directamente en el tamaño del árbol. En los datos de entrenamiento el número de muestras y su relevancia influyen en el rendimiento del modelo, de no tratarse, se puede presentar el sobreajuste como una desventaja observable en un árbol de decisión, siendo sus principales causas: valores atípicos, ruido y datos no etiquetados [20].

Dentro de las mejoras aplicables a un árbol de decisión se encuentra la simplificación, ésta aumenta la precisión en las predicciones. Para conseguirlo, los datos con la eliminación de características y muestras irrelevantes se reducen del conjunto de datos de entrenamiento, esta medida de prevención de sobreajuste evita la creación de ramas que reducen la capacidad de generalización del árbol de decisión. La comprensión de los

modelos de Machine Learning es importante para llevar a cabo técnicas que aumenten su precisión y rendimiento [20].

Un problema recurrente en la aplicación de técnicas de Machine Learning, es hallar datos no balanceados. Un estudio muestra que en árboles de decisión con este tipo de datos provoca un rendimiento pobre y propone el uso de la distancia de *Hellinger* como una acción correctiva, dando lugar a nodos con hojas de mayor pureza. La distancia de *Hellinger* es usada para cuantificar la similaridad entre dos distribuciones probabilísticas. En este documento la metodología propuesta consiste en realizar experimentos con técnicas de sub-muestreo y sobre-muestreo SMOTE (Synthetic Minority Oversampling Technique) en 19 conjuntos de datos no balanceados y con el uso de 4 clasificadores basados en árboles. Concluyen que el uso de la distancia de *Hellinger* muestra resultados satisfactorios y apunta que no es necesario tratar el balanceo de datos en árboles de decisión con técnicas de sub-muestreo y sobre-muestreo [21].

Otro problema que se presenta en los árboles de decisión es el sobreajuste, para solucionarlo existen técnicas de poda. Un estudio basado en el software Weka y árboles de decisión muestra que la pre-poda limita los parámetros de construcción del árbol como son la profundidad y el número de observaciones, y posteriormente se construye el árbol, por otra parte, en la post-poda primero se construye el árbol, se analiza la estructura del árbol generado y se procede con la poda. La finalidad de estos procedimientos es generalizar la estructura de clasificación al disminuir la complejidad excesiva para después aplicarla a datos nuevos y obtener buenas predicciones [22].

En contraste con el estudio anterior, otros muestran que la construcción de un árbol de decisión robusto basado en datos no balanceados, las técnicas de poda no funcionan bien y al igual que en [21], para resolverlo se usa *Hellinger* como criterio de división, además muestra como la entropía es insensible al sesgo de la distribución de los datos. En este trabajo se recomienda podar, sólo si las hojas no son significativas o cuando se requiere reemplazar la fracción de una rama por una hoja. Los experimentos fueron realizados con el software Weka y datos de prueba que tienen como variable predictora una clase con 2 atributos, y como estimador AUROC (Area Under the Receiver Operative Characteristics, por sus siglas en inglés) representado con gráficas isométricas [23].

El K-nearest Neighbor es otro tipo de algoritmo de clasificación, el cual propone una distancia ponderada dónde los vecinos más cercanos tienen mayor peso. Este algoritmo elige el atributo a predecir por votación y resalta que un valor de K vecinos muy pequeño o alto provoca predicciones ambiguas, dada la sensibilidad del parámetro K, emplean 20 *fold cross validation* como técnica para determinar su valor óptimo [24]. Por otra parte, otros trabajos proponen la relación entre una observación y su distancia respecto al cluster, y se indica que el error depende del tamaño del conjunto de entrenamiento, y para no afectar las predicciones, al momento de instanciar este algoritmo, el número de categorías ***n*** no debe exceder el número de clusters ***k*** [25].

2.4 Lenguaje de programación para cómputo científico.

La elección del lenguaje de programación se basó en criterios tales como; orientado a objetos, suficientemente rápido, flexible, sintaxis expresiva, *open source*, multiplataforma, portable, extensible, tipado dinámico, modular y de propósito general. Python posee este tipo de características, además de que es ampliamente usado en aplicaciones científicas y proyectos. Además, Python cuenta con bibliotecas de propósito general, con un enfoque orientado al cómputo científico, tal como; NumPy, SciPy, SymPy, Pandas, Matplotlib y los Science Kits, que pueden interactuar entre sí para generar aplicaciones potentes y de calidad [26], [27].

Jupyter Notebook es un entorno de trabajo para lenguajes de programación como Julia, Python, R, entre otros. Entre sus principales características se encuentran la ejecución de código en celdas, lenguaje de marcado propio, e interactividad [28]. Una de sus principales ventajas es el desarrollo ágil de prototipos, sin embargo, ante la imposibilidad de utilizar los ipywidgets de Jupyter para crear archivos ejecutables como aplicaciones finales de usuario, se utilizan en su lugar alternativas de software libre como Tkinter, WxPython, PyQt, PtGUI entre otros [29].

2.5 Discusión basada en matriz de referencias.

En la Tabla 2.5.1 se muestra la matriz de referencias sobre los trabajos más relevantes encontrados en el estudio del estado del arte.

Tabla 2.5.1. Matriz de referencias que se consultaron en esta tesis para el estado del arte.

Referencia	**Resumen**	**Metodología empleada**	**Áreas de Oportunidad**
N. Tkachenko, *Optical Spectroscopy: Methods and Instrumentations*, First. Tampere: Elsevier, 2006.	La espectroscopia óptica de emisión es una técnica no invasiva que por consecuencia no modifica las propiedades del plasma basándose en un espectro luminoso.	Es un libro con procedimientos e instrumentos de laboratorio para la espectroscopia óptica.	Potencialmente una fuente de consulta.
E. Restrepo and A. Devia, "Caracterización de materiales utilizando la espectroscopía óptica de emisión," *Rev. Colomb. Física*, vol. 34, no. 2, pp. 478–483, 2002.	Utiliza espectroscopia de emisión óptica para capturar la radiación luminosa emitida por un plasma capturado con un arreglo óptico y electrónica y se determinan especies de elementos, temperatura electrónica, vibracional y rotacional, y la densidad. La caracterización de superficies sólidas se realiza evaporando la superficie y analizando el espectro resultante.	Procedimental, se mencionan los métodos espectroscópicos para estimar la temperatura como la relación línea a línea, línea a continuo y la relación entre líneas de diferente grado de ionización para el cálculo de densidad electrónica.	Desarrollar software para la caracterización automática de especies de elementos resultantes de una espectroscopia óptica de emisión.

Referencia	**Resumen**	**Metodología empleada**	**Áreas de Oportunidad**
S. S. Hamed, "Spectroscopic Determination of Excitation Temperature and Electron Density in Premixed Laminar Flame," *Egypt. J. Solids*, vol. 28, no. 2, pp. 349–357, 2005.	Uso de la espectroscopia óptica de emisión para captar el espectro con un monocromador. Determina especies, temperatura electrónica, vibracional, rotacional y densidad Posibilidad de caracterizar especies sólidas evaporando la superficie. Determina la temperatura de excitación y a densidad del aire en flamas premezcladas de acetileno y aire. La ecuación de Boltzmann se ocupa para determinar la temperatura electrónica de una línea espectral con el software SigmaPlot.	Procedimental, se mencionan algunos detalles para estimar la temperatura y la densidad de una especie y se da una sugerencia para caracterizar superficies sólidas. Utiliza ecuación de Boltzmann para estimar la temperatura electrónica, considerando parámetros como el peso estadístico, energía, longitud de onda y probabilidad de transición. Es una ecuación sencilla en la que se consideran valores de los picos que se encuentran en los extremos opuestos de un espectro, y la pendiente de la regresión lineal aplicada a estos picos.	Desarrollar software para la caracterización automática de especies de elementos resultantes de una espectroscopia óptica de emisión. Implementar una función para estimar la temperatura electrónica. Implementar la ecuación de Boltzmann para estimar la temperatura electrónica de un espectro de emisión óptica.

Referencia	**Resumen**	**Metodología empleada**	**Áreas de Oportunidad**
A. Sáinz, M. C. García, and M. D. Calzada, "Spectroscopic determination of the electron temperature in non-LTE argon and neon plasmas," *32nd EPS Conf. Plasma Phys. 2005, EPS 2005, Held with 8th Int. Work. Fast Ignition Fusion Targets - Europhys. Conf. Abstr.*, vol. 29C, pp. 1842–1845, 2005.	Uso espectroscopia óptica de emisión como técnica óptica no intrusiva de fácil implementación. La temperatura obtenida fue cercana a los 1300 K.	Generación de plasma con microondas a la frecuencia de 2.45GHz con una potencia que oscila entre 50W y 250W. Las descargas de Neón y Argón se realizan a través de tubos capilares de Quarzo. Los gases empleados tienen una pureza del 99.99% con un flujo de 0.5 L/min	Desarrollar software para la caracterización automática de especies de elementos resultantes de una espectroscopia óptica de emisión. Implementar una función para estimar la temperatura electrónica.
W. Wang, S. Wang, F. Liu, W. Zheng, and D. Wang, "Optical study of OH radical in a wire-plate pulsed corona discharge," *Spectrochim. Acta - Part A Mol. Biomol. Spectrosc.*, vol. 63, no. 2, pp. 477–482, 2006, doi: 10.1016/j.saa.2005.05.033.	El uso de la descarga pulsada corona se debe a que produce bajas temperaturas de gas y alta temperatura de electrones. Se estudian especies de alta reactividad como OH, O, H, N, NO_2, N^+, N_2^+ y O_3 porque remueven gases ácidos de gases de combustión, eliminan compuestos	Empleo de una descarga pulsada corona de alto voltaje con una mezcla de gas N_2 y aire húmedo H_2O. Especial atención al radical OH por la oxidación que produce en muchos procesos fisicoquímicos.	Desarrollar software para la caracterización automática de especies de elementos y moléculas utilizando su espectro, así como un módulo para de almacenamiento de resultados para su posterior análisis.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
	orgánicos del agua, descontaminación bacteriana y descomposición de componentes tóxicos. La población de OH crece linealmente con los picos de voltaje y la tasa de pulsos de repetición en mezclas de gas N_2 y H_2O. La población de OH decrece exponencialmente cuando se agrega O_2 al N_2 y H_2O, pero a cambio se incrementa el flujo de O_2.		
A. Sarani, A. Y. Nikiforov, and C. Leys, "Atmospheric pressure plasma jet in Ar and Ar/H_2O mixtures: Optical emission spectroscopy and temperature measurements,"	Estudio del jet de plasma generado a presión atmosférica con mezclas de Ar y H_2O al 0.05% con tubos capilares de 1.3mm y 3mm a una frecuencia de voltaje de 71 kHz y un voltaje pico de 12.2 kV_{p-p} y 17		Implementar la ecuación de Boltzmann en una función para la estimación de temperatura, que considere los métodos de cálculo del

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
Phys. Plasmas, vol. 17, no. 6, pp. 1–8, 2010, doi: 10.1063/1.3439685.	kV_{p-p} y una fuente de voltaje de 12.8 W La temperatura del gas se incrementa con la adición de H2O al Ar desde 265 K hasta 1,125Km con una intensidad máxima de radicales de OH		área bajo la curva del trapecio y de Simpson.
A. Garduño Aparicio, "Adquisición de espectros ópticos para la estimación de temperatura electrónica," M.S. Thesis, Universidad Autónoma del Estado de México, Centro Universitario UAEM Atlacomulco, 2015.	Estimación de la temperatura electrónica en plasma con la ecuación de Boltzman.	Se descarga una página web correspondiente a la consulta de elementos y sus especies, que posteriormente se guarda en una Base de Datos local. Java es el lenguaje de programación empleado y hace uso de las bibliotecas de funciones JFreeChart y JExcelApi	Uso de lenguaje de programación Python y bibliotecas de cómputo científico como scipy, matplotlib, numpy y pandas optimizadas para el uso de memoria y procesador de un equipo de cómputo, y que al tratarse de software libre se tienen revisiones y optimizaciones periódicas de su código fuente.

Referencia	**Resumen**	**Metodología empleada**	**Áreas de Oportunidad**
E. Yoshida, K. Shizuma, S. Endo, and T. Oka, "Application of neural networks for the analysis of gamma-ray spectra measured with a Ge spectrometer," *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 484, no. 1–3, pp. 557–563, 2002, doi: 10.1016/S0168-9002(01)01962-3.	Empleo de una red neuronal con backpropagation con 47 neuronas de entrada, 28 neuronas de salida y 52 neuronas en la capa oculta, utilizada en el análisis de rayos gamma. Uso de la segunda derivada para la detección de picos. Los resultados oscilan desde el 21% hasta el 99% en precisión de identificación de radioisótopos.	Entrenamiento - 409 data sets Prueba 5 data sets	Predicción en espectros de rayos gamma con otros tipos de redes neuronales. Fijar un umbral para detectar picos.
R. Correa Deves, "Redes Neuronales Artificiales en Ingeniería y Física Nuclear . Caracterización de espectros PIXE," Ph.D. dissertation,	Empleo de emisiones de rayos X inducidas por protones. La transición de electrones produce una radiación de rayos X de un átomo excitado.	Implementación de una red neuronal con propagación hacia adelante y entrenamiento con retro propagación del error.	Dividir datos en conjuntos de entrenamiento y prueba. , en proporciones aproximadas de 80/20 respectivamente.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
Universidad de Granada, 2006.	Dispara un haz de protones de 3.5MeV creados por un generador de Van de Graff que pasa a través de un imán defecto que lo dirige a la muestra, la cual al interactuar con el haz de protones emitirá rayos X y se arrancan electrones de las capas más ligadas a los átomos de los diferentes elementos generando vacancias (espacios vacíos en dichas capas) que serán ocupadas por electrones de capas superiores. Los electrones que atraviesan la muestra se recopilan con una caja de Faraday.	22 conjuntos de datos de muestras orgánicas - 18 entrenamiento - 4 prueba 37 conjunto de datos de espectros de aerosoles - 29 entrenamiento - 8 prueba Se desarrolla una red neuronal por cada elemento químico de interés. -	
O. Miettinen, "Protostellar classification using supervised machine learning algorithms," *Astrophys. Space Sci.*, vol.	Clasificación de objetos estelares en las 3 clases O, I, flat; con los algoritmos de Machine Learning: Naive Bayes, k-Neaerst	Partición de datos: 80% entrenamiento, 20% prueba. 10 fold cross validation validation.	Ensamblar algoritmos de Machine Learning. Los algoritmos con el porcentaje de predicción

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
363, no. 9, pp. 1–17, 2018, doi: 10.1007/s10509-018-3418-7.	Neighbours, Support Vector Machines, Decision Tree, Logistic Regression, Neural Network, Random Forest, Gradient Boosting, Utilizan datos no balanceados con proporciones que oscilan desde el 1% hasta el 37.95%. Las predicciones más altas se alcanzan con Random Forest 81% y Gradient Boosting 82%		más alta corresponden a Random Forest y Gradient Boosting en un rango para ambos del 76% al 82% en precisión.
Y. Bai, J. Liu, S. Wang, and F. Yang, "Machine Learning Applied to Star–Galaxy–QSO Classification and Stellar Effective Temperature Regression," *Astron. J.*, vol. 157, no. 1, p. 9, 2018, doi: 10.3847/1538-3881/aaf009.	Clasificación de 3 clases: estrella, galaxia, QSO. Con 4.4 millones de registros. Se logra alcanzar una precisión del 99.2%. Los espectros similares dificultan la diferenciación espectral.	Implementación de 18 algoritmos, de las cuales se distinguen 3 familias de algoritmos: árboles, k-nearest neighbors y support vector machine. Los algoritmos con mejores algoritmos alcanzando 99.2% de	Explorar algoritmos basados en árboles y SVM.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
		precisión con Random Forest en horas y SVM en una semana. La familia de algoritmos basados en árboles tiene precisión en las predicciones del 97.6% al 98.8 con un costo computacional de minutos.	
J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/bf00116251.	Se menciona que un sistema de Machine Learning tiene 3 dimensiones: estrategia usada, representación del conocimiento y dominio de aplicación del sistema. Se usan los árboles para tareas de clasificación y concluye que una buena elección de características influye en la ganancia de información, velocidad y precisión.	Funcionamiento del algoritmo con ejemplo de salir un sábado por la noche considerando como características: outlook, temperature, humidity, windy.	Lleva a cabo una comparación de las predicciones de algoritmos basados en árboles orientados a problemas de clasificación, basado en los hiperparámetros índice de Gini visto en este artículo respecto a la entropía, utilizando las métricas derivadas de una matriz de confusión.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.	Combina árboles de decisión que se construyen a partir de características y muestras al azar, la predicción se hace por votación y gana la clase con mayor frecuencia. El sobreajuste y el ruido no es un problema y se llega a una buena solución por la ley de los grandes números. Este algoritmo también puede determinar la importancia de cada característica.	Experimento con 19 conjuntos de datos, se deduce que conforme se incrementa el número de características el error también lo hace.	Implementar Random Forest
M. Sebban, R. Nock, J. H. Chauchat, and R. Rakotomalala, "Impact of Learning Set Quality and Size," *Int. J. Comput. Syst. Signal*, vol. 1, no. 1, pp. 85–105, 2000.	Se mencionan la técnica de poda de control en el tamaño del árbol para disminuir el sobreajuste. Se sugiere eliminar valores atípicos, ruido, datos no etiquetados, características irrelevantes.	Experimento con un conjunto de datos de 200,000 muestras.	Considerar la profundidad del árbol en el espacio de búsqueda de optimización de hiperparámetros.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
	Cada algoritmo tiene características que deben se comprendidas y así hacer ajustes correctos.		
D. A. Cieslak and N. V. Chawla, "Learning decision trees for unbalanced data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5211 LNAI, no. PART 1, pp. 241–256, 2008, doi: 10.1007/978-3-540-87479-9_34.	La calidad y tamaño de los datos de entrenamiento influyen en el rendimiento de un árbol de decisión. Los datos no balanceados provocan un rendimiento pobre y se propone la distancia de Hellinger para crear particiones con hojas de mayor pureza.	Experimento con 19 conjuntos de datos y 4 variantes de los árboles de decisión en predicciones de clases binarias. Los conjuntos de datos no balanceados se prueban con 10-fold cross validation.	Implementar distancia de Hellinger.
N. Patel and S. Upadhyay, "Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA," *Int. J. Comput. Appl.*, vol. 60, no. 12, pp. 20–	El sobreajuste es la generación de reglas no deseadas o sin significado aparente. Con las técnicas de poda se reducir el tamaño del árbol y se vuelve más genérico. Se propone la	Experimento con 2 conjuntos de datos y se estudia las predicciones en comparación con la modificación de hiperparámetros.	Optimización de hiperparámetros en predicción de especies de elementos.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
25, 2012, doi: 10.5120/9744-4304.	pre y post poda mediante el uso del software Weka		
W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla, "A Robust Decision Tree Algorithm for Imbalanced Data Sets," *Proc. 10th SIAM Int. Conf. Data Mining, SDM 2010*, pp. 766–777, 2010, doi: 10.1137/1.9781611972801.67.	El uso de la métrica de Hellinger en árboles de decisión con datos no balanceados con el uso del software Weka. AUROC como estimador del algoritmo	Experimento con 5 variantes de árboles de decisión en datos con clase binaria.	Implementar Hellinger en árboles de decisión y algoritmos derivados para la predicción de especies de elementos.
J. Gou, L. Du, Y. Zhang, and T. Xiong, "A New Distance-weighted k-nearest Neighbor Classifier," *J. Inf. Comput. Sci.*, vol. 9, no. 6, pp. 1429–1436, 2012.	Se propone una distancia ponderada dónde los vecinos más cercanos tienen mayor peso. También se indica que, a menor tamaño de la muestra, menor es la precisión. El parámetro de k vecinos es sensible por un valor pequeño o	Experimento con 20-fold cross validation en 12 conjuntos de datos.	Implementar k-nearest neighbor aunque en apariencia las especies de elementos no presentan un patrón de agrupamiento.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
	grande produce predicciones ambiguas.		
S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Trans. Syst. Man Cybern.*, vol. 6, no. 4, pp. 325–327, 1976.	El número de instancias **n** no debe exceder el número de clusters **k**. El error guarda relación con el tamaño del set de entrenamiento.	Experimento con métrica euclidiana en 2 conjuntos de entrenamiento.	Comprobar con más conjuntos de entrenamiento la relación entre el error y el tamaño del conjunto de entrenamiento.
P. H. C. Eilers and H. F. M. Boelens, "Baseline Correction with Asymmetric Least Squares Smoothing," *Leiden Univ. Med. Cent. Rep.*, pp. 1–24, 2005.	Método de suavizado de mínimos cuadrados asimétricos para el cálculo de la línea base con ligera distorsión de los picos. Es un método rápido y ajustable que requiere del juicio humano por inspección visual para determinar los parámetros óptimos.	Variación de *lambda* en ajuste de línea base y *p* para determinar la altura.	Detección automática de convergencia.
V. Van Asch, "Macro-and micro-averaged evaluation measures," *University of*	En 1975 van Rijsbergen introduce la métrica F1 como medida de efectividad. Las variantes principales son **macro** en la que	Ejemplos y gráficas de validación cruzada.	Implementar una matriz de confusión multiclase para el cálculo de la métrica F1.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
Antwerp, vol. 49, pp. 1–27, 2013.	todas las clases tienen el mismo peso, y **micro** donde todas las predicciones tienen el mismo peso.		
T. E. Oliphant, "Python for Scientific Computing," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 10–20, 2007.	Destacan las siguientes características de Python: suficientemente rápido, flexible, open source, multiplataforma, portable, entendible, intérprete interactivo, embebible, gran número de bibliotecas, todo es un objeto, funciones lambda.	Ejemplos que muestran alguna funcionalidad de Python.	Considerar Python como el lenguaje de programación para implementar la solución de caracterización automática de especies.
R. Kumar, "Future For Scientific Computing Using Python," *Int. J. Eng. Technol. Manag. Res.*, vol. 2, no. 1, pp. 30–41, 2015.	Python es ampliamente usado en aplicaciones científicas y proyectos porque es un lenguaje limpio, simple, expresivo, tipado dinámicamente, manejo de memoria, interpretado, modular, orientado a objetos, manejo de excepciones.	Ejemplos que muestran alguna funcionalidad de Python.	Considerar Python como el lenguaje de programación para implementar la solución de caracterización automática de especies.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
	La posición en cómputo científico se debe a su ecosistema de bibliotecas como: numpy, scipy, matplotlib, pandas, sympy.		
D. J. Flannigan, "Spreadsheet-Based Program for Simulating Atomic Emission Spectra," *J. Chem. Educ.*, vol. 91, no. 10, pp. 1736–1738, 2014, doi: 10.1021/ed500479u.	Empleo de una hoja de cálculo para generar espectros sintéticos para los elementos Hidrógeno (^{1}H), Litio (^{3}Li), Neón (^{10}Ne), Sodio (^{11}Na) y Mercurio (^{80}Hg)	Utiliza las fórmulas de partición de temperatura e intensidad para generar espectros sintéticos.	Generación de espectros sintéticos para más elementos con la posibilidad de variar la temperatura, rango de longitud de onda, ancho del pico y elegir entre una longitud de onda observada al vacío o con aire.
B. M. Randles, I. V. Pasquetto, M. S. Golshan, and C. L. Borgman, "Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study," in *2017 ACM/IEEE Joint*	Jupyter Notebook es una herramienta original, libre y robusta para que los científicos puedan compartir código y documentación.	Se realiza una búsqueda en artículos, NBviewer, GitHub y Zenodo con el texto "Jupyter Notebook" para mostrar el número de ocurrencias y concluir que es Jupyter	Explorar las capacidades Jupyter Notebook y determinar en qué áreas del trabajo en curso se puede utilizar.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
Conference on Digital Libraries (JCDL), Jun. 2017, pp. 1–2, doi: 10.1109/JCDL.2017.7991618.		Notebook es una herramienta ampliamente utilizada.	
P. Podrzaj, "A brief demonstration of some Python GUI libraries," 2019, pp. 1–6.	Muestra las capacidades de software libre para desarrollar interfaces gráficas de usuario con PySimpleGUI, Flexx, IPyWidgets.	Muestra una figura de un objeto para la interfaz gráfica junto con el código que lo genera.	Utilizar software libre como IPyWidgets para la generación de interfaces gráficas con Jupyter Notebook. Explorar la posibilidad de crear una interfaz gráfica para un generador de espectros sintéticos.
W. Yu, M. Carrasco Kind, and R. J. Brunner, "Vizic: A Jupyter-based Interactive Visualization Tool for Astronomical Catalogs," *Astron. Comput.*, vol. 20, pp.	Construcción de una interfaz gráfica de usuario para crear un mapa interactivo estelar.	Crea una conexión entre Vizic y los Widgets en Jupyter Notebook.	Explorar las capacidades de los Widgets en Jupyter Notebook para crear una interactividad para un generados de espectros sintéticos.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
128–139, 2017, doi: 10.1016/j.ascom.2017.06.004.			
J. Barnes, *Azure Machine Learning Microsoft Azure Essentials*. Microsoft Press, 2015.	Utiliza el software Microsoft Azure en diseño de experimentos a través de objetos que se pueden arrastrar y soltar para crear flujos de trabajo. Software orientado a servicios web.	Se explica el diseño de experimentos a través de ejemplos con objetos o bloques, junto con la posibilidad de implementar programación en los lenguajes C#, Python y R.	Usar la versión de prueba de Microsoft Azure para explorar los bloques que componen un "experimento" y utilizar algunas de esas ideas en el diseño de experimentos de este trabajo.
O. Irsoy, O. T. Yildiz, and E. Alpaydin, "Design and Analysis of Classifier Learning Experiments in Bioinformatics: Survey and case studies," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 9, no. 6, pp.	Para problemas de clasificación se proponen métricas de rendimiento como confusion matrix y ROC-AUC; los procedimientos de remuestreo k-fold cross validation, leave-one-out y bootstrap; y las pruebas estadísticas de McNemar, 5x2 cv, f test, ANOVA, Wilcoxon's signed rank, Friedman y Nemenyi.	Realiza una comparativa de un conjunto de artículos por: atributos, métricas de rendimiento, tamaño del conjunto de datos, algoritmos de Machine Learning, estrategias de remuestreo y pruebas estadísticas. Finalmente crean un diseño de experimento en el que	Explorar los datos de esta investigación para inferir la viabilidad de implementar las pruebas estadísticas: confusion matrix ROC-AUC, cross-validation, Bootstrap y Nemenyi.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
1663–1675, 2012, doi: 10.1109/TCBB.2012.117.		comparan 5 algoritmos en 11 conjuntos de datos de tumores.	
I. Guyon, J. Makhoul, and R. Schwartz, "Design of experiments for the NIPS 2003 variable selection benchmark Isabelle Guyon – July 2003," *Test*, no. July, 2003.	Se propone el diseño de experimentos para 5 problemas distintos con implementación en Matlab. Un diseño de experimentos corresponde al problema de cáncer de ovario y próstata con datos provenientes de técnicas de espectroscopia.	El diseño de experimentos se compone de las siguientes etapas: • Calcular error, precisión, especificidad y sensibilidad a partir de una matriz de confusión. • Mezcla y unión de orígenes distintos de conjuntos de datos para crear un conjunto más grande. • Filtrado de datos. • Ajuste de espectro por línea base. • Suavizado y alineación del espectro.	Utilizar las ideas del experimento para el cáncer de ovario y próstata por su similitud con este trabajo al utilizar datos provenientes de técnicas de espectroscopia.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
		• División de datos en los conjuntos: entrenamiento, validación y prueba.	
D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Search," *Tech. Rep. SFI-TR-95-02-010*, pp. 1–38, 1996, doi: 10.1145/1389095.1389254.	Un modelo es una versión simplificada de las observaciones, si no se hacen supuestos sobre los datos, entonces no hay razón para pretender un modelo sobre otro. Dado que un algoritmo puede dar buenos resultados en un conjunto de datos y malos resultados en otros datos, la única forma de garantizar un buen resultado es probar un conjunto de algoritmos en los datos, evaluarlos y determinar el mejor.	Comparación de métricas en algoritmos de búsqueda, una métrica es el tiempo y se dice que su duración depende de la implementación de los algoritmos.	Probar un conjunto de algoritmos en los datos del NIST y elegir aquellos que ofrecen los mejores resultados.
A. Halevy, P. Norving, and F. Pereira, "The Unreasonable Effectiveness of Data," *IEEE*	En problemas complejos los datos importan más que los algoritmos.	Procesamiento del lenguaje natural con texto extraído de la Web para la clasificación de	Entrenar algoritmos con datos confiables, en este caso con las líneas de

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
Intell. Syst., vol. 24, no. 2, pp. 8–12, 2009.	No siempre es fácil o económico obtener datos adicionales de entrenamiento.	textos. El rendimiento de un algoritmo depende de la selección cuidadosa de datos.	especies de elementos del NIST. Generar datos sintéticos dado que, ante la contingencia sanitaria presentada en el año 2020, existen restricciones basadas en un semáforo de cuatro colores, que se mantiene en rojo y por tanto, en el Laboratorio de Física de Plasmas del ININ existen restricciones de acceso y retorno en actividades laborales que imposibilitan la generación de datos experimentales en espectroscopia de emisión óptica.

Referencia	**Resumen**	**Metodología empleada**	**Áreas de Oportunidad**
L. Breiman, "Bagging Predictors," *Mach. Learn.*, vol. 24, no. 421, pp. 123–140, 1996, doi: 10.1007/BF00058655.	Se construyen modelos, predictores o estimadores a partir de un subconjunto de datos, cada modelo tiene un subconjunto diferente, esto reduce la varianza en las predicciones.	Pruebas en 7 conjuntos de datos, 4 de ellos son de clasificación binaria y el resto son multiclase. Se utiliza 10 *fold cross-validation* y el estimador base es *decision tree*.	Utilizar *Bagging* para con *Decision Tree* con su respectiva búsqueda de hiperparámetros, si se tienen buenos resultados entonces utilizarlo.
P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.	Detalles del algoritmo *Extremely Randomized Trees*, específicamente: *bootstraping* sin reemplazo, y la selección de características junto con el punto de división por nodo es aleatorio y por tanto más rápido en su construcción que *Random Forest*.	Se presenta el pseudocódigo del algoritmo y una evaluación empírica en 24 conjuntos de datos. Se compara el error respecto al número de estimadores.	Probar *Extremely Randomized Trees* y si su el rendimiento es bueno en comparación a los otros algoritmos entonces utilizarlo.
J. Ingle, J. D. and S. R. Crouch, *Spectrochemical Analysis*. Upper Saddle River, NJ: Prentice Hall, 1988.	Libro que trata temas de análisis espectro químico, tiene un capítulo de componentes ópticos de espectrómetros y otros capítulos de espectrometría de emisión,	Presenta explicaciones detalladas figuras que ilustran conceptos.	Utilizar como fuente de consulta confiable en temas de espectrometría.

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
	absorción, fluorescente, molecular, ultravioleta e infrarroja.		
L. De Galan, R. Smith, and J. D. Winefordner, "The electronic partition functions of atoms and ions between 1500 K and 7000 K," *Spectrochim. Acta Part B At. Spectrosc.*, vol. 23, no. 8, pp. 521–525, 1968.	Incluye los coeficientes para la función de partición electrónica en 73 combinaciones de elemento-nivel_de_energía en los primeros 2 niveles de energía.	Utiliza una función polinómica de quinto orden en los que prueba los coeficientes.	Considerar el uso de los coeficientes y la función polinómica de quinto orden para generar espectros sintéticos, o aún mejor, buscar una función de partición electrónica que no requiera de coeficientes.
S. Vluymans, "Dealing with imbalanced and weakly labelled data in machine learning using fuzzy and rough set methods," Ph.D. dissertation, Universidad de Granada, Belgium, 2019.	Tratamiento de datos no balanceados con técnicas de sobre/sub muestreo y SMOTE. Se propone un método híbrido en sobre/sub muestreo, en problemas de clasificación binaria y multiclase.	Son 18 conjuntos de datos y algoritmos y a cada combinación se les aplica una técnica de muestreo distinta. Se mide la exactitud antes y después de balancear los datos.	Utilizar sobre muestreo a cada clase de elemento con su respectivo nivel de energía para que el construir un árbol a partir de la extracción aleatorio de subconjuntos de datos, todas las clases tengan la

Referencia	Resumen	Metodología empleada	Áreas de Oportunidad
			misma probabilidad de ser seleccionadas.
J. P. Mueller and L. Massaron, *Python for Data Science For Dummies*, Second. 2019.	Análisis y visualización de datos con software como: R, Hadoop, KNIME y D3.JS	Muestra algunas de las características de cada software con figuras y su explicación.	Contemplar la posibilidad de utilizar el flujo de trabajo de KNIME como diagramas en los procedimientos.
J. Robertson and M. Kaptein, Eds., *Modern Statistical Methods for HCI*. Cham: Springer International Publishing, 2016.	Estimar intervalos de confianza con bootstrapping con reemplazo para trabajar con distribuciones de varios tipos.	Comparación de técnicas con gráficas para distintos intervalos de confianza.	Utilizar esta técnica con datos de validación con un intervalo de confianza del 95%,

Se concluye con base en la literatura, la implementación de la ecuación de Boltzmann para la estimación de la temperatura electrónica [11]. El uso de algoritmos basados en árboles [16]–[19] para la caracterización automática de especies [14], [16], [17], utilizar la técnica de *Hellinger* junto con *Gini* y *Entropía* para mejorar los resultados [21], [23]. Medir efectividad con la métrica F1 [5]. Ensamblar algoritmos que tengan las predicciones más altas para determinar la especie de un elemento por votación, toda esta implementación en el lenguaje de programación Python debido a su entorno de programación aplicable en cómputo científico [26], [27]. Por otra parte, tratar datos de espectros proporcionados por el Laboratorio de Física de Plasmas del ININ, con corrección del espectro de fondo continuo [4], y de desplazamiento óptico.

Implementar un repositorio local contenido en un objeto *pickle*, para generar espectros sintéticos del mismo tipo y evaluar el algoritmo con otro tipo de especies, de las cuales no se tienen datos experimentales. Está reportado en la literatura que la validación de modelos con datos sintéticos en el área de Machine Learning es una opción viable. Se mencionan algunos ejemplos dónde se usan datos sintéticos: a) validar la selección de características para clasificar valores aleatorios, b) simular la impedancia y el cambio de fase en un circuito de corriente alterna y c) simular espectros de emisión atómica [30]–[32].

En el tema de la interfaz de usuario su implementación es con QT5 enfocándose en una experiencia de usuario fácil, con la posibilidad de cargar espectros experimentales y de fondo continuo. Además, posteriormente con un clic graficarlos y al mismo tiempo realizar la corrección de desplazamiento óptico y de fondo continuo, así como su caracterización y etiquetado con especie y su probabilidad. Finalmente se agrega la funcionalidad de exportar la gráfica resultante en formato png y pdf.

El diseño de experimentos en el diseño de bloques de Microsoft Azure [33], [34] abarca todas las etapas, desde la adquisición de datos, pasando por la optimización de parámetros, hasta la predicción, dado que el formato de esta tesis tiene una división de capítulos para metodología y experimentación, en la metodología se tratan temas relacionados con: datos, algoritmos, temperatura electrónica y desarrollo de la interfaz gráfica; y en experimentación: diseño de experimentos, métricas de rendimiento, ROC-AUC y Nemenyi y las pruebas sobre datos de validación y prueba [30], [35].

Para la validación del modelo final se crea un generador de espectros sintéticos basado en [32] porque el análisis de estos datos es tan importante como el de los algoritmos [36], y se utilizará Jupyter Notebook [28], [37] para generar la interactividad. Esta integración permite variar los parámetros: tamaños del paso, anchura a media altura del pico y temperatura del espectro; y su efecto en la exactitud de las predicciones del modelo.

Se concluye así que los árboles de decisión [38] y sus algoritmos derivados [19], [39] y afines [31] son una buena alternativa a las redes neuronales para tareas de clasificación. En [16] se tienen precisiones de predicción para tres clases en el rango del 70% al 82% utilizando el espectro luminoso de entes estelares, mientras que en [17] para un problema de tres clases y 4.4 millones de registros de espectros se alcanza precisión en las predicciones en el rango del 72.9% al 99.2%. Esto en comparación a [14] dónde existen tres conjuntos de datos de espectros de rayos gamma para ocho clases con una precisión en las predicciones que oscila del 21% al 99.2%.

La problemática tratada en este trabajo está integrada por 9 clases no balanceadas con 3,899 observaciones, en la literatura se encontraron técnicas de Machine Learning que hacen uso de espectros para clasificar estrellas y radioisótopos de espectroscopia de rayos gamma, y ningún artículo de Machine Learning para espectroscopía de emisión óptica en la predicción de especies, ante la incertidumbre de un umbral mínimo o máximo, y considerando los trabajos afines, se plantea en la hipótesis un umbral de predicción del 70%, con intensión de superar ese umbral y que los resultados obtenidos sirvan de referente para otros trabajos afines.

3 METODOLOGÍA

3.1 ADQUISICIÓN, TRATAMIENTO Y CREACIÓN DE DATOS.

Este capítulo describe el procedimiento para obtener el conjunto de datos de la investigación. Se presentan los procedimientos empleados para: 1) adquirir datos del NIST, 2) preprocesar datos estructurados y 3) crear espectros sintéticos.

3.1.1 Adquisición de datos.

En de esta sección se muestran diagramas que usan la simbología mostrada en la Figura 3.1.1.1.

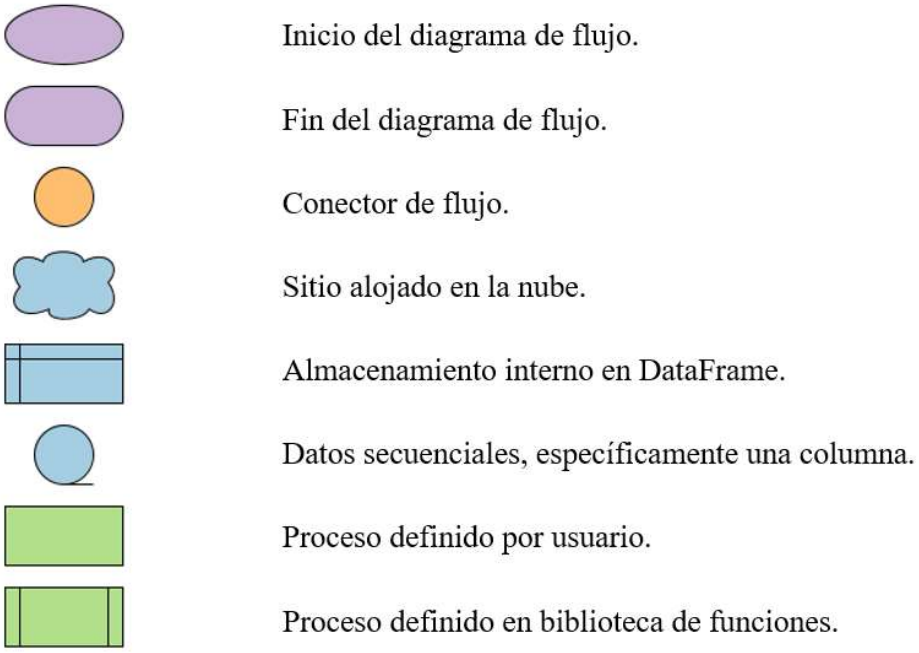


Figura 3.1.1.1. Simbología utilizada de los diagramas de flujo [40].

En el diagrama de flujo de la Figura 3.1.1.2 se observa que la adquisición de datos es el primer procedimiento realizado. Este es un punto clave en el desarrollo de este trabajo porque se trata de los datos de entrenamiento para los algoritmos de estudio.

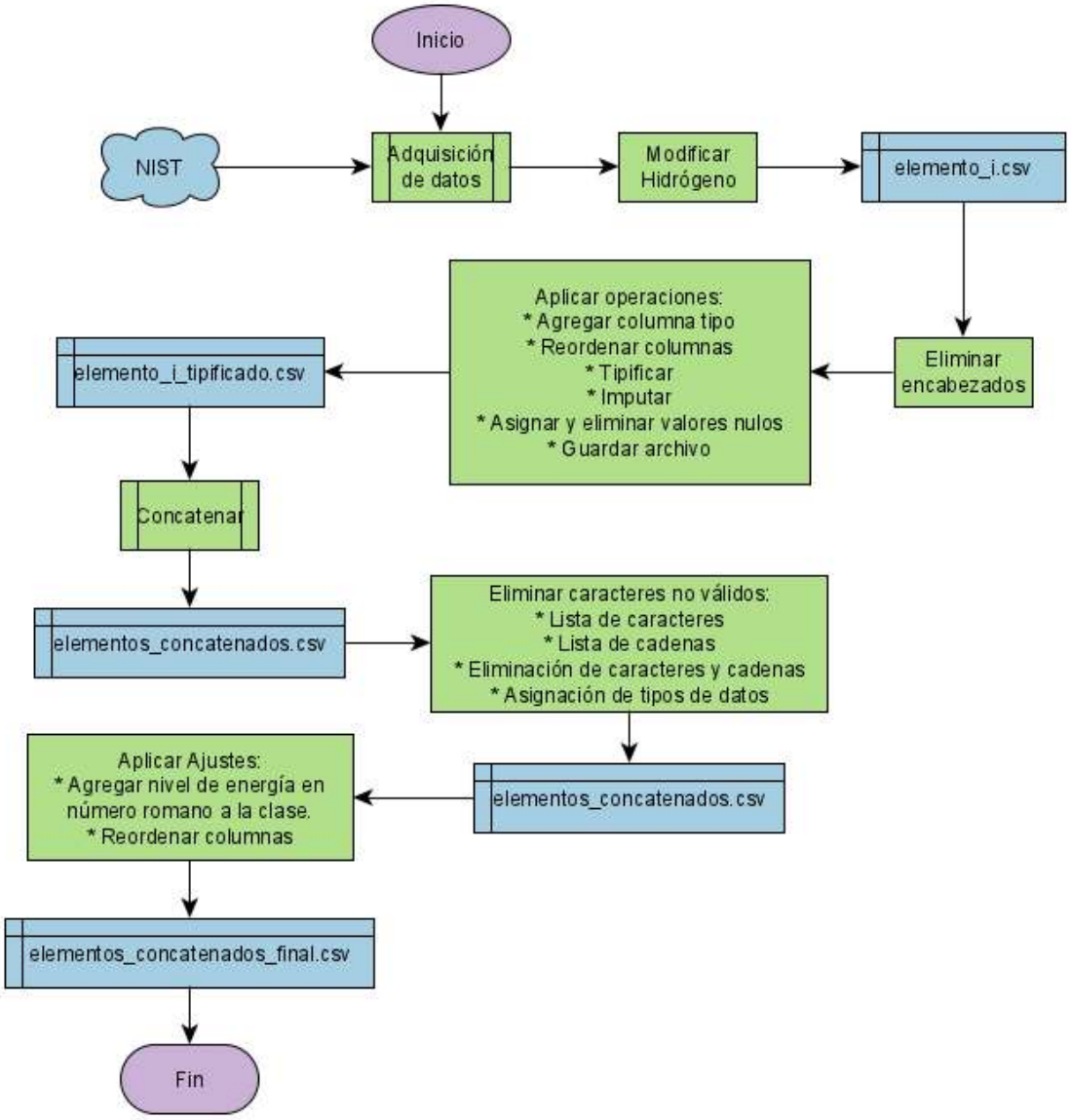


Figura 3.1.1.2. Diagrama de flujo de trabajo para la adquisición y preprocesamiento de datos.

La caracterización de especies requiere de la consulta de líneas reportadas en la base de datos en línea conocida como NIST. Para automatizar este proceso, en primer lugar, se consideró utilizar técnicas de *Web Scraping* con el uso de *Beautiful Soup*, sin embargo, se encontraron errores en los tiempos de respuesta de las consultas. Esto se evitó con la implementación retardos en las consultas, debido a que se producían interrupciones intermitentes en la descarga de los datos. Un método más efectivo fue utilizado para la creación de un repositorio local con los datos del NIST, utilizado la biblioteca de funciones

Wget de Python, y un archivo CSV con un listado de 118 elementos de la tabla periódica utilizado para descargar las líneas de especies.

Las opciones elegidas para obtener la URL utilizada con *Wget* se muestran en la Tabla 3.1.1.1.

Tabla 3.1.1.1. Opciones elegidas del NIST para obtener URL de descarga.

Opción	Valor
Spectrum	Ar
Wavelength Units	Nm
Format output	CSV (text)
Lines	Only with transition probabilities
Wavelength Data	Observed
Level Information	G

Con las opciones de la Tabla 3.1.1.1 es obtenida la URL que se muestra en la Figura 3.1.1.3. En color azul se observa resaltado Ar (gas Argón), debido a que será la cadena de sustitución para descargar las líneas de especies de elementos.

```
https://physics.nist.gov/cgi-bin/ASD/lines1.pl?spectra=Ar&limits_type=0&low
_w=&upp_w=&unit=1&submit=Retrieve+Data&de=0&format=2&line_out=1&en_unit=1&o
utput=0&page_size=15&show_obs_wl=1&order_out=0&max_low_enrg=&show_av=2&max_
upp_enrg=&tsb_value=0&min_str=&A_out=0&intens_out=on&max_str=&allowed_out=1
&forbid_out=1&min_accur=&min_intens=&conf_out=on&term_out=on&enrg_out=on&J_
out=on&g_out=on
```

Figura 3.1.1.3. URL para susitución de descarga.

El objetivo del **Algoritmo 3.1** es descargar las especies de elementos del NIST. Cabe aclarar que todos los algoritmos están escritos en pseudocódigo para su posterior implementación en Python.

Algoritmo 3.1: *Pseudocódigo para la descarga de especies de elementos del NIST.*

Entradas: lista_elementos, nombre_directorio, url,
Salidas: $elemento_i$.csv

```
1  para elementoᵢ en lista_elementos hacer:
2      nombre_archivoᵢ ← concatenar(elementoᵢ, 'csv')
3      urlᵢ ← sustituir(url, elementoᵢ)
4      descargar_guardar(urlᵢ, concatenar(nombre_directorio + nombre_archivoᵢ))
5  fin
```

Donde i, es un subíndice y representa la posición de un elemento en lista_elementos desde 1 hasta 118.

3.1.2 Preprocesamiento de datos estructurados.

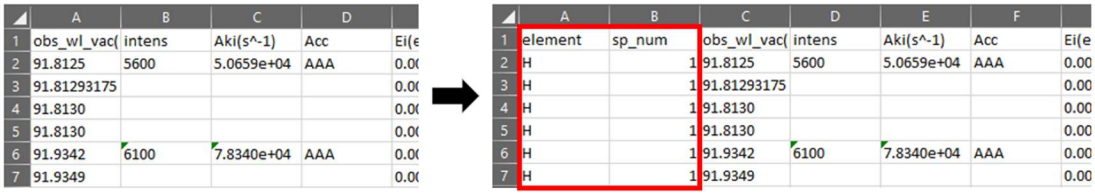
Una vez que se descargaron los datos, se realiza una inspección visual para detectar inconsistencias en estos, las acciones aplicadas se detallan a continuación:

a) Eliminar aquellos archivos con el contenido mostrado en la Figura 3.1.2.1 debido a que no proporcionan información referente a las especies de elementos. Esto sucede porque un elemento consultado no cuenta con todas las características indicadas como criterios.

	A	B	C	D	E	F	G	H	I	J
1	<!DOCTYPE html									
2	PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"									
3	"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">									
4	<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US">									
5	<head>									
6	<title>NIST ASD : Input Error</title>									
7	<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />									
8	</head>									
9	<body bgcolor="white">									
10	<table style="width:100%; border:0; margin-top:0">									
11	<tr><td style="text-align:left"><img									
12	src="https://physics.nist.gov/PhysRefData/ASD/Images/ASD-sm.gif" style="border:0" width="100" height="47"									
13	alt="[ASD Home]" title="ASD Home"><img									
14	src="https://physics.nist.gov/PhysRefData/ASD/Images/Lines.gif" style="border:0" width="70" height="47"									
15	alt="[Lines Query]" title="Lines Query"><img									
16	src="https://physics.nist.gov/PhysRefData/ASD/Images/Levels.gif" style="border:0" width="79" height="47"									
17	alt="[Levels Query]" title="Levels Query"><img									
18	src="https://physics.nist.gov/PhysRefData/ASD/Images/List.gif" style="border:0" width="77" height="47"									
19	alt="[List of Spectra]" title="List of Spectra"><a href="https://physics.nist.gov/PhysRefData/ASD/ionEnergy.html"><img									

Figura 3.1.2.1. Contenido no útil en archivo descargado del NIST.

b) Modificar el archivo del elemento H (ver Figura 3.1.2.2), debido a que es el único que no cuenta con la columna ***element*** (elemento) y **sp_num** (nivel de energía).



	A	B	C	D	
1	obs_wl_vac(	intens	Aki(s^-1)	Acc	Ei(e
2	91.8125	5600	5.0659e+04	AAA	0.0(
3	91.81293175				0.0(
4	91.8130				0.0(
5	91.8130				0.0(
6	91.9342	6100	7.8340e+04	AAA	0.0(
7	91.9349				0.0(

	A	B	C	D	E	F	
1	element	sp_num	obs_wl_vac(	intens	Aki(s^-1)	Acc	Ei(e
2	H	1	91.8125	5600	5.0659e+04	AAA	0.00
3	H	1	91.81293175				0.00
4	H	1	91.8130				0.00
5	H	1	91.8130				0.00
6	H	1	91.9342	6100	7.8340e+04	AAA	0.00
7	H	1	91.9349				0.00

Figura 3.1.2.2. Fragmento del archivo del elemento ^{1}H.

Las siguientes operaciones se aplicaron a cada archivo descargado:

a) Agregar columna ***Type*** con el objetivo de diferenciar entre una longitud de onda observada al vacío y una longitud de onda observada con aire.
b) Reordenar columnas de cada archivo.
c) Tipificar, es decir, asignar valores en la columna ***Type*** en las filas dónde se intercala información para diferenciar las longitudes de onda, dónde ***0*** es el valor asignado a una longitud de onda observado al vacío, y ***1*** es el valor asignado a longitud de onda observada con aire.
d) Imputar valores con reemplazo hacia adelante.
e) Asignar valores ***Nulos*** a las filas con información intercalada.
f) Eliminar filas con valores ***Nulos***.
g) Guardar cada archivo modificado con el sufijo ***_tipificado.csv***.

Una vez que los datos se encuentran modificados, se concatenan. El **Algoritmo 3.2** aplica para cada archivo:

Algoritmo 3.2: *Pseudocódigo para concatenar archivos modificados.*

Entradas: lista_archivos_csv, lista_columnas_interés
Salidas: elementos_concatenados.csv
df ← crear_dataframe()
para archivo$_i$ **en** lista_archivos_csv **hacer**:
    archivo_temp$_i$ ← **cargar**(archivo$_i$)
    archivo_temp$_i$ ← **filtrar**(archivo_temp$_i$, lista_columnas_interés)
    df ← **concatenar**(df, archivo_temp$_i$)
fin
guadar(df, elementos_concatenados.csv)

Para eliminar caracteres no válidos en archivo ***elementos_concatenados.csv***, se realizaron los pasos que se enuncian a continuación:

1. Crear lista de caracteres no deseados: ***['"', '=', '[', ']', '(', ')', '¿', '?', 'u']***
2. Crear lista de cadenas no deseadas: ***['+x', '†', '+y']***
3. Eliminar caracteres no deseados en todo el *DataFrame* con ***broadcasting***
4. Asignar explícitamente el tipo ***int*** a ***['obs_wl_X(nm)','Aki(s^-1)','Ek(eV)']***
5. Guardar cambios como ***elementos_concatenados.csv***.

Comprobar integridad de ***elementos_concatenados.csv*** verificando: columnas que lo integran, tipos de datos y existencia de valores de nulos. El resultado de la verificación es satisfactorio como se observa en la Figura 3.1.2.3.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65597 entries, 0 to 65596
Data columns (total 7 columns):
obs_wl_X(nm)    65597 non-null float64
Aki(s^-1)       65597 non-null float64
Ek(eV)          65597 non-null float64
g_k             65597 non-null int64
Type            65597 non-null int64
sp_num          65597 non-null int64
element         65597 non-null object
dtypes: float64(3), int64(3), object(1)
memory usage: 3.5+ MB
```

Figura 3.1.2.3. Salida de la comprobación del DataFrame generado.

Posteriormente se agregó la columna ***clase*** de la forma ***element*** + ***sp_num*** (en número romano). Finalmente se reordenaron las columnas del *DataFrame* para ser guardado en el archivo ***elementos_concatenados_final.csv***.

3.1.3 Espectros Sintéticos.

Debido a que la base de datos etiquetados experimentales no es suficiente para validar la caracterización automática de espectros [36], se generaron espectros sintéticos basados en el trabajo de Flannigan [32]. El autor describe parte del procedimiento para generar cinco espectros sintéticos de los elementos Hidrógeno (H), Litio (Li), Neón (Ne), Sodio (Na) y Mercurio (Hg) utilizando hoja de datos de Excel. Las principales desventajas son que parámetros como el rango en longitud de onda y cantidad de elementos se encuentra

limitado, por esta razón, se implementaron algunas mejoras. En primer lugar, se amplió el número de elementos de 5 a 84, y en segundo lugar se logró variar parámetros tales como: longitud de onda en cualquier rango válido especificado por el usuario (anteriormente el rango permanecía fijo entre los 390 nm y 700 nm), nivel de energía, el tipo vacío o con aire y la normalización como opcional. Todo esto de manera interactiva con el uso de ***Jupyter Notebook***, ***Python***, ***Matplotlib***, ***Pandas*** y ***Numpy*** [28], [37].

En la Figura 3.1.3.1 se observa el procedimiento diseñado para generar espectros sintéticos.

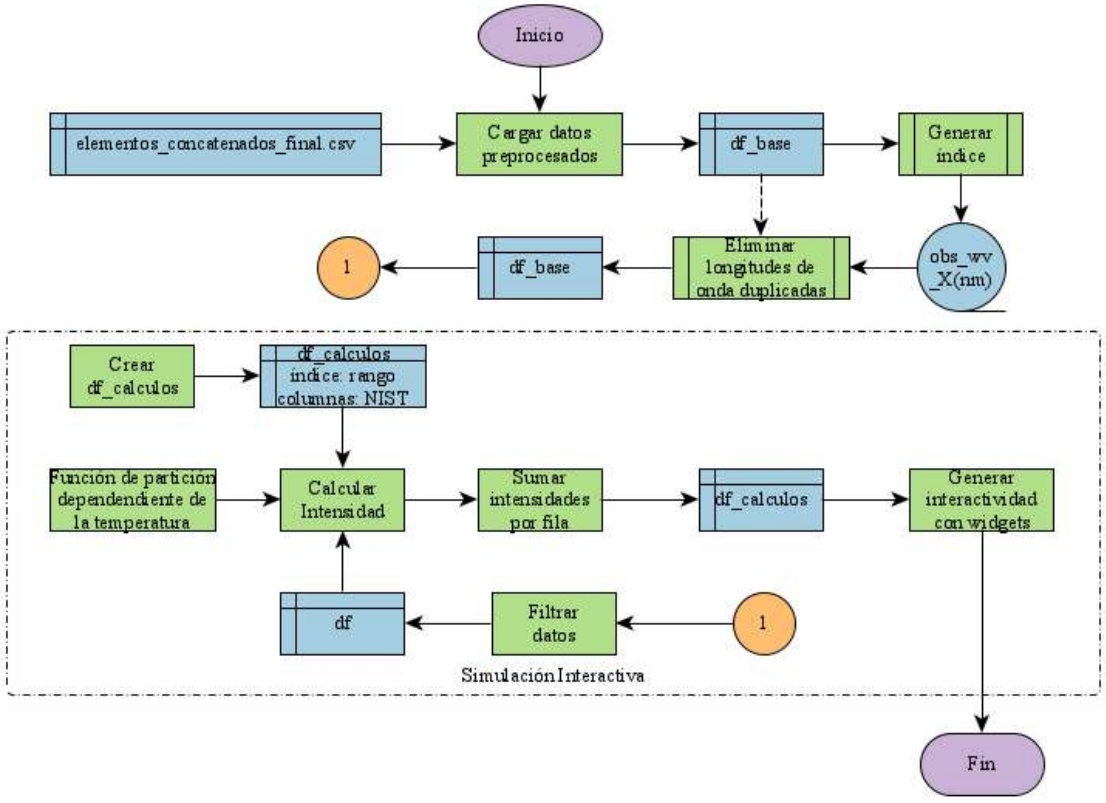


Figura 3.1.3.1. Diagrama de flujo de trabajo implementado para la simulación de espectros sintéticos.

La creación de espectros sintéticos comienza con la carga de datos previamente preprocesados y unificados del archivo ***elementos_concatenados_final.csv*** en un *DataFrame* ***df_base***. Parte de su contenido se observa en la Figura 3.1.3.2.

	element	sp_num	obs_wl_X(nm)	Aki(s^-1)	Ek(eV)	g_k	Type
0	Ag	2	95.62436	150000000.0	17.821972	5	0
1	Ag	2	95.88663	220000000.0	17.982062	5	0
2	Ag	2	97.08875	730000000.0	17.821972	5	0

*Figura 3.1.3.2. Fragmento del DataFrame **df_base**.*

El nombre es estas columnas corresponde tal cual se encuentran en los archivos descargados, con excepción de la columna ***Type*** que se añadió para distinguir la longitud de onda. El significado de las columnas se indica a continuación, y serán utilizadas en ecuaciones posteriores:

element	Elemento de estudio.
sp_num	Grados de ionización.
obs_wl(nm)	Longitud de onda observada.
Aki (s^-1)	Probabilidad de transición.
Ek (eV)	Energía en electrón Volts.
g_k	Peso estadístico.
Type	Permite diferenciar entre una longitud de onda observada al vacío (***0***) u observada con aire (***1***).

Para agilizar el proceso de búsqueda en las longitudes de onda, se creó un índice con ***obs_wl_X(nm)***, el resultado es el que se muestra en Figura 3.1.3.3.

obs_wl_X(nm)	Aki(s^-1)	Ek(eV)	g_k	Type	sp_num	element
95.62436	150000000.0	17.821972	5	0	2	Ag
95.88663	220000000.0	17.982062	5	0	2	Ag
97.08875	730000000.0	17.821972	5	0	2	Ag

*Figura 3.1.3.3. DataFrame **df_base** con índice basado en columna **obs_wl_X(nm)**.*

Basado en la teoría de la ciencia de datos, se buscaron longitudes de onda duplicadas para su eliminación conservado la primera ocurrencia (Figura 3.1.3.4) [41]. Eliminando así 2,890 longitudes de onda duplicadas.

```
print(df_base.shape)
df_base = df_base.loc[~df_base.index.duplicated(keep='first')]
print(df_base.shape)
```

```
(65597, 6)
(62707, 6)
```

*Figura 3.1.3.4. Eliminación de duplicados en DataFrame **df_base** tomando su índice como referencia.*

Implementar funciones de propósito específico que al final integraron el sistema final. La primera función crea un *DataFrame* para los cálculos (**df_calculos**) que toma como argumentos ***limite_inferior***, ***limite_superior*** y ***paso***, todos estos argumentos corresponden a la longitud de onda y su finalidad es crear una estructura como la de la Figura 3.1.3.5.

```
def crear_df_calculos(limite_inferior, limite_superior, paso):
    indice = np.arange(limite_inferior, limite_superior, paso)
    df_calculos = pd.DataFrame(index=indice)          # Establecer índice inicial
    df_calculos.index = df_calculos.index.map(float)  # Indicar tipo flotante
    return df_calculos
```

*Figura 3.1.3.5. Función para crear DataFrame de cálculos **df_calculos**.*

La función (3.1) corresponde a la ***función de partición dependiente de la temperatura*** implementada en Python, mediante el uso de técnicas de broadcasting (aplicación directa y sucesiva de operaciones sobre vectores y matrices) sobre el **df_base**, las primeras pruebas iniciales muestran como resultado un tiempo de ejecución promedio es de 9.85 ms con una desviación estándar de 0.18 ms [42].

$$Q(T) = \sum_{k=0}^{n} g_k e^{-E_k/k_b T} \tag{3.1}$$

Dónde:

g_k	Peso estadístico *(u.a.)*.
E_k	Energía del nivel k en electrón Volts *(eV)*.
k_b	Constante de Boltzmann *(eV/K)*.
T	Temperatura *(K)*.

k Nivel electrónico *(u.a.)*.

n Última muestra del *DataFrame* en el rango de longitud de onda indicado*(u.a.)*.

La función para calcular la ***intensidad*** [43] se muestra en (3.2).

Ecuación

$$I = \frac{2(\frac{g_k A_{ki}}{Q\lambda_c} e^{-E_k/k_b T})}{\pi} \frac{w}{4(\lambda - \lambda_c)^2 + w^2} \tag{3.2}$$

Dónde:

g_k Peso estadístico *(u.a.)*.

A_{ki} Probabilidad de transición *(s^{-1})*.

E_k Energía electrónica *(eV)*.

k_b Constante de Boltzmann *(eV/K)*.

λ Paso de longitud de onda entre cada par de puntos adyacentes *(nm)*.

λ_c Longitud de onda del NIST dentro del rango indicado *(nm)*.

T Temperatura *(K)*.

w Anchura a media altura *(nm)*.

Q Función de partición dependiente de la temperatura.

I Intensidad (u.a.)

Cada vez que se genera un *DataFrame* de intensidades es necesario sumar las filas y colocar el resultado en una columna que corresponde al espectro sintético a graficar (***espectro_final***).

Dado que solo se trabaja sobre un rango específico de datos, se deben filtrar los datos cada vez que el usuario modifique un control para generar espectros sintéticos, esta operación se realiza sobre ***df_base*** y se aplica cuando se modifica: elementos (***element***), grados de ionización (***sp_num***), tipo (***0: vacío, 1: aire***) y rango de longitud de onda (***index: obs_wl_X(nm)***).

El proceso para generar la interactividad requiere de la biblioteca de funciones ***ipywidgets*** propia de ***Jupyter Notebook*** [37] para definir los controles, se muestran en Tabla 3.1.3.1, estos valores son de inicio, y se pueden cambiar desde la interfaz gráfica generada.

Tabla 3.1.3.1. Opciones utilizadas en los controles de Jupyter Notebook.

Control	Valor Predeterminado	Mínimo	Máximo	Paso	Descripción
FloatSlider	0.2	0	2	0.1	'Anchura:'
IntSlider	5000	0	50000	500	'Temperatura K:'
Checkbox	False	-	-	-	'Normalizado:'
Dropdown	'Ne'	-	-	-	'Elemento:'
RadioButtos	'Aire'	-	-	-	'Tipo'
SelectMultiple	[1]	-	-	-	'Nivel Energía:'
IntRangeSlider	[200, 890]	0	2000		'Rango:'

Cada uno de estos controles de describe en Tabla 3.1.3.2.

Tabla 3.1.3.2. Controles usados en la interactividad de espectros sintéticos.

Control	Características
FloatSlider	Control deslizante para valores de tipo flotante.
IntSlider	Control deslizante para valores de tipo entero.
Checkbox	Casilla de verificación para valores booleanos.
Dropdown	Lista desplegable para valores de tipo cadena.
RadioButtos	Botón de opción para valores booleanos y enteros.
SelectMultiple	Selección múltiple para valores de tipo cadena, enteros y flotantes.
IntRangeSlider	Control deslizante por rango para valores enteros.

Estos controles se distribuyeron en dos pestañas (Filtros y Gráfica), el resultado se puede observar en la Figura 3.1.3.6.

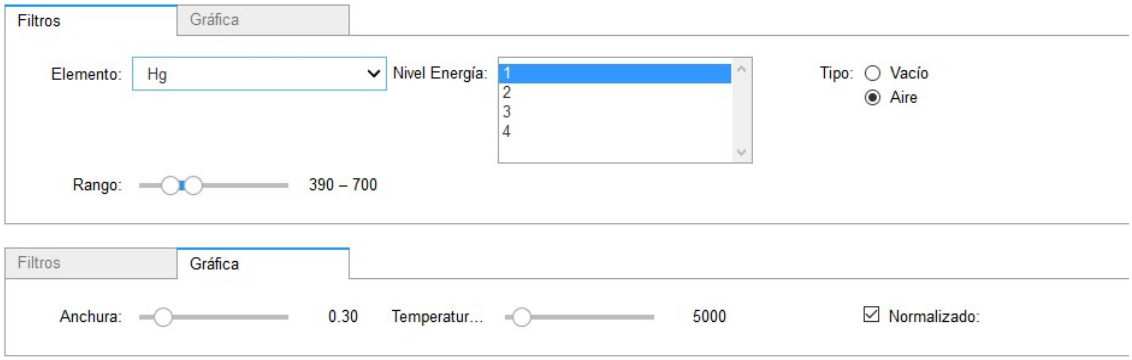
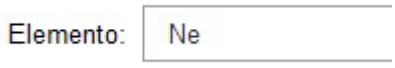


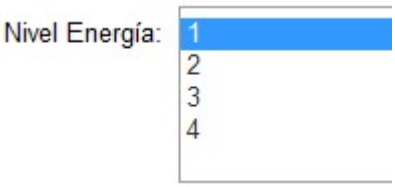
Figura 3.1.3.6. Distribución de objetos en pestañas para manipular el espectro sintético.

En la Figura 3.1.3.3 se muestra el código que genera cada objeto visto en la Figura 3.1.3.6.

Tabla 3.1.3.3. Definición de objetos que integran el simulador de espectros sintéticos.

Objeto	Código

	`_LISTA_ELEMENTOS = widgets.Dropdown(`
	`  options=LISTA_ELEMENTOS,`
	`  value='Ne',`
	`  description='Elemento:',)`

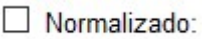
	`_NIVEL_ENERGIA = widgets.SelectMultiple(`
	`  options=NIVEL_ENERGIA,`
	`  value=[1],`
	`  description='Nivel Energía:',)`

	`_TIPO = widgets.RadioButtons(`
	`  options=TIPO,`
	`  value=TIPO['Aire'],`
	`  description='Tipo:',)`

	`_RANGO = widgets.IntRangeSlider(`
	`  value=[200, 890],`
	`  min=0,#177,`
	`  max=2000,#891,`
	`  step=1,`
	`  description='Rango:',`
	`  disabled=False,`
	`  continuous_update=False,`
	`  orientation='horizontal',`
	`  readout=True,`
	`  readout_format='d',`
	`)`

	`_ANCHO_PICO = widgets.FloatSlider(`
	`  value=0.2,`
	`  min=0,`
	`  max=2,`
	`  step=0.1,`
	`  description='Anchura:',)`

	`_TEMPERATURA = widgets.IntSlider(`
	`  value=5000,`
	`  min=500,`
	`  max=50000,`
	`  step=500,`
	`  description='Temperatura K:',)`

	`_NORMALIZADO = widgets.Checkbox(`
	`  value=False,`
	`  description='Normalizado:',)`

	`tab1 = widgets.VBox(children=[` `  widgets.HBox(children=[_LISTA_ELEMENTOS,`

Posteriormente se creó una función que integra todo lo desarrollado hasta ahora, como se muestra en la sección punteada (***interactividad***) de la Figura 3.1.3.1 mediante una función ***filtro***, que a su vez se subdivide en tres bloques: 1) filtrado, 2) cálculo y, 3) graficación. Finalmente, la ***interactividad*** se genera con el siguiente código de la Figura 3.1.3.7.

```
# Interactividad
out = widgets.interactive_output(
    filtro, {
        'lista_elementos':_LISTA_ELEMENTOS,
        'nivel_energia':_NIVEL_ENERGIA,
        'tipo':_TIPO,
        'rango':_RANGO,
        'ancho_pico':_ANCHO_PICO,
        'temperatura':_TEMPERATURA,
        'normalizado':_NORMALIZADO}
)

display(ui, out)
```

*Figura 3.1.3.7. Código que integra la función **filtro** y los objetos que manipularán cada parámetro de entrada.*

3.1.4 Simulación de Espectros Sintéticos con Jupyter Notebook.

El resultado al integrar todos los procedimientos descritos hasta el momento es la interfaz mostrada en la Figura 3.1.4.1 dónde se observa la ventana donde se grafican los datos sintéticos junto con los controles que permiten desplazarse, aplicar acercamiento y guardar la gráfica. La escala en el eje y esta normalizada, esto consiste en dividir entre el valor máximo todas las intensidades del espectro para mantener los valores en un rango entre 0 y 1.

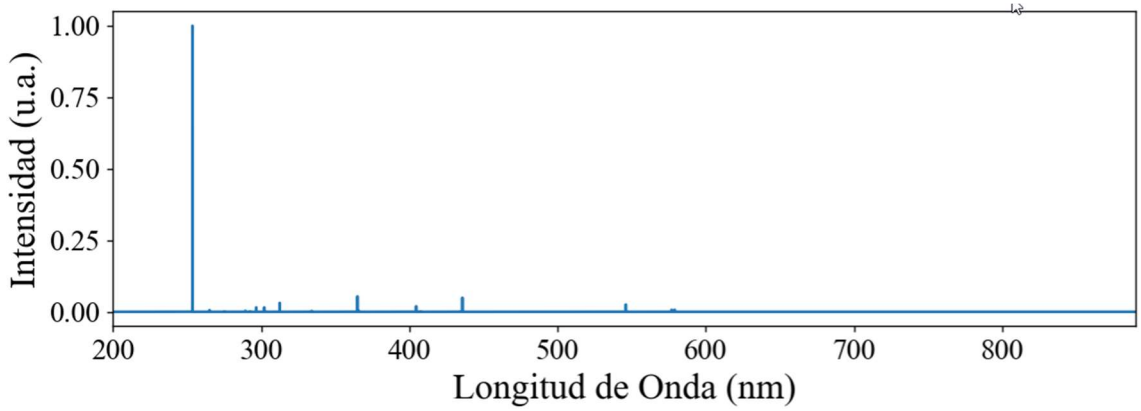


Figura 3.1.4.1. Espectro sintético normalizado de Hg I con un Paso de 0.01 nm, Anchura de 0.01 nm y una Temperatura de 8,000 K.

En la Figura 3.1.4.2 se observan los controles de mando programados para interactuar con el usuario, como son: elemento, nivel de energía, tipo, rango, anchura, temperatura y normalizado. El botón **Guardar Espectro** se usa para exportar en formato CSV los datos definen el espectro sintético, y su utilidad se aprecia en las secciones 5.1 y 5.2.

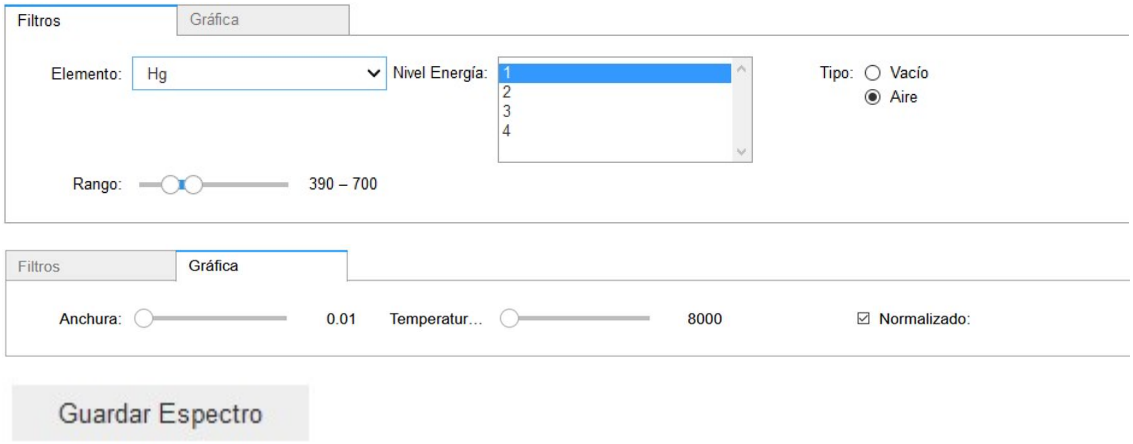


Figura 3.1.4.2. Simulación de espectros sintéticos con Jupyter Notebook.

La combinación de valores en cada control permite generar distintos espectros sintéticos, por ejemplo, en la Figura 3.1.4.3 se observa la especie de Ar I a una temperatura de 8,000 K.

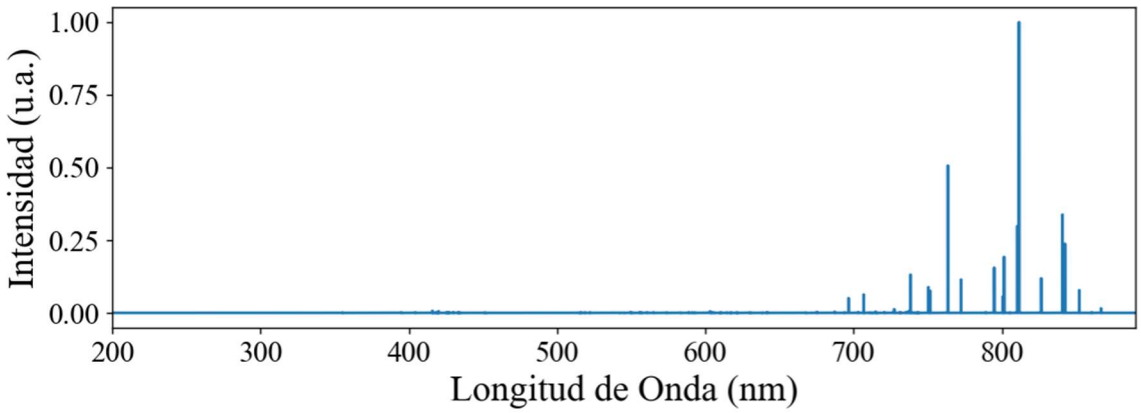


Figura 3.1.4.3. Espectro sintético normalizado de Ar I con un Paso de 0.01 nm, Anchura de 0.01 nm y una Temperatura de 8,000 K.

Por otra parte, en la Figura 3.1.4.4 se muestra Ar I con una temperatura de 80,000 K. Se observa que a medida que incrementa la temperatura, aumenta la intensidad de especies de Ar I que no se apreciaban en la Figura 3.1.4.3.

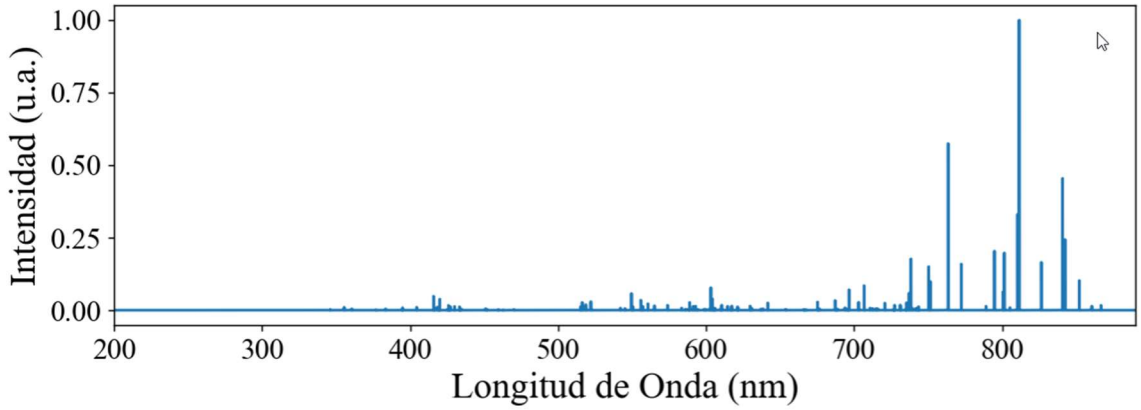


Figura 3.1.4.4. Espectro sintético normalizado de Ar I con un Paso de 0.01 nm, Anchura de 0.01 nm y una Temperatura de 80,000 K.

Hay una etapa de la implementación que se sustituyó por operaciones de ***broadcasting*** que consiste en realizar operaciones sobre un arreglo de valores como si se tratara de vectores. Anteriormente, las operaciones se realizaba elemento a elemento dentro de bucles ***for*** y ***while***, esta implementación consumió inicialmente tiempos de hasta 28 minutos por espectro generado, posteriormente se implementaron mejoras en la definición

de tipos de dato con tiempos de hasta 22 minutos. Finalmente, con técnicas de ***broadcasting*** se alcanzaron tiempos máximos de 350 ms por cada espectro generado. Es importante destacar que hay espectros de elementos que tienen más especies que otras, y en el caso del elemento Fe hay un tiempo máximo de 2.96 s.

Estos datos sintéticos son los utilizados en la sección 5 para validar la caracterización de especies en diferentes condiciones de anchura, paso y temperatura.

3.2 CARACTERIZACIÓN AUTOMÁTICA DE ESPECIES.

La caracterización automática de especies de elementos en espectros proveniente de plasma frío requiere de una secuencia de etapas que se detallan a continuación. Como se ha analizado en el estado del arte, y asimismo concluido en la misma sección, el objetivo fue describir una metodología, no sólo para la sección de diseño e implementación de Machine Learning, sino también de pruebas de este.

3.2.1 Filtrado de datos.

Esta etapa se enfoca en cargar en un *DataFrame* el archivo ***elementos_concatenados_final.csv*** para realizar un filtrado con base a 4 parámetros ajustados a los requerimientos del Laboratorio de Física de Plasmas del ININ. Las opciones con sus filtros son:

1. Elementos: **['Hg', 'Ar', 'N', 'O', 'He']**
2. Grados de Ionización: **[1, 2]**
3. Tipo: **[0, 1]**
4. Rango de longitud de onda: **[200, 890]**

Partiendo de estos criterios es necesario hacer el tratamiento de los datos según su clase, como se describe a continuación.

3.2.2 Balanceo de Clases.

Cuando se adquieren datos, estos pueden tener clases en diferentes proporciones, es decir, en diferentes cantidades por clase, lo que se conoce como desbalance de clases, afectando las predicciones. Al momento de verificar la distribución de los datos respecto a la clase, se observa (Figura 3.2.2.1) que los datos no están balanceados, en suma con las 9 clases se tiene un total de 1,284 especies, y como clase mayoritaria se tiene O II con 326 especies, mientras que como clase minoritaria se encuentra el elemento Hg I con 35 especies. Cabe mencionar que, en un inicio se tenían 3,899 especies, y como clase mayoritaria Ar II con 1,768 especies, sin embargo, una observación del usuario fue que las únicas especies a considerar deben ser las emisoras con los datos de A_{ki}, por lo tanto, se redujeron los datos, con base a esta característica.

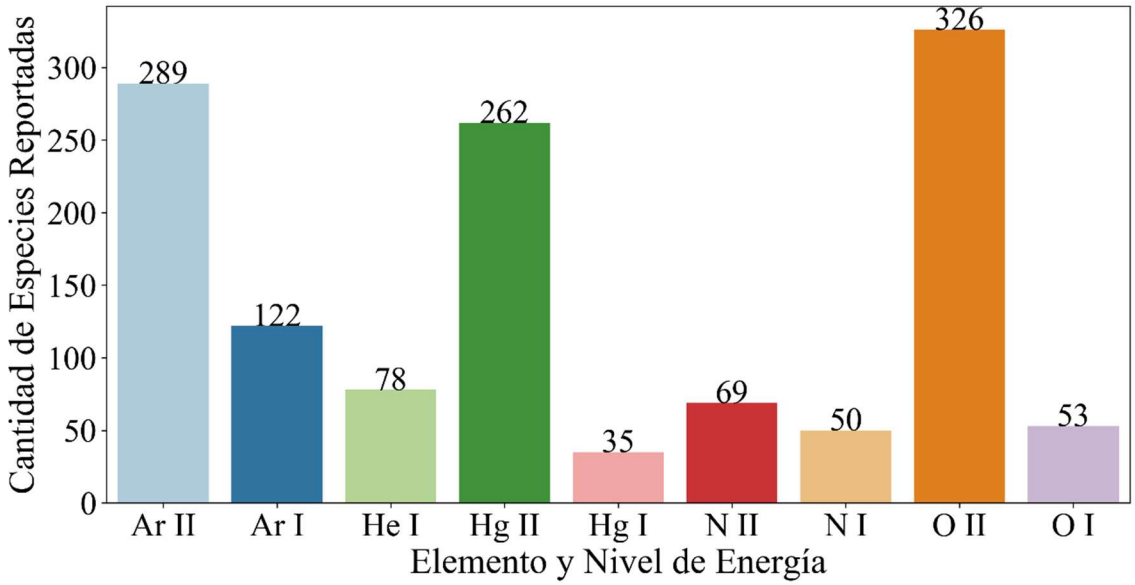


Figura 3.2.2.1. Cantidad de especies por elemento y nivel de energía.

Este desbalance de clases se puede tratar con técnicas de sub-muestreo y sobre-muestreo [44]. En el sub-muestreo se elimina aleatoriamente muestras de las clases mayoritarias hasta que todas las clases tengan la misma cantidad de muestras de la clase minoritaria, esto afecta en la predicción porque no se pueden predecir especies que fueron eliminadas. Por otra parte, el sobre-muestreo selecciona aleatoriamente muestras de cada clase minoritaria hasta alcanzar la misma cantidad de especies de la clase mayoritaria, esto permite la inclusión de especies con clases minoritarias en cada árbol generado. Dado que O II tiene 326 especies, al aplicar sobre-muestreo, cada clase alcanza este valor, el efecto es mostrado en la Figura 3.2.2.2, dando una suma total de 2,934 especies por las 9 clases.

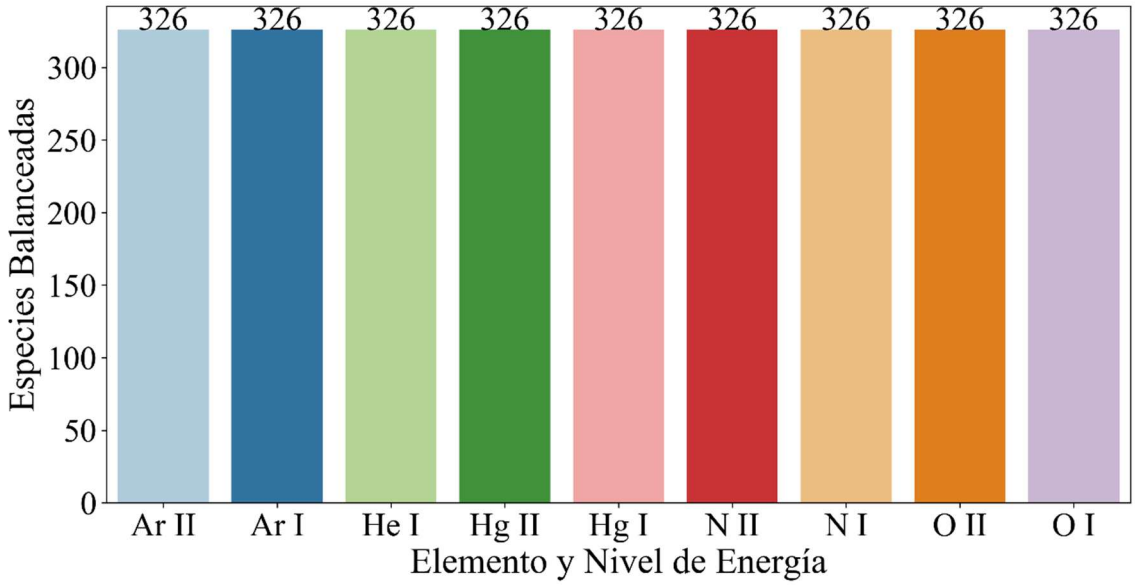


Figura 3.2.2.2. Especies por elemento y nivel de energía después de aplicar sobre-muestreo.

3.2.3 Codificación de Clase.

Es más sencillo realizar operaciones con números que en cadenas, las características de los datos reportados en el NIST contienen valores categóricos, este inconveniente se soluciona asignando un valor numérico a cada clase. La tabla de equivalencias para la clase de interés del Laboratorio de Física de Plasmas del ININ se muestra en la Tabla 3.2.3.1.

Tabla 3.2.3.1. Tabla de equivalencias entre valores categóricos y numéricos.

Valor Categórico	Valor Numérico
Ar I	0
Ar II	1
He I	2
Hg I	3
Hg II	4
N I	5
N II	6
O I	7
O II	8

Esta tabla de equivalencias también se usa para realizar el proceso inverso en las predicciones, al convertir un número en cadena y mostrarlo en la caracterización.

3.2.4 Detección de Picos.

La detección de picos se realiza eligiendo un segmento de datos que supera un umbral de intensidad y para determinar la posición del valor máximo, este proceso se repite sucesivamente y así se eligen los picos, este procedimiento se detalla mediante el pseudocódigo del Algoritmo 3.1.

Algoritmo 3.1: *Pseudocódigo para la detección de picos.*

```
    Entradas: intensidades, umbral, tamaño_ventana
    Salidas: índice_picos
 1  intensidades = intensidades_i > umbral
 2  inicio ← 0
 3  fin ← tamaño_ventana
 4  i ← 0
 5  para tamaño_ventana en intensidades hacer:
 6      tamaño_ventana ← tamaño_ventana[inicio, fin]
 7      índice_picos[i] ← arg_max(tamaño_ventana)
 8      inicio ← fin + 1
 9      fin ← tamaño_ventana + 1
10      i ← i + 1
11  fin
12  retornar índice_picos
```

Se puede observar en la Figura 3.1.4.1 una línea punteada de color naranja que representa el umbral determinado por el usuario, a partir del cual se detectan los picos. Además, los puntos en color rojo con sombra azul son los picos detectados con el Algoritmo 3.1.

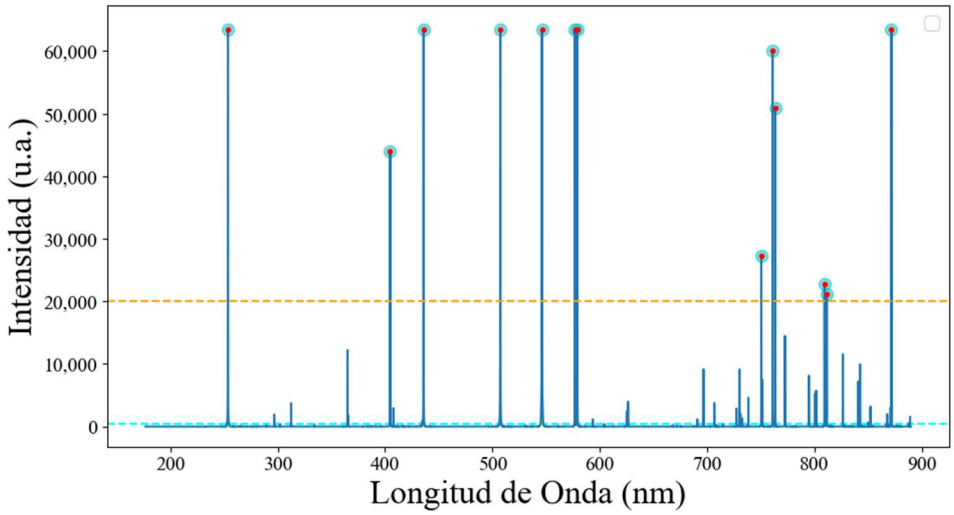


Figura 3.2.4.1. Detección de picos con el algoritmo propuesto aplicado a espectro experimental de la lámpara calibración HG-1.

3.2.5 Selección de Algoritmos de Machine Learning.

La selección de algoritmos de Machine Learning se realizó con los algoritmos extraídos en [16] si bien en este trabajo no se predicen especies, se usa el espectro proveniente de estrellas para su clasificación. Estos algoritmos se usan para predecir las clases de los elementos filtrados y solo se seleccionan aquellos con la puntuación más alta, estos resultados se observan en la Tabla 3.2.5.1.

Tabla 3.2.5.1. Rendimiento para clasificadores de Machine Learning [16].

Clasificador	Precisión
Naive Bayes	0.47
k-Nearest Neighbours	0.54
Support Vector Machine	0.68
Decision Tree	0.71
Logistic Regresion	0.79
Neural Network	0.80
Random Forest	0.81
Gradient Boosting Machine	0.82

En este caso, al usar estos algoritmos con los datos del NIST sin preprocesamiento, la precisión más alta se obtuvo con Decision Tree y Random Forest. En la búsqueda de algoritmos alternativos, se encontró el reporte técnico de Wolpert, este menciona no existe algo como "el mejor algoritmo de aprendizaje", un algoritmo puede ser bueno en un conjunto de datos y malo en otro [45], por tal motivo se realizaron pruebas con otros algoritmos y se encontró con Extremely Randomized Trees y Bagging, también se obtienen resultados favorables.

En las siguientes secciones se detallan los algoritmos empleados y al final se explica el procedimiento para obtener modelos con poco sesgo y varianza.

3.2.6 Decision Tree.

Un árbol de decisión está compuesto por nodos y hojas, cada nodo *padre* corresponde a una comparación o decisión, mientras que los nodos terminales (hojas) corresponden a una clase. Para su construcción y aprendizaje se requiere de un conjunto de datos finitos como se define en (3.3):

$$\mathcal{L} = \{X, y\} \tag{3.3}$$

Dónde:

- $\mathcal{L}$ Conjunto de aprendizaje.
- X Es el conjunto de características de entrada integrado por muestras en una matriz bidimensional que tiene la forma $(n\_muestras, n\_características)$.
- y Son los valores de salida en un vector con la forma $(n\_muestras)$.

El objetivo es construir un estimador $\varphi_{\mathcal{L}}$ como se muestra en (3.4):

$$\varphi_{\mathcal{L}}: X \to y \text{ minimiza } Err(\varphi_{\mathcal{L}}) = \mathbb{E}_{X,y}\{\mathcal{L}(y, \varphi_{\mathcal{L}}.predice(X)\} \tag{3.4}$$

El algoritmo del árbol de decisión [38], es el siguiente:

Algoritmo 3.2: *Pseudocódigo para árboles de clasificación y regresión.*

Entradas: $\mathcal{L}$

Salidas: nodo ***t***

Función construir_árbol_decisión($\mathcal{L}$):
  Crear nodo ***t***
  Si es alcanzado el criterio de detención para ***t*** **entonces**:
    $\hat{y} \leftarrow$ modelo
  Si no:
    Encontrar división en $\mathcal{L}$ que maximiza el decrecimiento de impureza
    $s^* = \arg\max i(t) - p_L i(t_L^s) - p_R i(t_R^s)$
    Partición $\mathcal{L}$ dentro $\mathcal{L}_{t_L} \cup \mathcal{L}_{t_R}$según s^*
    $t_L = construir\_árbol\_decisión(\mathcal{L}_{t_L})$
    $t_R = construir\_árbol\_decisión(\mathcal{L}_{t_R})$
  Fin Si
  Retornar ***t***
Fin Función

Dónde:

- s^* Es la mejor de las mejores divisiones definidas en cada variable de entrada.
- $\hat{y}$ Son los valores de salida predichos en un vector con la forma $(n\_muestras)$.
- t_L Nodo de decisión izquierdo.
- t_R Nodo de decisión derecho.
- $i(t)$ Evalúa qué tan bueno es el nodo t.

Las características de operación de un árbol de decisión son:

- Es un modelo no paramétrico, esto lo hace consistente en los resultados.
- Soporta datos heterogéneos (continuos, discretos, ordenados y categóricos).
- Rápido de entrenar y predecir. Su complejidad promedio es $\Theta = (pN\ log^2 N)$.
- Fácil de interpretar al graficar cuando al árbol es reducido, de lo contrario, resulta complejo.
- Bajo sesgo, y usualmente alta varianza (si los datos de entrenamiento varían ligeramente, el árbol resultante y las predicciones pueden cambiar significativamente). Esto se soluciona combinando varios árboles en un solo modelo.

3.2.7 Bagging

Una mejora que se puede realizar a un árbol de decisión es crear un conjunto de este predictor, dónde cada instancia se entrena con un subconjunto aleatorio de muestras de entrenamiento (Figura 3.2.7.1). Cuando el muestreo se realiza con reemplazo (la misma muestra puede tener más de una ocurrencia), a este método se le denomina *bagging* (contracción de *bootstrap aggregating*), y cuando el muestreo se realiza sin reemplazo es denominado *pasting* [13].

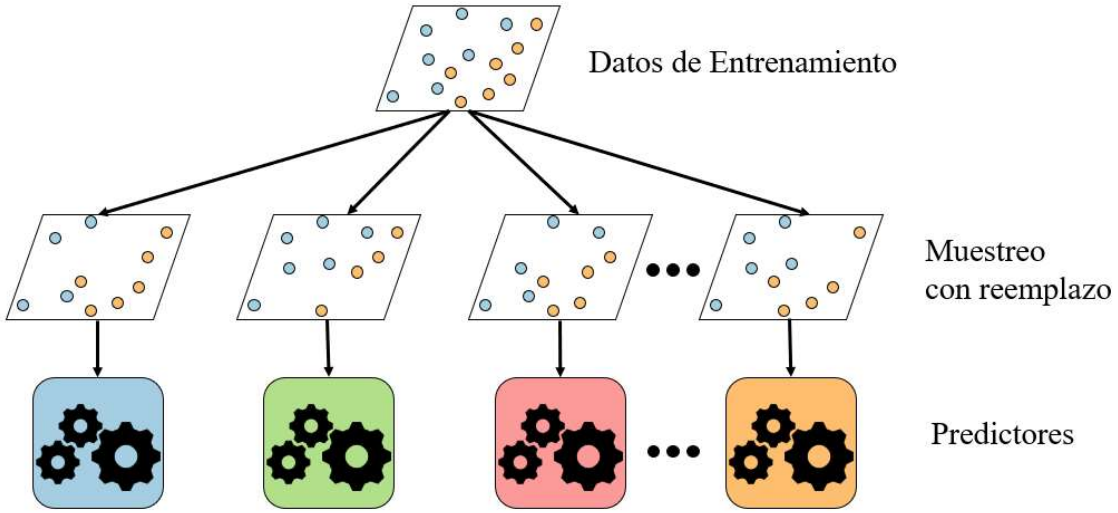


Figura 3.2.7.1. Conjuntos de muestreo con bagging.

Una vez que todos los predictores son entrenados, la predicción para una muestra nueva se puede realizar en el caso de una clasificación por votación como se describe en la sección 3.2.10, o calculando el promedio si se trata de una regresión. Una ventaja del *bagging* es que reduce la varianza de algoritmos con alta varianza como el *decision tree*, sin embargo, los modelos construidos con esta técnica pueden tener similaridad estructural que deriva en predicciones correlacionadas, esta desventaja se corrige con *random forest*, el cual se detalla en la siguiente sección.

3.2.8 Random Forest.

En este algoritmo, se construyen varios árboles de decisión agregando aleatoriedad al proceso de construcción para dar origen a un bosque. El proceso para elegir el umbral de división es el óptimo calculado, y las muestras para cada estimador se extraen con

bagging, y tiene el efecto de reducir la correlación en las predicciones al tomar un subconjunto aleatorio de características [19], como se observa en la Figura 3.2.8.1.

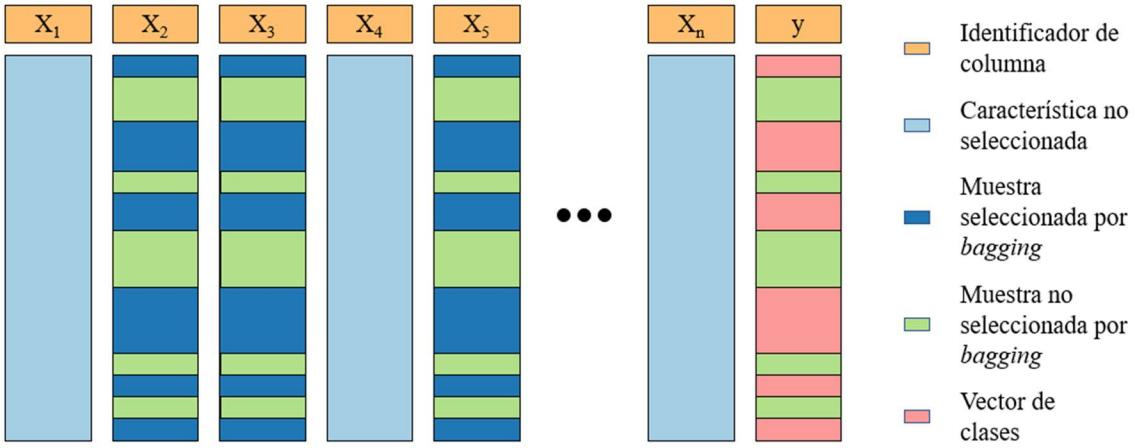


Figura 3.2.8.1. Selección de muestras por bagging para construir un árbol.

El número de características se calcula en base a las ecuaciones (3.6) y (3.7). Por ejemplo, para $p = 25$ se obtiene $m = 5$ con (3.6) y $m = 4$ con (3.7).

$$p = \sum_{i=1}^{n} X_i \tag{3.5}$$

$$m = \lfloor \sqrt{p} \rfloor \tag{3.6}$$

$$m = \lfloor \log_2 p \rfloor \tag{3.7}$$

Dónde:

- X Vector de características
- n Última característica
- p Número total de características.
- m Subconjunto de características a elegir aleatoriamente.

Entre las ventajas de este algoritmo se encuentran que se reduce el sobre-ajuste, la varianza y la correlación en las predicciones.

3.2.9 Extremely Randomize Trees.

Este algoritmo es similar al Random Forest, con la diferencia de que el umbral de división para cada nodo de decisión es aleatorio y la extracción de observaciones es aleatoria y sin reemplazo, es decir, *pasting* [13], [39] como se aprecia en la Figura 3.2.9.1.

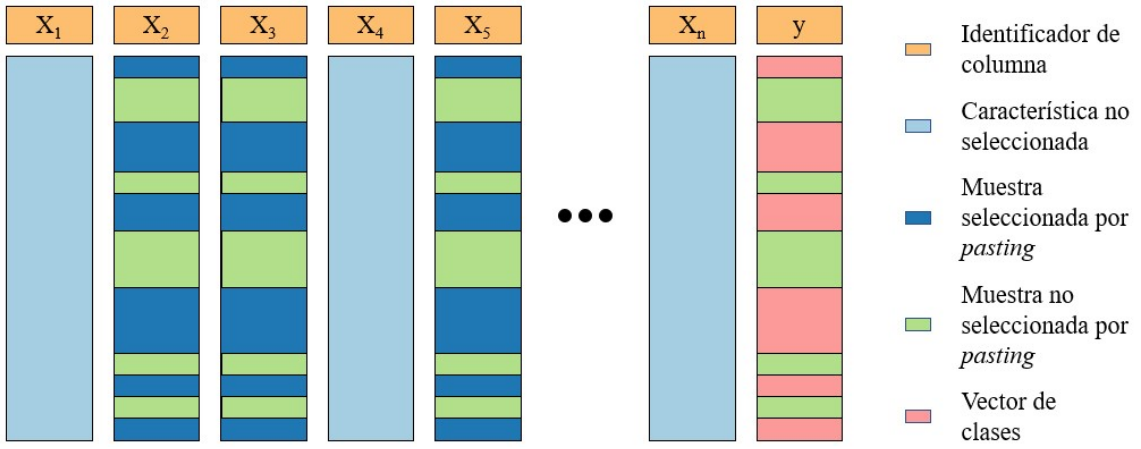


Figura 3.2.9.1. Selección de muestras por pasting para construir un árbol.

Extremely Randomized Trees tiene las mismas ventajas que Random Forest, con la adición de que su proceso de construcción es más rápido porque el umbral de división en lugar de ser calculado es aleatorio, esto se aprecia en la Tabla 4.4.2, dónde en la búsqueda de hiperparámetros Random Forest tiene un tiempo máximo en días de 25.418 y Extremely Randomized Trees con 25.967 días.

3.2.10 Clasificación por Votación.

Una forma de incrementar la precisión y la efectividad ***F1 macro*** es con técnicas de votación. La primera de estas técnicas es la denominada ***Votación Fuerte*** (Figura 3.2.10.1), esta consiste en tomar la predicción de todos los algoritmos y la predicción corresponde a su moda [44, p. 664].

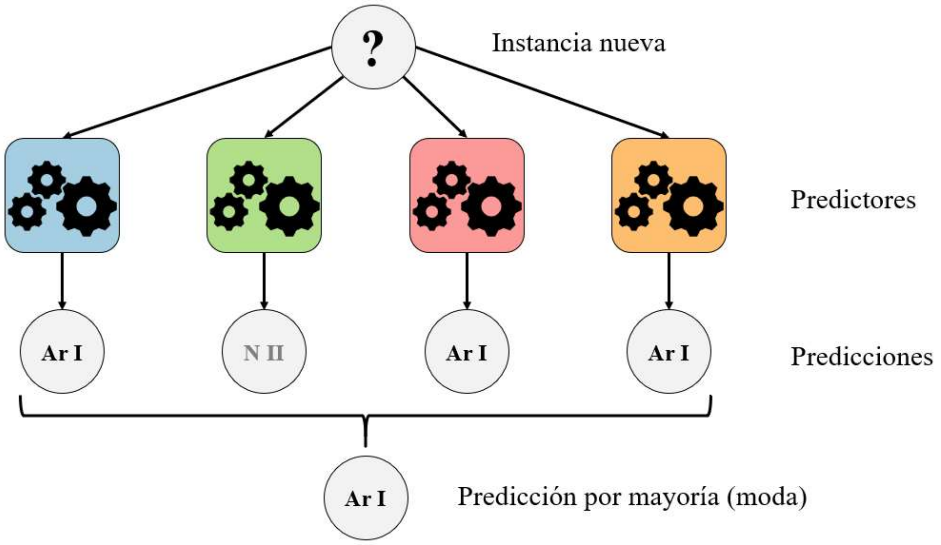


Figura 3.2.10.1. Predicciones de clasificación por votación fuerte.

La predicción de la etiqueta para la clase $\hat{y}$ por voto mayoritario (pluralidad) de cada clasificador $\varphi_{\mathcal{L}}(X)$ se expresa:

$$\hat{y} = moda\{\varphi_{\mathcal{L}1}(X), \varphi_{\mathcal{L}2}(X), \dots, \varphi_{\mathcal{L}N}(X)\} \tag{3.8}$$

Dónde:

N Es el número del clasificador.

Se tiene así con las predicciones de la Figura 3.2.10.1 que $\hat{y} = moda\{Ar\ I, N\ II, Ar\ I, Ar\ I\} = Ar\ I$.

La segunda técnica es ***Votación Suave*** (Figura 3.2.10.2), aquí se consideran las probabilidades de predicción de cada modelo, y la clase a predecir es aquella que obtuvo la probabilidad de predicción más alta.

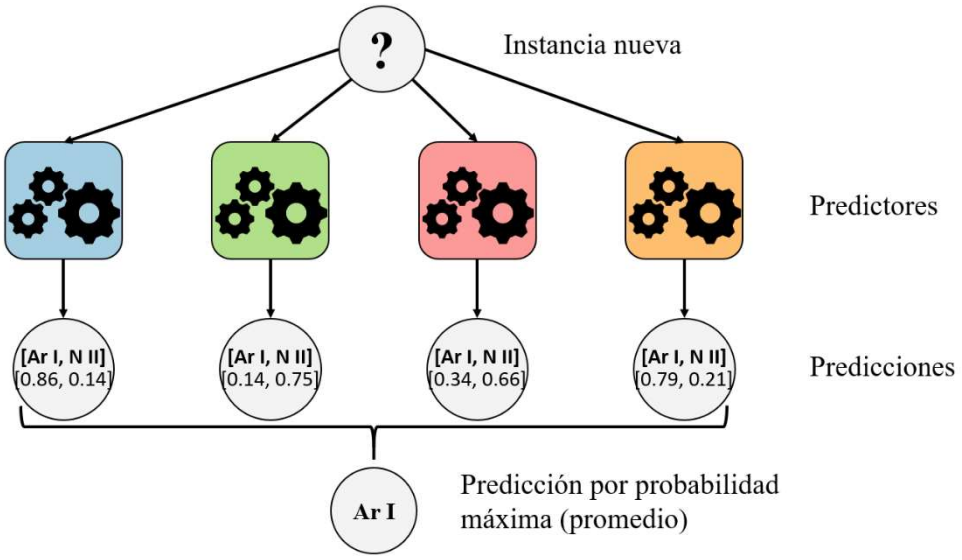


Figura 3.2.10.2. Predicciones de clasificación por votación suave.

La predicción $\hat{y}$ de las etiquetas de clase se basa en la probabilidad máxima de las predicciones p_{ij} para el clasificador $\varphi_{\mathcal{L}}$ así:

$$\hat{y} = arg\ max_i \sum_{j=1}^{m} p_{ij} \tag{3.9}$$

Dónde:

m	Número total de clasificadores.
i	Conjunto único de etiquetas de clase.
j	Es el j-ésimo clasificador.

Por ejemplo, para la Figura 3.2.10.2: i_0 es Ar I, i_1 es N II, y el promedio de sus probabilidades se calcula como:

$$p(i_0 j) = \frac{0.86 + 0.14 + 0.34 + 0.79}{4} = \frac{2.13}{4} = 0.5325 \tag{3.10}$$

$$p(i_1 j) = \frac{0.14 + 0.76 + 0.66 + 0.21}{4} = \frac{1.77}{4} = 0.4425 \tag{3.11}$$

Por lo que $\hat{y}$ es $arg\ max_i$ de $[0.5325, 0.4425]$, lo que corresponde a i_0, y que equivale a Ar I.

3.2.11 Corrección del Desplazamiento Óptico.

Los datos proporcionados por el Laboratorio de Física de Plasmas del ININ tienen un desplazamiento óptico, y la corrección de este desplazamiento se hizo con tres conjuntos de archivos de la lámpara de calibración HG-1 (Ar y Hg) de Ocean Optics™, cada conjunto corresponde a una ejecución experimental integrada por treinta y tres archivos, cada archivo contiene 1,024 observaciones y dos columnas, la primera es la longitud de onda, y la segunda la intensidad. No se requieren más porque el error en el espectrómetro se mantiene en el tiempo a menos que este equipo sufra un golpe o descompostura. En caso de requerir una recalibración, debe repetirse la rutina de corrección de desplazamiento óptico.

El procedimiento para corregir este desplazamiento óptico consiste en:

1. Se busca la hoja de datos de la lámpara de calibración HG-1 de Ocean Optics™ y se obtienen las longitudes de onda con sus respectivas especies, éstas se guardan en el arreglo **x_hg1**.
2. Graficar los datos de lámpara de calibración e identificar las longitudes de onda en las que se encuentran los picos y se guardar en un arreglo **x_exp**.
3. Los pasos 1 y 2 se repiten para cada ejecución experimental y se concatenan.
4. Una vez obtenidas todas las longitudes de onda de las ejecuciones experimentales de lámpara de calibración HG-1 de Ocean Optics™, se calcula su diferencia (3.12) y se almacena en arreglos independientes de la siguiente manera:

$$\mathrm{x\_dif}_i = \mathrm{x\_hg1}_\mathrm{i} - \mathrm{x\_exp}_i \tag{3.12}$$

Dónde:

x_hg1 Vector de longitudes de onda para los picos reportados en la hoja de datos de la lámpara de calibración HG-1.

x_exp Vector de longitudes de onda para los picos detectados en espectro experimental de la lámpara de calibración HG-1.

x_dif Vector con diferencias de longitud de onda.

i i-ésimo valor de un arreglo.

5. Se calculan los coeficientes (3.13) de una regresión lineal por cada ejecución experimental de la siguiente manera:

$$\alpha, \beta 1, \beta 2 = regresión\_lineal(x\_\exp_i, x\_dif_i) \quad (3.13)$$

Dónde:

α Punto donde la recta intercepta el eje de las ordenadas.

$\beta 1$ Coeficiente para: $\beta 1 \cdot 1$, desplazamiento en eje de las abscisas.

$\beta 2$ Coeficiente para: $\beta 2 \cdot x\_\exp_i$, pendiente de la recta.

6. Se calculan los coeficientes de una regresión polinomial de grado 2 por cada ejecución experimental de la siguiente manera:

$$\alpha, \beta 1, \beta 2, \beta 3 = regresión\_polinomial\ (x\_\exp_i, x\_dif_i)$$

Dónde:

α Punto donde la curva intercepta el eje de las ordenadas.

$\beta 1$ Coeficiente para: $\beta 1 \cdot 1$, desplazamiento en eje de las abscisas.

$\beta 2$ Coeficiente para: $\beta 2 \cdot x\_\exp_i$

$\beta 3$ Coeficiente para: $\beta 2 \cdot x\_\exp_i{}^2$

7. Con los coeficientes se implementan las funciones de predicción (3.14) y (3.15) para observar su comportamiento respecto al desplazamiento óptico.

$$fl(x\_exp_i) = \beta 1 + \beta 2 \cdot x\_exp_i \quad (3.14)$$

$$fp(x\_exp_i) = \beta 1 + \beta 2 \cdot x\_exp_i + \beta 3 \cdot x\_exp_i{}^2 \quad (3.15)$$

Dónde:

$fl(x\_exp_i)$ Vector corregido de longitudes de onda.

$fp(x\_exp_i)$ Vector corregido de longitudes de onda.

8. El comportamiento observado en la Figura 3.2.11.1 muestra una diferencia entre la regresión lineal y la regresión polinomial.

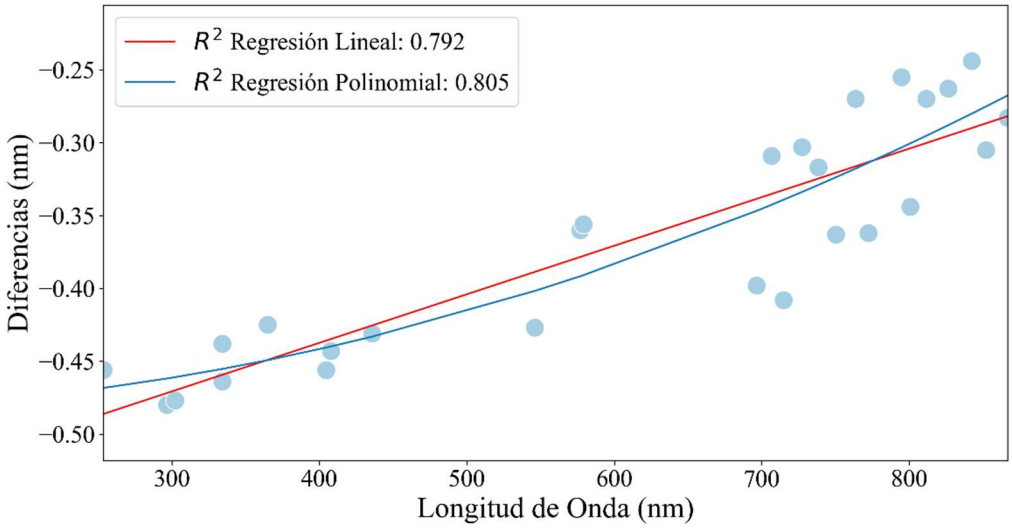


Figura 3.2.11.1. Regresión lineal y regresión polinomial para cada ejecución experimental de la lámpara de calibración HG-1 concatenado en un único archivo.

9. Los valores R^2 (coeficiente de determinación) mostrados en la Figura 3.2.11.1 se calcula con la ecuación (3.16), este valor explica el porcentaje de varianza existente entre el valor real y el valor predicho.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{3.16}$$

Dónde:

y_i Valor actual x_dif$_i$.

$\hat{y}_i$ Valor predicho $fl(x\_exp_i)$ o $fp(x\_exp_i)$.

$\bar{y}$ Media de y_i.

En la Tabla 3.2.11.1 se muestra que la regresión lineal tiene un valor R^2 menor que la regresión polinomial, por lo que esta última es la que explica mejor la varianza de los datos alrededor de la línea.

Tabla 3.2.11.1. Valores de R^2 para cada ejecución experimental de la lámpara de calibración HG-1.

Regresión	R^2	α	$\beta 1$	$\beta 2$	$\beta 3$
Lineal	0.791903	-0.570907	0.000333	-	-
Polinomial	0.805007	-0.48287	0	-0.000022	0.0000003115681

10. En la Figura 3.2.11.2 se observa el efecto de restar al vector de longitudes de onda para los picos reportados en la hoja de datos de la lámpara de calibración HG-1 (x_hg1) el vector de longitudes de onda para los picos detectados en espectro experimental de la lámpara de calibración HG-1 (x_exp), así como la corrección de desplazamiento óptico con el uso de las funciones (3.14) y (3.15).

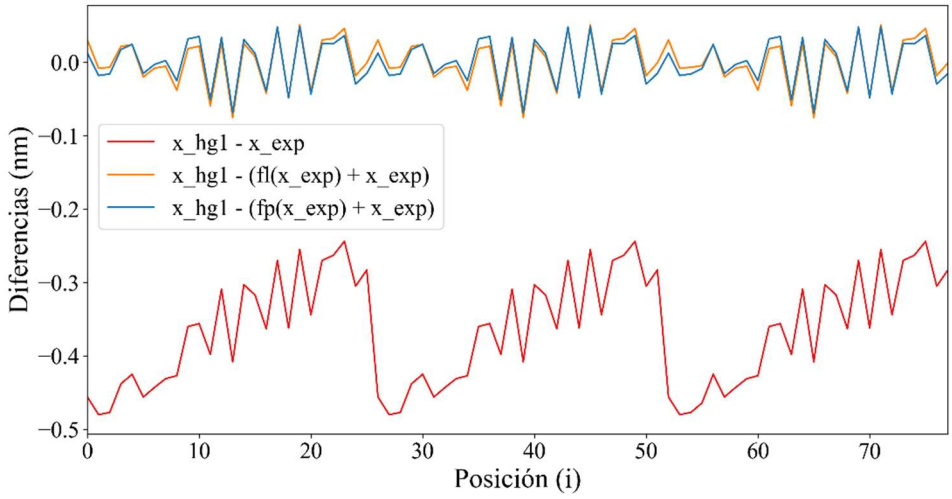


Figura 3.2.11.2. Regresión lineal sin valores atípicos para cada corrida experimental de la lámpara de calibración HG-1.

En la Figura 3.2.11.3 se observa la corrección del espectro experimental de la lámpara de calibración HG-1 utilizando la regresión lineal y polinomial.

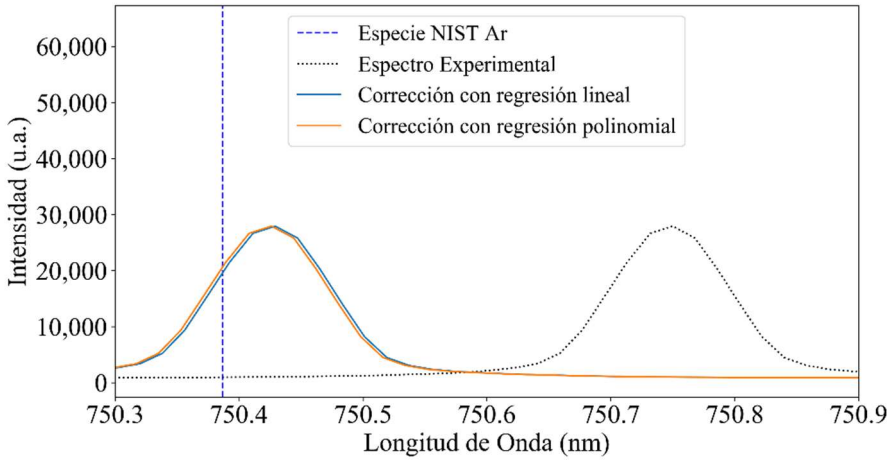


Figura 3.2.11.3. Corrección de desplazamiento óptico en espectro de lámpara HG-1.

Se concluye así que el desplazamiento óptico inducido por el espectrómetro es independiente del tipo de especie de estudio, y como sugieren las Figura 3.2.11.2 y Figura 3.2.11.3, la corrección del espectro experimental con regresión lineal y regresión polinomial tienen diferencias mínimas, por lo que se incluirán ambas en la interfaz gráfica y la elección dependerá del usuario.

3.2.12 Optimización de Hiperparámetros.

Los hiperparámetros son el conjunto de propiedades de configuración que definen un modelo. Hasta este punto se tienen: a) los datos del NIST, b) espectros sintéticos, c) un conjunto de algoritmos clasificadores, d) la función de ajuste y como adición, e) un conjunto de 3 ejecuciones experimentales que corresponden a una lámpara de calibración HG-1 utilizada en el Laboratorio de Física de Plasmas del ININ. Sin embargo, los algoritmos aún no están optimizados para hacer sus mejores predicciones, por lo que es necesario realizar ajuste de hiperparámetros, para esto se realiza el siguiente procedimiento:

1. Definir un diccionario que tiene como clave el hiperparámetro y como valores una lista de opciones. En la Figura 3.2.12.1 se muestra un diccionario con los hiperparámetros a optimizar: *max_features* (4), *criterion* (3), *max_depth* (26), *n_estimators* (10), *bootstrap* (2), *min_samples_split* (3) y *min_samples_leaft* (3).

La cantidad de opciones se contabiliza e indica al final de cada instrucción, dónde el espacio total de exploración es: $4 \times 3 \times 33 \times 91 \times 2 \times 9 \times 5 = 3{,}243{,}240$.

```
_GRID_RFC = {
    # número de características a considerar por cada división
    'max_features':      ['auto', 'sqrt', 'log2', None],                          # 4
    # Criterio
    'criterion':         ['gini', 'entropy', hellinger],                          # 3
    # número máximo de niveles en el árbol
    'max_depth':         [int(x) for x in linspace(3, 35, num=33)],               # 33
    # número árboles
    'n_estimators':      [int(x) for x in linspace(start=10, stop=100, num=91)],# 91
    # Método de selección de muestras para cada árbol de entrenamiento
    'bootstrap':         [True, False],                                           # 2
    # Número mínimo de muestras requeridas para dividir un nodo
    'min_samples_split': [2, 3, 4, 5, 6, 7, 8, 9, 10],                            # 9
    # Número mínimo de muestras requeridas en cada nodo hoja
    'min_samples_leaf':  [1, 2, 3, 4, 5]                                          # 5
}
```

Figura 3.2.12.1. Diccionario con opciones de búsqueda de hiperparámetros.

2. Una técnica de búsqueda de hiperparámetros es ***GridSearch*** (Figura 3.2.12.2). Aquí se explora cada combinación posible de parámetros y al final se muestra aquella que tiene la métrica más alta. La principal desventaja es el tiempo que toma explorar todas las opciones dado el costo computacional que toma explorar todas las opciones. Si se considera que hay 56,160 opciones y cada una toma en promedio 30 segundos para completarse, la búsqueda total tomaría 19.5 días.

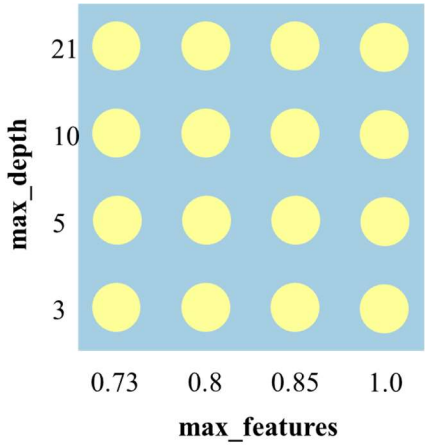


Figura 3.2.12.2. Representación gráfica de un GridSearch bidimensional.

3. Otra técnica de búsqueda de hiperparámetros es RandomGridSearch (Figura 3.2.12.3). Esta consiste en elegir un subconjunto de datos y gradualmente elegir la

combinación con la mejor solución local sin garantía de que se trate de la mejor solución global.

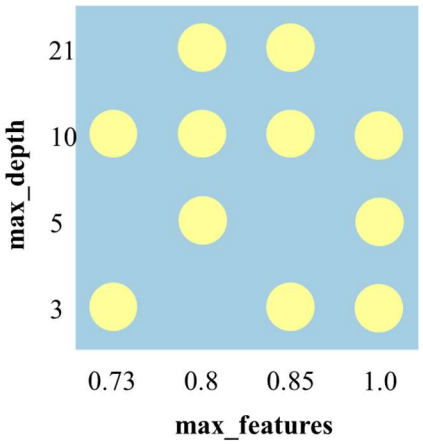


Figura 3.2.12.3. Representación gráfica de un RandomGridSearch bidimensional.

Los parámetros elegidos para optimización de hiperparámetros son los que se detallan en la Tabla 3.2.12.1

Tabla 3.2.12.1. Parámetros para optimizar por algoritmo.

Algoritmo(s)	Parámetro	Rango/ Opciones	Descripción
Decision Tree ***Random Forest*** ***Extremely Randomized Trees***	max_features	auto, sqrt, log2, None (4)	Características máximas.
Decision Tree ***Random Forest*** ***Extremely Randomized Trees***	criterion	gini, entropy, hellinger (3)	Función de impureza.
Decision Tree ***Random Forest*** ***Extremely Randomized Trees***	max_depth	[3..35] (33)	Profundidad máxima.
Decision Tree ***Random Forest*** ***Extremely Randomized Trees***	min_samples_split	[2..10] (9)	Cantidad mínima de muestras para dividir un nodo.
Decision Tree ***Random Forest*** ***Extremely Randomized Trees***	min_samples_leaf	[1..5] (5)	Cantidad mínima de muestras por cada nodo hoja.
Bagging	n_estimators	[10...100]	Número de estimadores.

Algoritmo(s)	Parámetro	Rango/ Opciones	Descripción
Random Forest		(91)	
Extremely Randomized Trees			
Bagging	max_samples	[0.63..1.0] (38)	Porcentaje de muestras para cada estimador.
Bagging	bootstrap	True, False (2)	Extraer muestras con reemplazo
Random Forest			
Extremely Randomized Trees			
Bagging	bootstrap_features	True, False (2)	Extraer características con reemplazo
Bagging	base_estimator	default, optimizado (2)	Clasificador por defecto o el optimizado en búsqueda de hiperparámetros

En el capítulo 4 se detallan las mejores combinaciones encontradas de hiperparámetros, así como los resultados obtenidos con el repositorio local de especies del NIST y los espectros sintéticos generados.

3.3 TEMPERATURA DE EXCITACIÓN ELECTRÓNICA.

Para estimar la temperatura de las especies se requiere del conocimiento y experiencia del usuario. Esta etapa requiere de las predicciones de las especies, así como de las características de intensidad y longitud de onda de cada pico, para posteriormente consultarse con los datos reportados en el NIST. A continuación, se describe el procedimiento empleado para estimar la temperatura.

3.3.1 Recolección de Datos.

Para la estimación de temperatura los datos de las predicciones realizadas y los datos reportados en el NIST deben ser conjuntados en un *DataFrame*, para agilizar el proceso de consulta y cálculo. En la Tabla 3.3.1.1 se muestran en color rojo las columnas requeridas de la predicción, y en color azul las columnas del NIST, los nombres de estas columnas son los que se usaron en el *DataFrame*.

Tabla 3.3.1.1. Campos usados para estimar la temperatura electrónica.

Columna	Descripción
longitud_onda_ama	Se traza una línea horizontal a media altura de cada pico detectado, posteriormente se calcula el punto medio y se usa como valor de longitud de onda.
longitud_onda_xp_cdo	Este valor requiere de la corrección del desplazamiento óptico y de la detección de picos en el espectro, cada pico detectado se guarda como una longitud de onda.
intensidad_xp_cfc	Al aplicar la corrección de fondo continuo se calcula la altura a la que se encuentra cada pico detectado y se almacena este valor.
clase_1	Los valores almacenados en esta columna corresponden a la especie predicha por los algoritmos de Machine Learning.
obs_wl_X(nm)	Longitud de onda observada en nanómetros.
Aki(s^-1)	Probabilidad de transición.

Ek(eV)	Energía en eV.
g_k	Peso estadístico.
clase	Especie reportada.
ln((I*L)/(g*A))	Es una columna calculada, donde: • I: intensidad_xp_cfc, • L: longitud_onda_* • g: g_k • A: Aki(s^-1)

3.3.2 Estimación de la Temperatura de Excitación Electrónica.

La estimación de temperatura se realiza buscando en el NIST las especies identificadas en los picos del espectro de emisión óptica para ser reemplazados en las ecuaciones (3.17) [11] y (3.18) [47].

$$\ln\left(\frac{I \cdot A}{A_{ki} \cdot g_k}\right) = -\frac{E_k}{k \cdot T} \tag{3.17}$$

$$\ln\left(\frac{I \cdot \lambda}{A_{ki} \cdot g_k}\right) = -\frac{E_k}{k \cdot T} \tag{3.18}$$

Dónde:

I Intensidad de la línea *(u.a.)*.

A Área bajo un pico considerando la anchura a media altura *(u.a.)*.

λ Longitud de onda *(nm)*.

A_{ki} Probabilidad de transición *(s^{-1})*.

g_k Peso estadístico *(u.a.)*.

E_k Energía *(eV)*.

k Constante de Boltzmann.

T Temperatura de Excitación Electrónica *(K)*.

A partir de (3.17) se obtienen las siguientes ecuaciones:

$$\mathrm{y} = \ln\left(\frac{I \cdot A}{A_{ki} \cdot g_k}\right), y = \ln\left(\frac{I \cdot \lambda}{A_{ki} \cdot g_k}\right) \tag{3.19}$$

$$\mathrm{x} = E_k \tag{3.20}$$

$$m = \frac{1}{k \cdot T} \tag{3.21}$$

Dado que se identifica una relación lineal entre x e y, se les aplica una regresión lineal, se obtiene su pendiente m, y se obtiene la temperatura despejando T de (3.21)

En la Tabla 3.3.2.1 se observa un fragmento de los datos de la predicción para el espectro de la lámpara de calibración HG-1 junto con sus especies correspondientes con el NIST, así como tres campos calculados.

Tabla 3.3.2.1. Datos de ejemplo para estimar la temperatura de excitación electrónica.

	especie_pred_1	longitud_onda_ama	intensidad	área_simpson	área_trapecio	Ek(eV)	Aki(s^-1)	g_k	ln((I*L)/(g*A))	ln((I*área_simpson)/(g*A))	ln((I*área_trapecio)/(g*A))
0	Hg I	253.639166	63460.882772	10875.655889	10773.030930	4.886495	8400000.0	3	-21.171529	-22.935426	-22.944907
1	Hg I	296.746122	1899.339479	205.740020	202.878033	8.844537	46000000.0	3	-26.223892	-28.446540	-28.460548
2	Hg I	364.990710	12191.996389	1299.424553	1282.639197	8.856338	129000000.0	7	-26.036092	-28.274950	-28.287952
3	Hg I	365.421799	1804.671315	116.234866	114.842324	8.851985	18400000.0	5	-25.661379	-28.403900	-28.415953
4	Hg I	404.671363	43928.224871	4408.602263	4357.859334	7.730455	20700000.0	3	-21.974135	-24.273134	-24.284711
5	Hg I	407.786584	2936.337132	304.535952	300.845010	7.926077	4000000.0	1	-21.929409	-24.195538	-24.207732
6	Hg I	434.767580	936.032683	83.744932	83.148910	9.554714	8400000.0	5	-25.359984	-27.773859	-27.781001
7	Hg I	435.830345	63466.571957	10365.474416	10289.418539	7.730455	56000000.0	3	-22.527219	-24.339252	-24.346616
8	Hg I	576.938795	63468.385835	9647.017284	9574.334003	8.851985	23600000.0	5	-21.893428	-23.777321	-23.784884
9	Hg I	579.030503	63462.758434	9295.778956	9231.181130	8.844171	31000000.0	5	-22.162638	-24.083531	-24.090504
10	Ar I	696.584651	9163.971387	853.636398	845.561912	13.327857	6400000.0	3	-21.824462	-24.197992	-24.207496

Las filas que se seleccionan para estimar la temperatura de excitación electrónica deben tener las siguientes características:

- Pertenecer al mismo elemento.
- Encontrarse en ambos extremos, en base a E_k (mostrado como **Ek(eV)** en la Tabla 3.3.2.1).

Al sustituir en las ecuaciones (3.19), (3.20) los datos de la Tabla 3.3.2.1, tomando como criterio de selección los 3 menores y los 3 mayores de E_k para aplicar una regresión lineal y obtener la pendiente con sustitución en (3.21) con despeje sobre T, se obtiene:

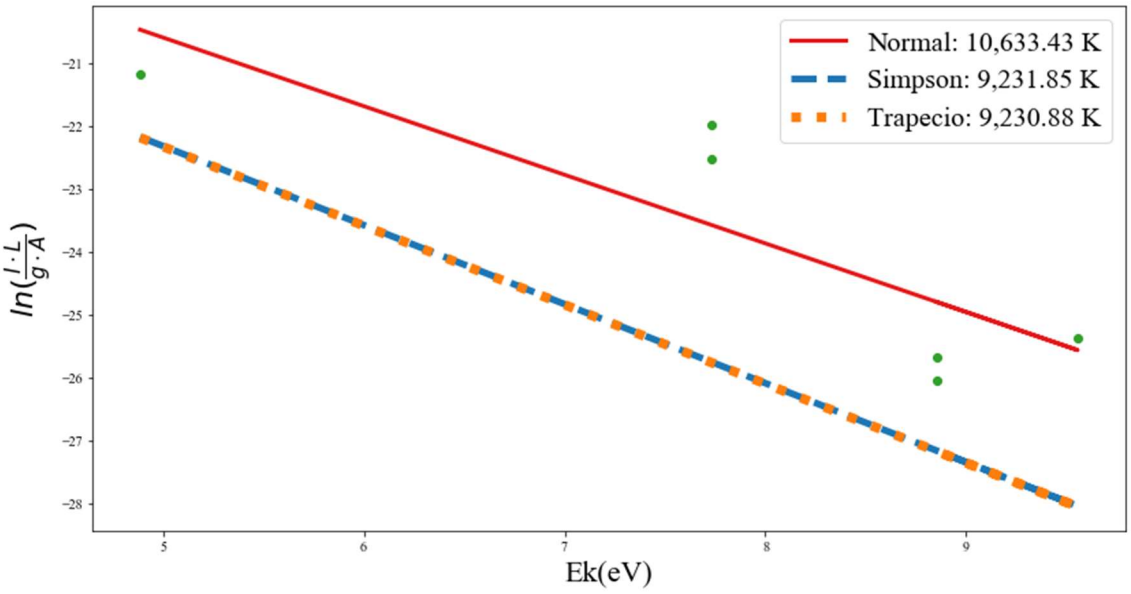


Figura 3.3.2.1. Temperatura estimada en tres formas con el espectro de la lámpara de calibración HG-1.

Se puede observar en la Figura 3.3.2.1 que la temperatura de excitación electrónica es similar en los 3 casos, por *Normal* se entiende cuando en (3.19) se usa $y = \ln\left(\frac{I \cdot \lambda}{A_{ki} \cdot g_k}\right)$, y *Simpson* o *Trapecio* cuando estos métodos de integración bajo la curva reemplazan λ, y dependiendo del método usado para obtener el área bajo el pico seleccionado, los valores de temperatura se ven afectados. De Simpson y Trapecio se obtienen valores semejantes casi idénticos por lo que se puede seleccionar como métodos adecuados y se descarta el método Normal por generar un valor de temperatura que se puede considerar sobreestimado. En la mayoría de los casos se puede optar por tomar el valor bajo, aunque pueda considerarse subestimado.

3.4 DESARROLLO DE LA INTERFAZ GRÁFICA.

En este capítulo se describe la implementación de la interfaz gráfica de usuario que reúne el tratamiento de datos mostrados en las secciones 3.1, la caracterización automática de especies de la sección 3.2, y la estimación de la temperatura de excitación electrónica vista en la sección 3.3. Se integran los conceptos para formalizar la interfaz de usuario con base a los requerimientos y especificaciones.

3.4.1 Requerimientos o especificaciones.

Para llevar a cabo la interfaz gráfica de usuario (Graphical User Interface, GUI por sus siglas en inglés) fue necesario establecer las especificaciones requeridas en el Laboratorio de Física de Plasmas del ININ. Los requerimientos funcionales se observan en la Figura 3.4.1.1 y es el resultado de:

1. Importar espectros en formato CSV.
2. Caracterización automática de las especies de los elementos He, N, O, Ar y Hg.
3. Corrección de línea base y desplazamiento óptico.
4. Estimar temperatura de excitación electrónica.
5. Exportar caracterización con etiquetas en formato PNG (Portable Network Graphics por sus siglas en inglés) y CSV.

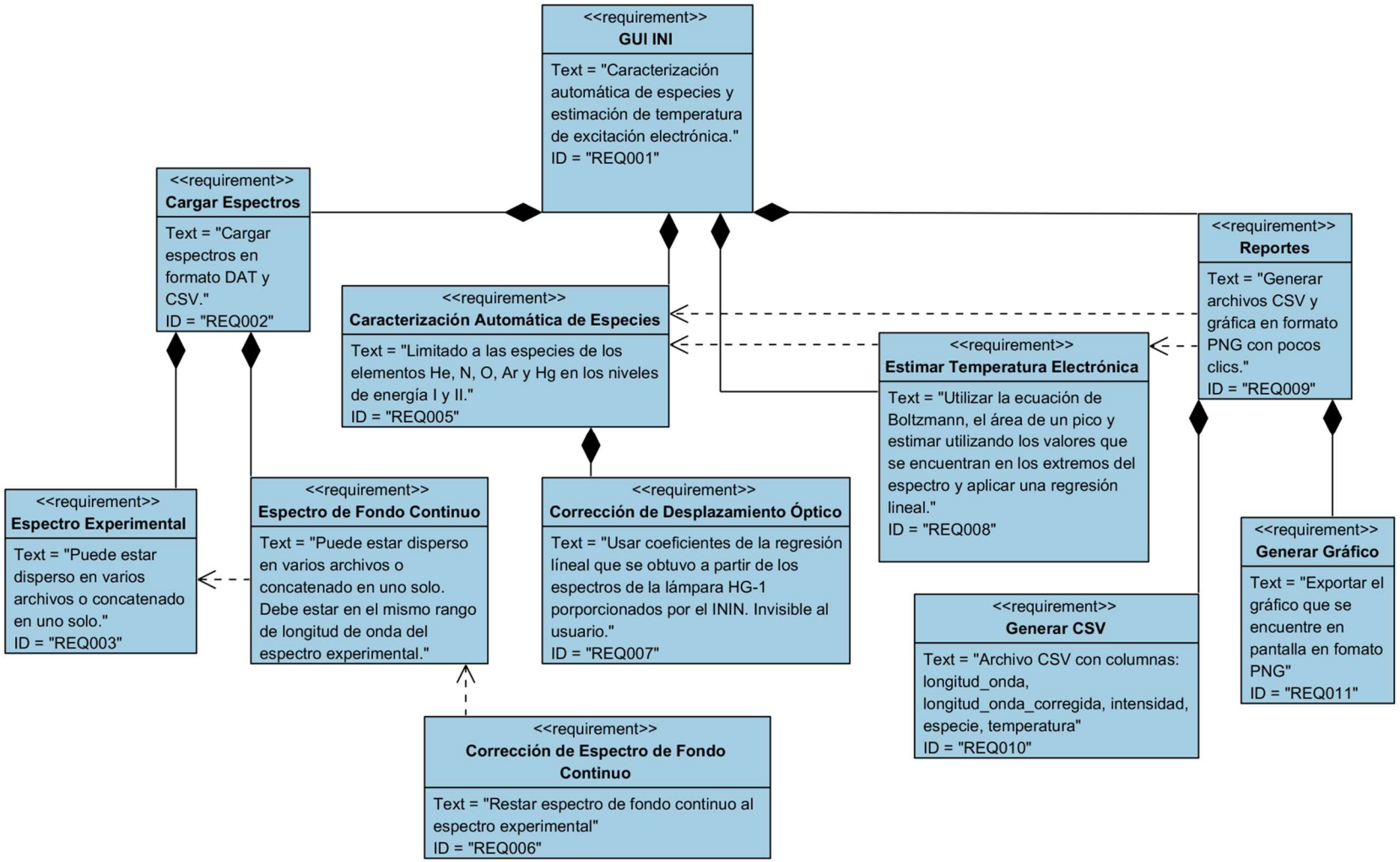


Figura 3.4.1.1. Diagrama de requerimientos funcionales para la interfaz gráfica a desarrollar del Laboratorio de Física de Plasmas del ININ.

Por otra parte, los requerimientos no funcionales (Figura 3.4.1.2) o características de sistema son:

1. GUI sencilla de usar.
2. Manejo intuitivo de GUI, con poco o nula capacitación del usuario.
3. Ayuda en pantalla con *Tooltips*.
4. GUI legible.

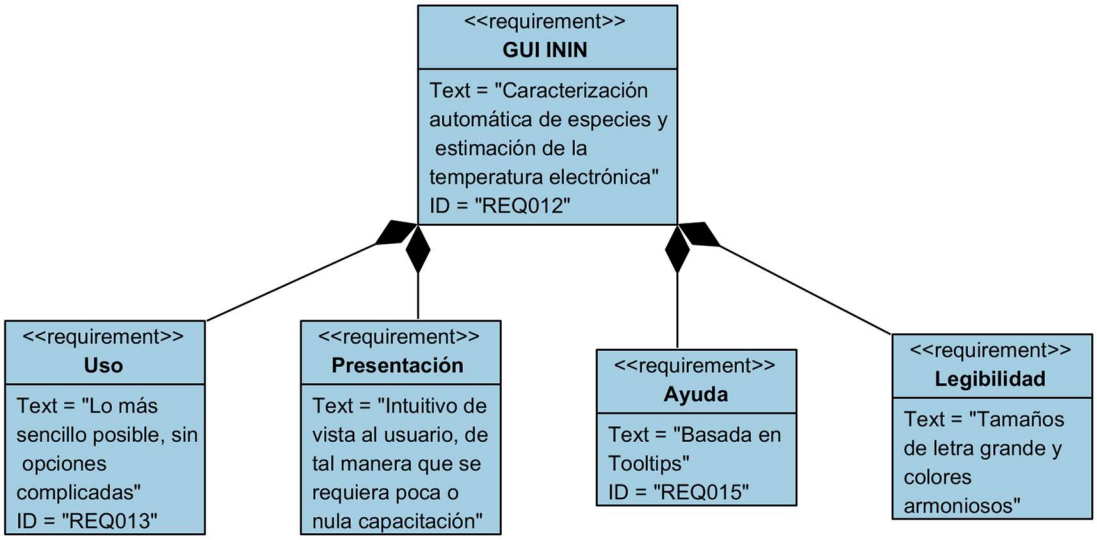


Figura 3.4.1.2. Diagrama de requerimientos no funcionales para la interfaz gráfica a desarrollar del Laboratorio de Física de Plasmas del ININ.

Estos requerimientos se encuentran restringidos a las siguientes especificaciones de hardware:

a) Procesador: Pentium Core i5 4570.
b) Memoria: 8 GB.
c) Sistema Operativo: Windows 8 de 64 bits.

3.4.2 Implementación de la Interfaz Gráfica de Usuario

Con base a los diagramas de requerimientos se diseñó una interfaz gráfica con una ventana y seis secciones: a) carga de datos, b) configuración, c) graficar espectro, d) estimar temperatura, e) exportar datos, f) información. En la Figura 3.4.2.1 se muestra el diagrama de casos de uso para esta interfaz gráfica.

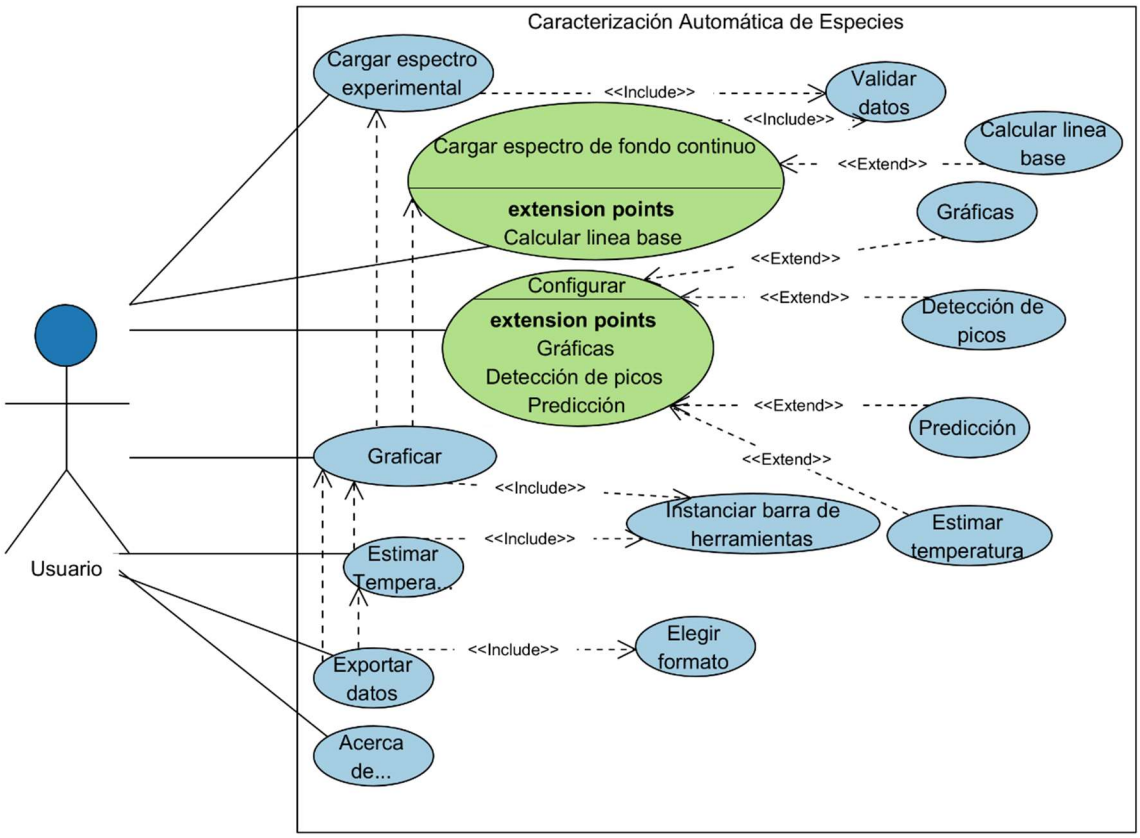


Figura 3.4.2.1. Diagrama de casos de uso para la interfaz propuesta.

En la Figura 3.4.2.2 se muestra el flujo de datos y las operaciones aplicadas desde la carga de espectros hasta la caracterización de especies y la estimación de temperatura, y los nemotécnicos utilizados se describen en la Tabla 3.4.2.1.

Tabla 3.4.2.1. Nemotécnicos utilizados en los diagramas de flujo.

Nemotécnico	**Descripción**
Df	DataFrame.
df_fc	DataFrame de fondo continuo.
df_xp	DataFrame de espectro experimental.
df_pred	DataFrame de predicciones.
X_fc	Longitud de onda de fondo continuo
y_fc	Intensidad de fondo continuo.
X_xp	Longitud de onda de espectro experimental.
y_xp	Intensidad de espectro experimental.
X_xp_cdo	Longitud de onda de espectro experimental con corrección de desplazamiento óptico.
y_xp_cdo	Intensidad de espectro experimental con corrección de espectro de fondo continuo.

.

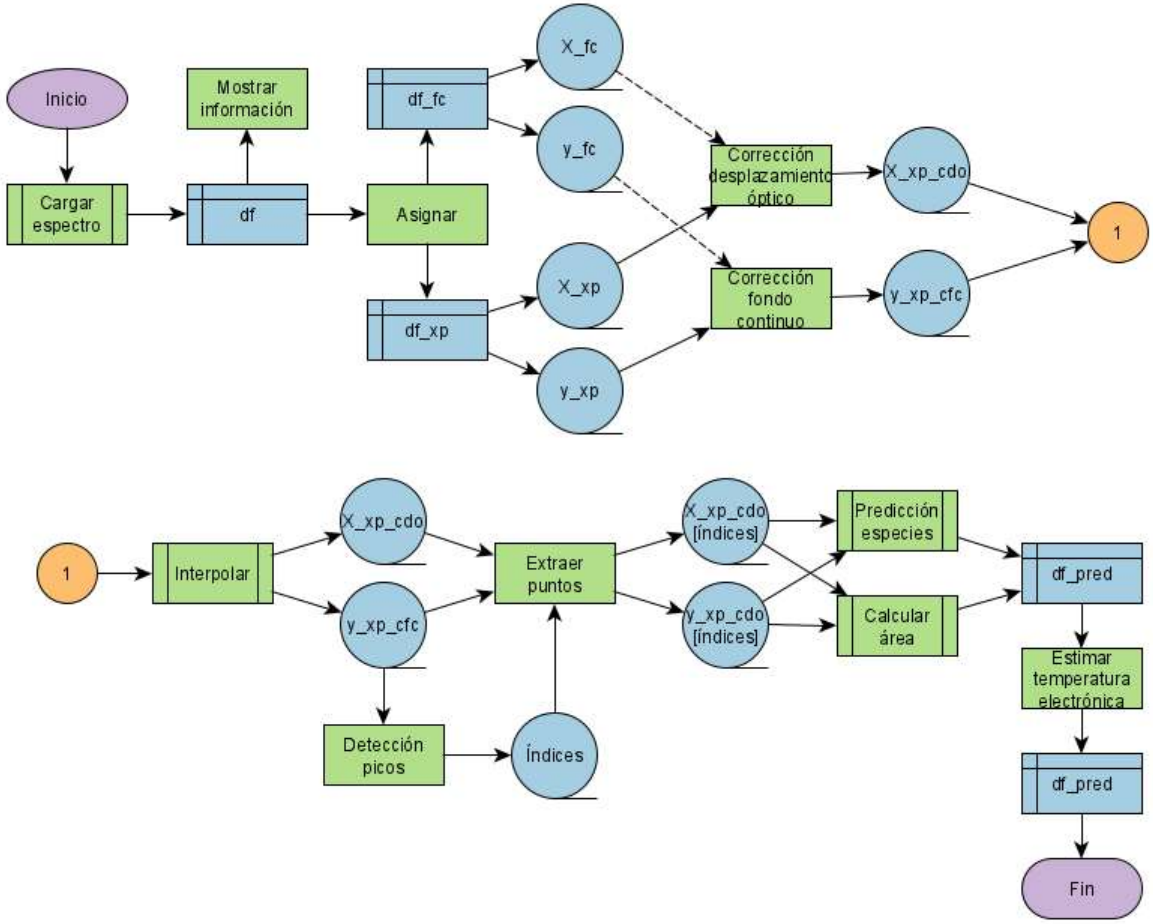


Figura 3.4.2.2. Flujo de datos utilizado en la GUI.

Los apartados que integran la GUI se describen a continuación.

3.4.3 Carga de Datos.

Este apartado está divido en dos pestañas. La primera pestaña se nombra ***Espectro*** (Figura 3.4.3.1). Aquí se define si el tipo de archivo es ***CSV*** o ***DAT***, en el caso de ***CSV*** debe elegirse un delimitador. Además, debe indicarse si la ***Captura*** del espectro fue con ***Aire*** o al ***Vacío***. Una vez que se da clic al botón ***Cargar*** y se eligen los archivos del espectro, en el grupo de opciones ***Información*** se mostrará: ***Número de Archivos*** cargados, ***Número de Líneas*** de especies, ***Número de Columnas*** y el ***Espacio en Memoria*** ocupado en ***MiB***.

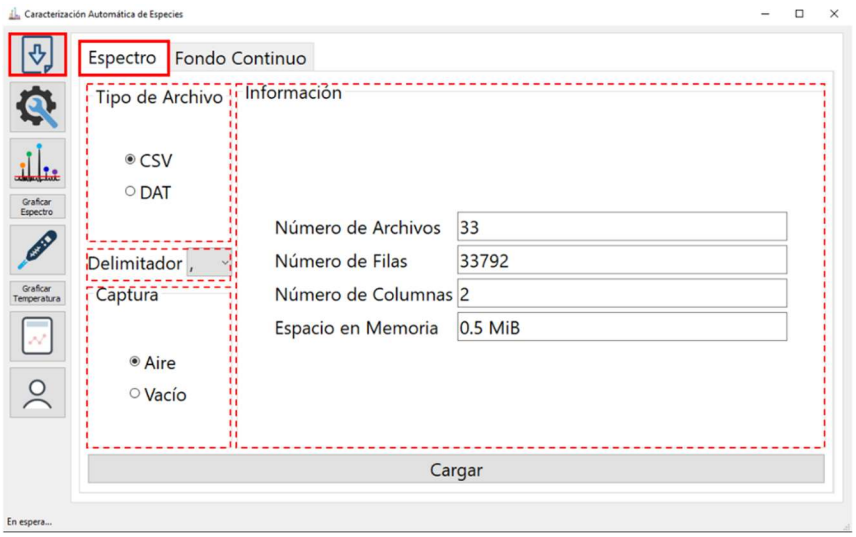


Figura 3.4.3.1. Pestaña Espectro en el apartado carga de datos.

La segunda pestaña es Fondo Continuo (Figura 3.4.3.2). Aquí se encuentra la misma información vista en la Figura 3.4.3.1 con excepción del apartado ***Captura***. Y con la diferencia de que el ***Espacio en Memoria*** ocupado se indica en ***KiB***.

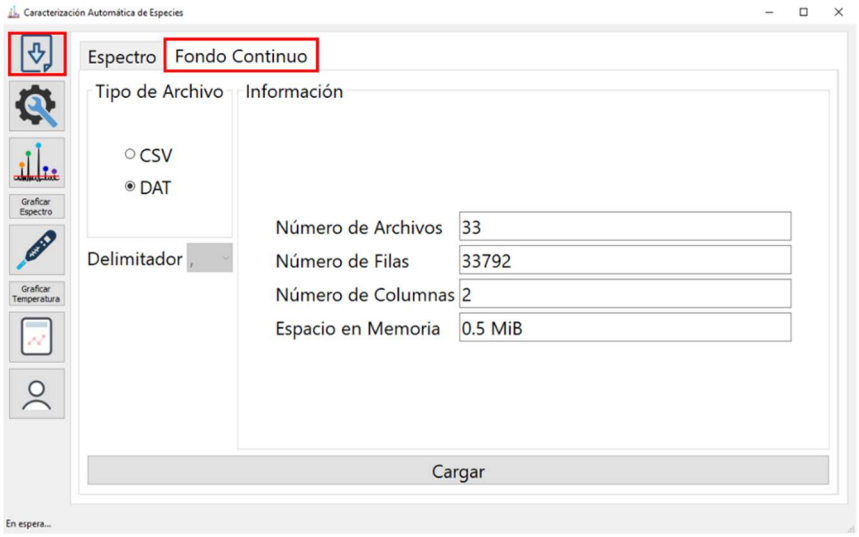


Figura 3.4.3.2. Pestaña Fondo Continuo en el apartado de carga de datos.

Es importante aclarar que las unidades de espacio en memoria están basadas en ISO/IEC 80000-13, por lo que un kibibyte (***KiB***) equivale en 2^{10} bytes, y un mebibyte (**MiB**) corresponde a 2^{20} bytes.

3.4.4 Configuración.

Este apartado tiene cuatro grupos de opciones. Las opciones del grupo ***Gráficas*** (Figura 3.4.4.1) son: a) ***Interpolar*** el espectro indicando la cantidad de puntos y el método de interpolación, b) ***Espectro Experimental*** sirve para mostrar el espectro adquirido sin ninguna modificación, c) ***Espectro de Fondo Continuo*** es para mostrar su gráfica, d) ***Corrección de Desplazamiento Óptico*** con la opción de indicar el tipo de corrección ***Lineal*** o *Polinomial*, e) ***Corrección de Fondo Continuo*** grafica el espectro adquirido con cada corrección indicada, y f) ***Espectro Experimental Corregido*** es el resultado de aplicar todas las correcciones y esta opción está habilitada por defecto.

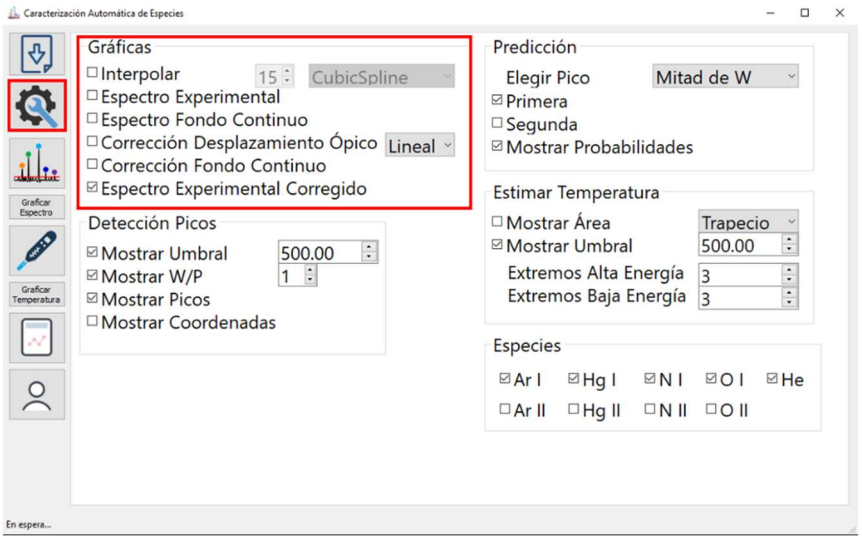


Figura 3.4.4.1. Grupo de opciones Gráficas en el apartado Configuración.

En el grupo de opciones ***Detección de Picos*** (Figura 3.4.4.2) se encuentran: a) ***Mostrar Umbral*** sirve para graficar una línea discontinua horizontal en el umbral indicado, si esta opción no se encuentra seleccionada no se mostrará el umbral, pero el valor indicado seguirá siendo usado para la detección de picos, b) ***Mostrar W/P*** muestra la anchura a media altura (FWHM, por sus siglas en inglés) y su prominencia (línea vertical desde la base hasta el pico detectado), el valor colocado en el *spinbox* frontal es la cantidad mínima de valores de longitud de onda que debe tener un pico para proceder con su detección, c) ***Mostrar Picos*** grafica un punto rojo con sombra azul en cada pico detectado, finalmente d) ***Mostrar Coordenadas*** coloca una etiqueta con longitud de onda e intensidad de cada pico.

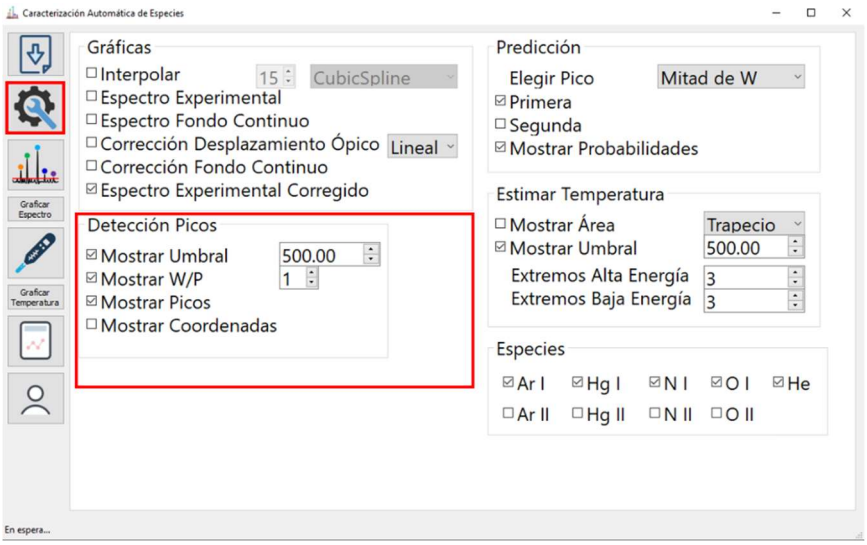


Figura 3.4.4.2. Grupo de opciones Detección de Picos en el apartado Configuración.

En el grupo de opciones ***Predicción*** (Figura 3.4.4.3) se encuentra el *combobox* de ***Elegir Pico*** dónde se puede elegir ***Experimental*** o ***Mitad de W***, el primero sirve para usar el pico detectado en el espectro experimental con la posibilidad de detectarlo desplazado de su centro real, el segundo calcula la anchura a media altura y el punto central de ésta, descrito como FWHF/2, el cual es usado como la posición en longitud de onda del pico (ver Figura 3.4.5.1). Para mostrar la ***Primera*** y ***Segunda*** predicción se selecciona estas opciones, además se pueden ***Mostrar Probabilidades*** eligiendo esta opción.

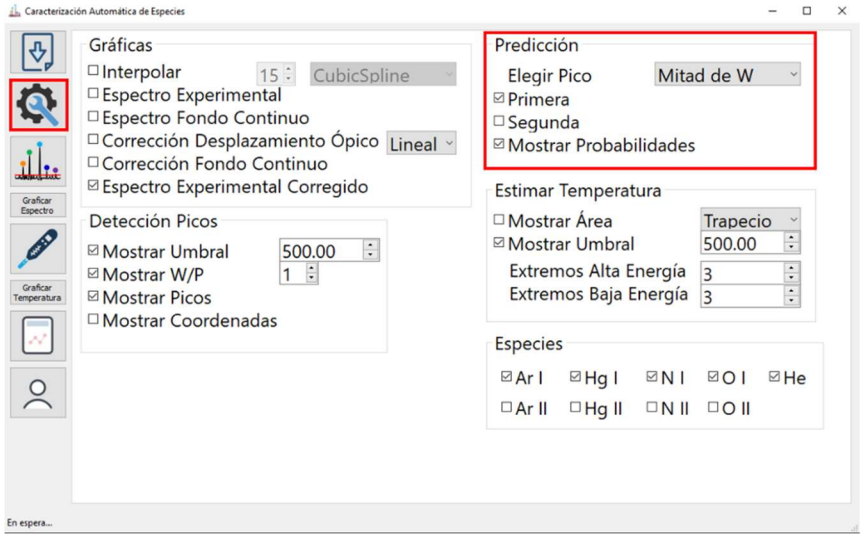


Figura 3.4.4.3. Grupo de opciones de Predicción en el apartado Configuración.

En el grupo de opciones ***Estimar Temperatura*** (Figura 3.4.4.4) se requiere indicar el ***Umbral*** en intensidad a partir del cual se usarán picos en los cálculos y la cantidad de puntos extremos de alta y baja energía. En este trabajo también puede estimar la temperatura con el área bajo la curva de cada pico detectado, para visualizar esta área basta seleccionar la opción ***Mostrar Área***.

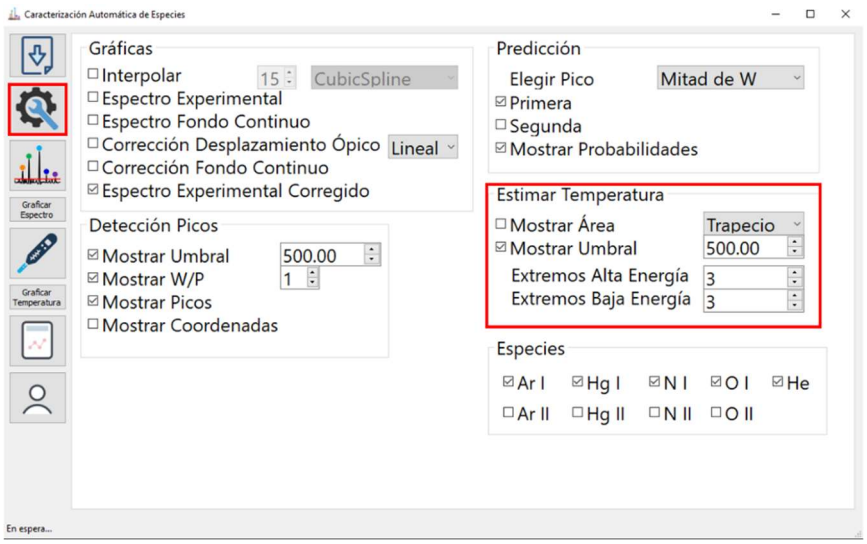


Figura 3.4.4.4. Grupo de opciones Estimar Temperatura en el apartado Configuración.

Finalmente, en el grupo de opciones ***Especies*** (Figura 3.4.4.5) se encuentran las nueve especies de estudio en este trabajo, y por medio de un *checkbox* es posible indicarle al modelo las que son de interés.

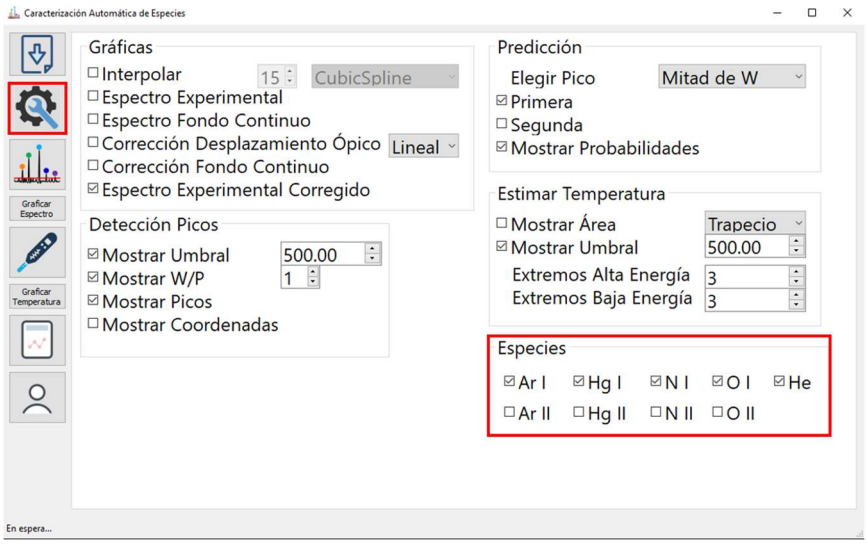


Figura 3.4.4.5. Grupo de opciones Especies en el apartado Configuración.

3.4.5 Graficar Espectro.

Este apartado es mostrado en la Figura 3.4.5.1, para visualizar el espectro hay que presionar la combinación de teclas ***Ctrl*** + ***G***. La primera vez que se ejecuta esta combinación de teclas mostrará una barra de herramientas que permite variar el nivel de acercamiento, desplazar el gráfico y guardar el resultado mostrado en pantalla.

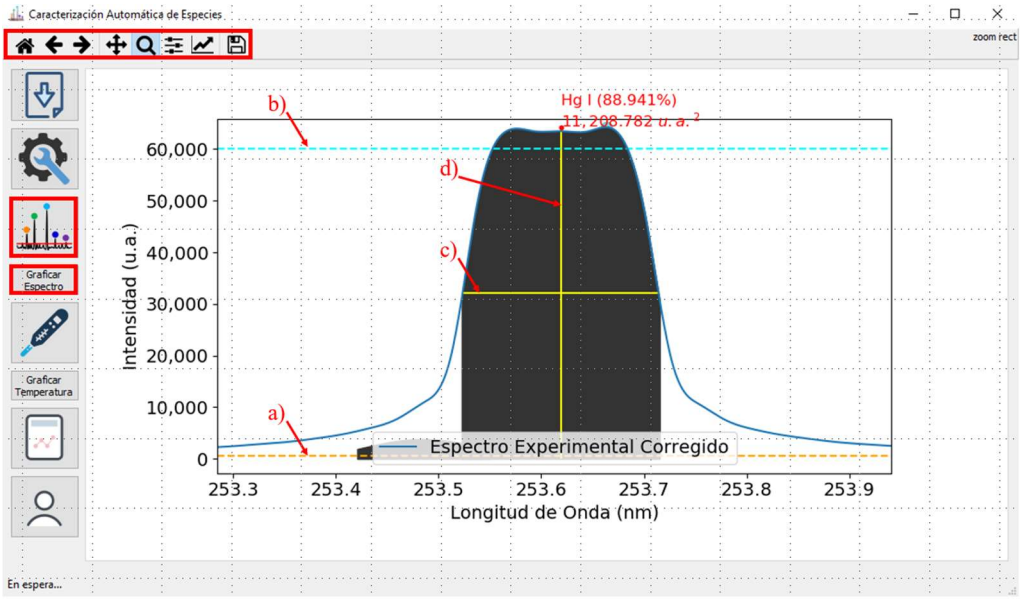


Figura 3.4.5.1. Apartado Graficar Espectro.

La Figura 3.4.5.1 está compuesta del conjunto de líneas: a) la línea punteada horizontal amarilla es el umbral en intensidad desde dónde inicia la detección de picos, b) la línea punteada horizontal azul es el umbral en intensidad a partir del cual se utilizan picos para estimar la temperatura, en este ejemplo se muestra con una intensidad de 60,000, sin embargo su valor por defecto cada vez que se inicia la interfaz gráfica es 0, c) la línea continua horizontal amarilla es la anchura a media altura, y d) es la altura desde el pico hasta su base. La etiqueta en rojo indica el elemento detectado, en paréntesis se muestra un porcentaje de certeza para dicho elemento, y a continuación el área bajo la curva.

3.4.6 Graficar Estimación de Temperatura de Excitación Electrónica.

En este apartado se muestra el resultado al estimar la temperatura de excitación electrónica. Se observa en la Figura 3.4.6.1 una línea punteada de color azul representa el umbral en intensidad para utilizar solo los picos que la superen en la estimación de la

temperatura de excitación electrónica, todos los picos que superen el umbral indicado en la interfaz gráfica son candidatos para buscar en los datos descargados del NIST, aquellos que pertenezcan al mismo elemento y se encuentren en los extremos de mayor y menor energía E_k para estimar la temperatura de excitación electrónica como se explica en la sección 3.3.2.

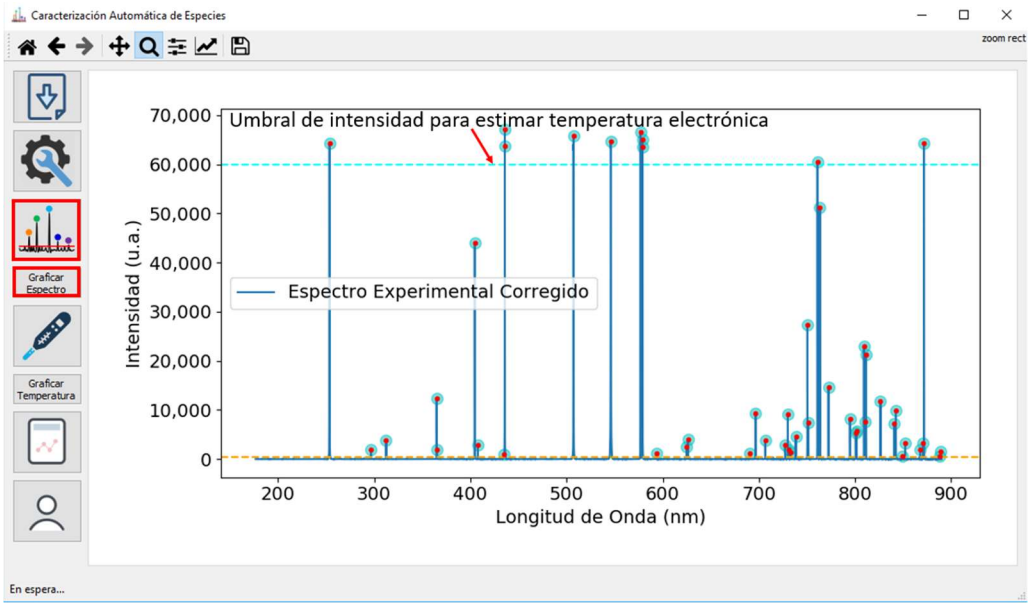


Figura 3.4.6.1. Espectro con umbral para detección de temperatura de la lámpara de calibración HG-1.

Al aplicar (3.17), (3.18), (3.19), (3.20) y (3.21), junto con una regresión lineal se obtiene el un gráfico con barra de herramientas independiente cuyo resultado mostrado en la Figura 3.4.6.2, este es sensible al valor de la pendiente y puntos elegidos automáticamente en los extremos del espectro, esta elección de puntos requiere de la experiencia del usuario, por esta razón se creó el apartado reportes, el cual se detalla en la siguiente sección.

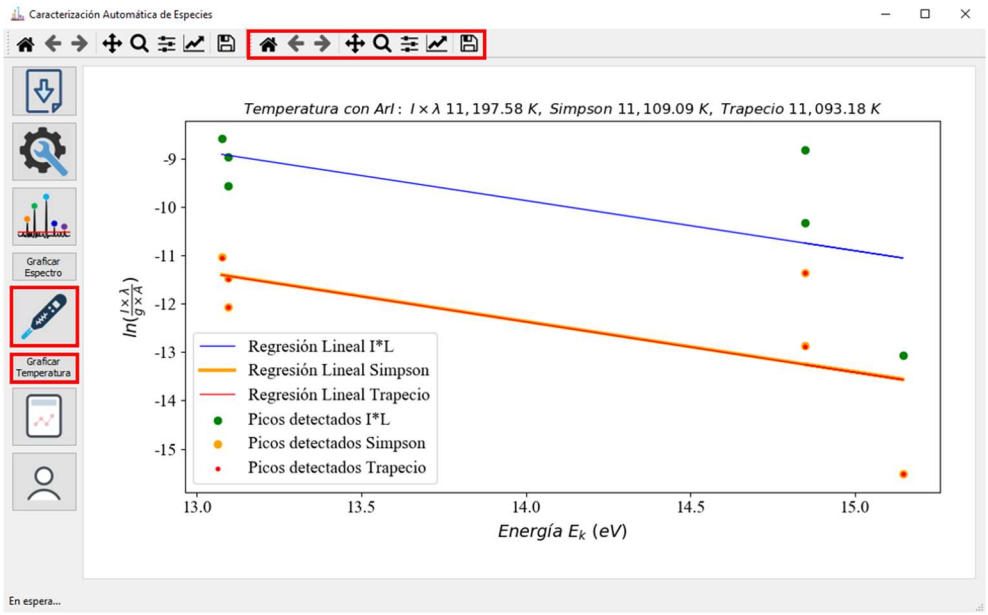


Figura 3.4.6.2. Temperatura de excitación electrónica estimada con distintos métodos.

3.4.7 Reporte de la GUI para el usuario.

Este apartado permite exportar los datos almacenados en memoria en formato **CSV** o **XLSX**. Para esto solo hay que seleccionar el formato deseado y hacer clic en el botón **Exportar** como se muestra en la Figura 3.4.7.1.

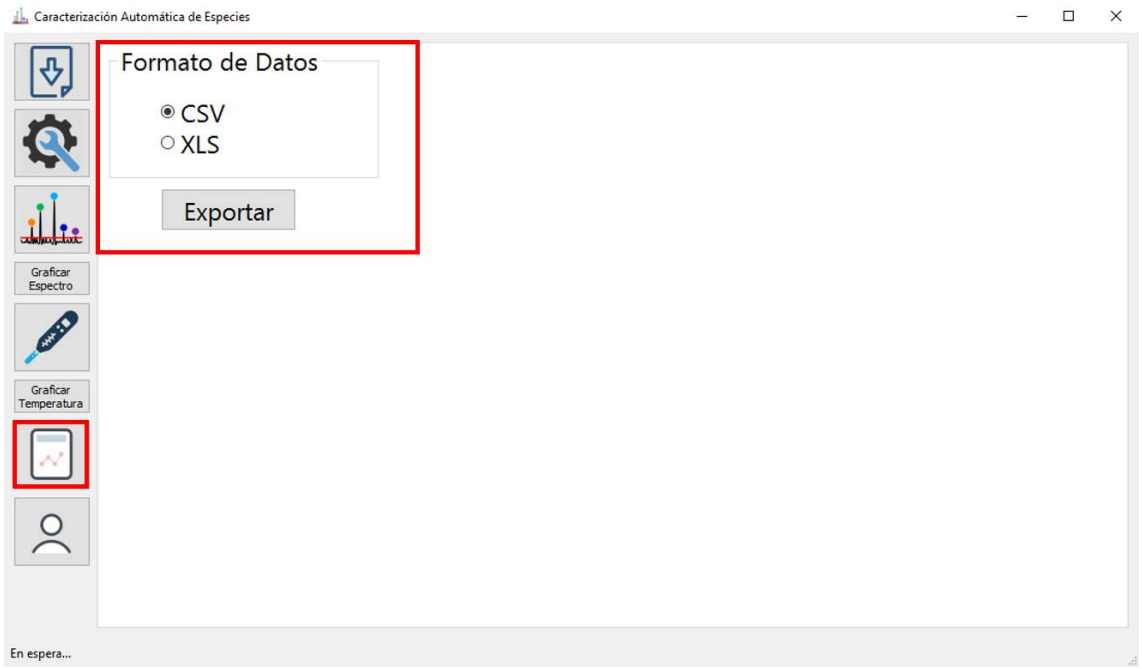


Figura 3.4.7.1. Apartado para exportar datos.

En la Figura 3.4.7.2 se muestra una sección de captura de pantalla para un archivo exportado en formato **XLSX**.

	A	B	C	D
1	picos_indice	picos_X_xp_cdo	picos_y_xp_cfc	an
2	582844	667.7428199	14.02350803	4.078
3	618679	696.5327717	162.0482578	77.82
4	628690	704.5752082	13.96858079	4.738
5	631426	706.7732743	107.7974471	51.44
6	641308	714.7120203	14.69002273	4.936
7	652560	723.7519418	12.93761819	3.657

Figura 3.4.7.2. Datos exportados en formato XLSX.

Este reporte contiene los siguientes datos de los picos detectados: a) posición, b) intensidad, c) anchura a media altura y su punto central, d) áreas, y e) primera y segunda predicción con sus probabilidades. La finalidad de este reporte es realizar cálculos manuales para verificar y/o corregir la estimación de temperatura realizada en la interfaz gráfica de usuario, ya sea por ejemplo: a) porque la corrección del desplazamiento óptico no fue suficiente (ver Figura 3.2.11.3), b) una especie se encuentre en otra región de decisión (ver Figura 4.5.1), o por características propias del espectro como paso, anchura y temperatura (ver figuras de la secciones 5.1 y 5.2).

3.4.8 Información en la Interfaz Gráfica.

El apartado de la Figura 3.4.8.1 contiene el nombre y logo de la Universidad Autónoma del Estado de México, una descripción simple de la interfaz gráfica, y el nombre del desarrollador. Esta pantalla se muestra cada vez que se inicia el software y es posible regresar a este apartado en cualquier momento.

Figura 3.4.8.1. Datos de la universidad y desarrollador.

La interfaz gráfica de usuario se diseñó de manera que su uso sea lo más sencillo y ágil posible. Los modelos de Machine Learning que la integran pasaron por un proceso de experimentación y validación que se detallan en la siguiente sección.

4 EXPERIMENTACIÓN.

En [33], [34] el diseño de experimentos están incluidas las etapas de adquisición y tratamiento de datos, selección de algoritmos y clasificación vistos en el capítulo 3, junto con las métricas, pruebas estadísticas y gráficas de esta sección. En este capítulo se tratan detalles teóricos y técnicos que complementan la experimentación realizada.

4.1 Diseño de Experimentos.

Los principios en el diseño de experimentos, con base a la literatura encontrada y de acuerdo con los datos con los que se cuenta son [12], [35]:

- **Aleatorización**: su propósito es generar variación en los grupos de tratamiento de datos para que los resultados de cada experimento sean independientes.
- **Replicación**: dada la variabilidad de los datos, es posible obtener un resultado favorable o desfavorable que sea producto de la casualidad, por esto es necesario ejecutar un algoritmo un número suficiente de veces en datos variables, y sostener los resultados en análisis estadístico. En Machine Learning esto se hace con técnicas de validación cruzada, como se detalla en la sección 4.4.
- **Blocking**: controlar la variabilidad en cada grupo de tratamiento de datos asignando un subconjunto de variables y un bloque aleatorio de datos. En Machine Learning todos los algoritmos deben usar el mismo subconjunto de datos muestreados, de lo contrario, las diferencias en las métricas dependerían no solo de los algoritmos, sino también de la aleatoriedad en los datos.
- **Estratificación**: es una técnica de muestreo donde los bloques de datos tienen clases proporcionales a un valor elegido, de esta manera cada bloque es representativo a todas las clases.

4.2 Especificaciones de Hardware y Software.

Con el fin de garantizar las condiciones de reproducibilidad de este diseño de experimentos, se indican a continuación las especificaciones del hardware y software empleados. En la Tabla 4.2.1 se describe el hardware utilizado.

Tabla 4.2.1. Especificaciones del hardware utilizado.

	Equipo 1	**Equipo 2**
Procesador	Intel i7 3770k	AMD FX 8320e
Memoria RAM	Kingston HyperX 16GB 1600MHz	Kingston HyperX 16GB 1600MHz
Almacenamiento	ADATA SSD 480 GB	ADATA SSD 240 GB
Tarjeta Madre	LENOVO MAHOBAY	GIGABYE GA-990FXA-UD3

La versión utilizada de Python en ambos equipos es la 3.7.1 con las dependencias mostradas en la Tabla 4.2.2:

Tabla 4.2.2. Especificaciones del software utilizado.

Biblioteca de Funciones	**Versión**
NumPy	1.18.0
Pandas	0.25.3
Matplotlib	3.1.2
SciPy	1.4.1
Scikit Learn	0.22.1

4.3 Métricas de Desempeño.

Con la finalidad de evaluar los algoritmos estudiados, se asume que la predicción de todas las especies de elementos es igual de importante, la métrica capaz de realizar esto es F1, y los elementos que la integran se explican a continuación.

Para determinar si la salida de un algoritmo de clasificación es correcta se verifica su salida, y las cuatro opciones posibles son:

1. Verdadero Negativo (*VN*): clase y predicción negativas.
2. Falso Positivo (*FP*, error tipo I): clase negativa y predicción positiva.
3. Falso Negativo (*FN*, error tipo II): clase positiva y predicción negativa.
4. Verdadero Positivo (*VP*): clase y predicción positivas.

Estas opciones se muestran en la salida $\boldsymbol{S}$ de la Figura 4.3.1 para cada combinación de las entradas $\boldsymbol{y}$ y $\hat{\boldsymbol{y}}$.

$\boldsymbol{y}$	$\hat{\boldsymbol{y}}$	$\boldsymbol{S}$
0	0	*VN*
0	1	*FP*
1	0	*FN*
1	1	*VP*

$\boldsymbol{y}$	$\hat{\boldsymbol{y}}$	$\boldsymbol{S}$
~ Ar I	~ Ar I	*VN*
~ Ar I	Ar I	*FP*
Ar I	~ Ar I	*FN*
Ar I	Ar I	*VP*

Figura 4.3.1. Opciones de clasificación de un valor predicho respecto al valor real.

Dónde:

0, ~Ar I	No es Ar I.
1, Ar	Es Ar I.
$\boldsymbol{y}$	Valores de salida reales en un vector con la forma (*n_muestras*).
$\hat{\boldsymbol{y}}$	Valores de salida predichos en un vector con la forma $(n\_muestras)$.
$\boldsymbol{S}$	Tipo de salida en un vector con la forma (*n_muestras*).

Para visualizar $\boldsymbol{S}$ se utiliza una matriz bidimensional nombrada matriz de confusión, que compara $\hat{\boldsymbol{y}}$ respecto a $\boldsymbol{y}$, como la que se observa en Figura 4.3.2.

		$\boldsymbol{y}$		
		Ar I	~ Ar I	Total
$\hat{\boldsymbol{y}}$	Ar I	*VP*	*FP*	$\boldsymbol{p'}$
	~ Ar I	*FN*	*VN*	$\boldsymbol{n'}$
	Total	$\boldsymbol{p}$	$\boldsymbol{n}$	$\boldsymbol{N}$

Figura 4.3.2. Matriz de confusión para una clasificación binaria.

La suma total de las filas y columnas se define como:

$$\boldsymbol{p} = VP + \sum FN \tag{4.1}$$

$$\boldsymbol{n} = \sum VN + \sum FP \tag{4.2}$$

$$\boldsymbol{p'} = VP + \sum FP \tag{4.3}$$

$$\boldsymbol{n'} = \sum VN + \sum FN \tag{4.4}$$

Dónde:

$\boldsymbol{p}$	Valores reales positivos de la clase actual.
$\boldsymbol{n}$	Valores reales negativos de la clase actual.
$\boldsymbol{p'}$	Valores predichos positivos de la clase actual.
$\boldsymbol{n'}$	Valores predichos negativos de la clase actual.
$\boldsymbol{N}$	Es la suma total de la matriz de confusión.

El problema de clasificación de este trabajo es multiclase, en la Figura 4.3.3 se encuentra un ejemplo de matriz de confusión para una clasificación con tres clases.

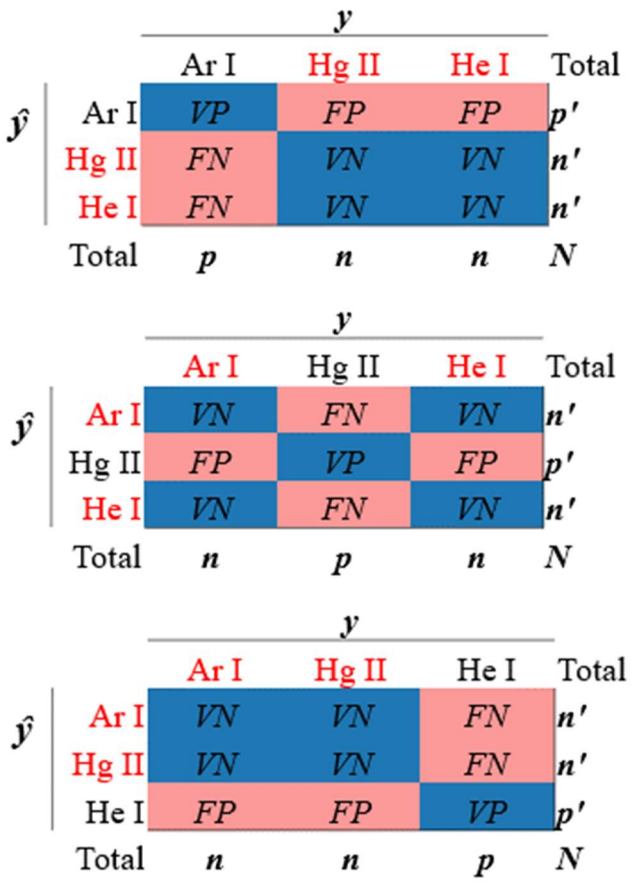


Figura 4.3.3. Matriz de confusión para una clasificación con 3 clases.

De manera análoga a lo mostrado en Figura 4.3.3, es posible extender una matriz de confusión a problemas de clasificación dónde el número de clases es mayor o igual a 2.

A partir de una matriz de confusión se derivan un conjunto de ecuaciones que se resumen en la Tabla 4.3.1.

Tabla 4.3.1. Ecuaciones derivadas a partir de la matriz de confusión [12], [35].

Nombre	Fórmula Binaria	Fórmula Multiclase	Descripción
Accuracy	$= \frac{VP + VN}{\boldsymbol{N}}$ $= 1 - Error$	$= \frac{VP + \sum VN}{\boldsymbol{N}}$ $= 1 - Error$	Porcentaje de clasificaciones correctas.
Error	$= \frac{FP + FN}{\boldsymbol{N}}$ $= 1 - Accuracy$	$= \frac{\sum FP + \sum FN}{\boldsymbol{N}}$ $= \frac{(\boldsymbol{p}' - VP) + (\boldsymbol{p} - VP)}{\boldsymbol{N}}$ $= 1 - Accuracy$	Porcentaje de clasificaciones incorrectas.
VP-rate *Recall* *Sensitivity*	$= \frac{VP}{\boldsymbol{p}}$	$= \frac{VP}{\boldsymbol{p}}$	Porcentaje de la clase actual clasificada correctamente. Si la clase es Ar I, ¿Con qué frecuencia predice Ar I?
FP-rate	$= \frac{FP}{\boldsymbol{n}}$ $= 1 - Specificity$	$= \frac{\sum FP}{\sum \boldsymbol{n}}$ $= \frac{p' - VP}{\sum \boldsymbol{n}}$ $= 1 - Specificity$	Porcentaje de FP respecto a la clase actual. Si la clase no es Ar I, ¿Con qué frecuencia predice Ar I?
Precision	$= \frac{VP}{\boldsymbol{p}'}$	$= \frac{VP}{\boldsymbol{p}'}$	Porcentaje de la clase actual predicha correctamente. Si predice Ar I, ¿Con qué frecuencia acierta?
Specificity	$= \frac{VN}{\boldsymbol{n}}$ $= 1 - FPrate$	$= \frac{\sum VN}{\sum \boldsymbol{n}}$ $= 1 - FPrate$	Porcentaje de las otras clases clasificadas correctamente. Si la clase no es Ar I, ¿Qué tan frecuente predice no es Ar I?

A partir de las ecuaciones de la Tabla 4.3.1 se define la métrica F1 (4.5) como la media armónica de *Precision* y *Recall* así:

$$F1 = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.5}$$

Debido a que se trata de un problema multiclase la ecuación final es (4.6)

$$\overline{F1} = \frac{1}{c} \sum_{i=1}^{c} F1_i \tag{4.6}$$

Dónde:

$\overline{F1}$ F1 promedio.

c Número total de clases.

4.4 Configuración del Experimento.

Encontrar la mejor combinación de parámetros para un algoritmo específico, es una de las tareas que más tiempo consume cuando se explora un espacio de posibilidades específico. Dentro de las de validación cruzada existe la *validación cruzada repetida estratificada*, es una técnica que abarca todos los principios del diseño de experimentos tratados en la sección 4.1. En la Tabla 4.4.1 se muestran las opciones de la configuración utilizada para la validación cruzada repetida estratificadas.

Tabla 4.4.1. Opciones para la validación cruzada repetida estratificada.

Parámetro	Opciones	Descripción
n_splits	[2,3,5]	Cantidad de bloques utilizados en la validación cruzada.
n_repeats	[100]	Cantidad de veces que se repite la validación cruzada.
random_state	2020	Semilla utilizada para generar el estado aleatorio.

Para el supuesto caso de un problema con 3 clases y **n_splits=3**, se tendría un caso como el mostrado en la Figura 4.4.1 dónde en cada iteración de la validación cruzada se tienen grupos de datos para entrenamiento y prueba proporcionales a **n_splits**. Esto se realiza en cada repetición indicada en **n_repeteats**, y los grupos de datos se generan aleatoriamente a partir del valor colocado en **random_state**.

Figura 4.4.1. Gráfica de ejemplo para una división de datos estratificada.

El costo computacional se multiplica por el valor de **n_repeats** para cada posible combinación por cada algoritmo del espacio de hiperparámetros mostrado en la sección 3.2.12. En la Tabla 4.4.2 se resumen las combinaciones de cada espacio de hipeparámetros con el tiempo total alcanzado por cada algoritmo.

Tabla 4.4.2. Tiempos estimados por algoritmo y configuración.

Algoritmo	Espacio de Hiperparámetros	n_splits	n_repeats	Combinaciones Totales	Horas	Días
Decision Tree	17,820	2	100	3,564,000	1.426	0.059
		3		5,346,000	2.116	0.088
		5		8,910,000	3.380	0.141
Bagging	27,664	2	100	5,532,800	2.525	0.105
		3		8,299,200	2.943	0.123
		5		13,832,000	5.738	0.239
Random Forest	3,243,240	2	100	648,648,000	260.825	10.868
		3		972,972,000	360.487	15.020
		5		1,621,620,000	610.025	25.418
Extremely Randomized Trees	3,243,240	2	100	648,648,000	234.276	9.762
		3		972,972,000	375.983	15.666
		5		1,621,620,000	616.721	**25.967**

La mejor combinación de parámetros por clasificador se observa en la Tabla 4.4.3

Tabla 4.4.3. Hiperparámetros encontrados para cada clasificador.

Clasificador	Parámetros
Decision Tree	max_features=None criterion='entropy' max_depth=17

Clasificador	Parámetros
	min_samples_split=2
	min_samples_leaf=1
Bagging	base_estimator=*Decision Tree*
	max_samples=0.88
	n_estimators=100
	bootstrap=True
	bootstrap_features=False
Random Forest	criterion='entropy'
	max_depth=14
	max_features='log2'
	min_samples_split=2
	min_samples_leaf=1
	n_estimators=100
	bootstrap=True
Extremely Randomized Trees	criterion='entropy'
	max_depth=16
	max_features='auto'
	min_samples_split=2
	min_samples_leaf=1
	n_estimators=100
	bootstrap=False

Con base a estos resultados se asignaron estos hiperparámetros a su respectivo algoritmo y se probaron nuevamente con los datos de entrenamiento en una validación cruzada de 2, 3, 5 y 10, cada una repetida 100 veces con/sin estratificar, estos datos se muestran en Figura 4.4.2 con gráficas de caja y bigotes.

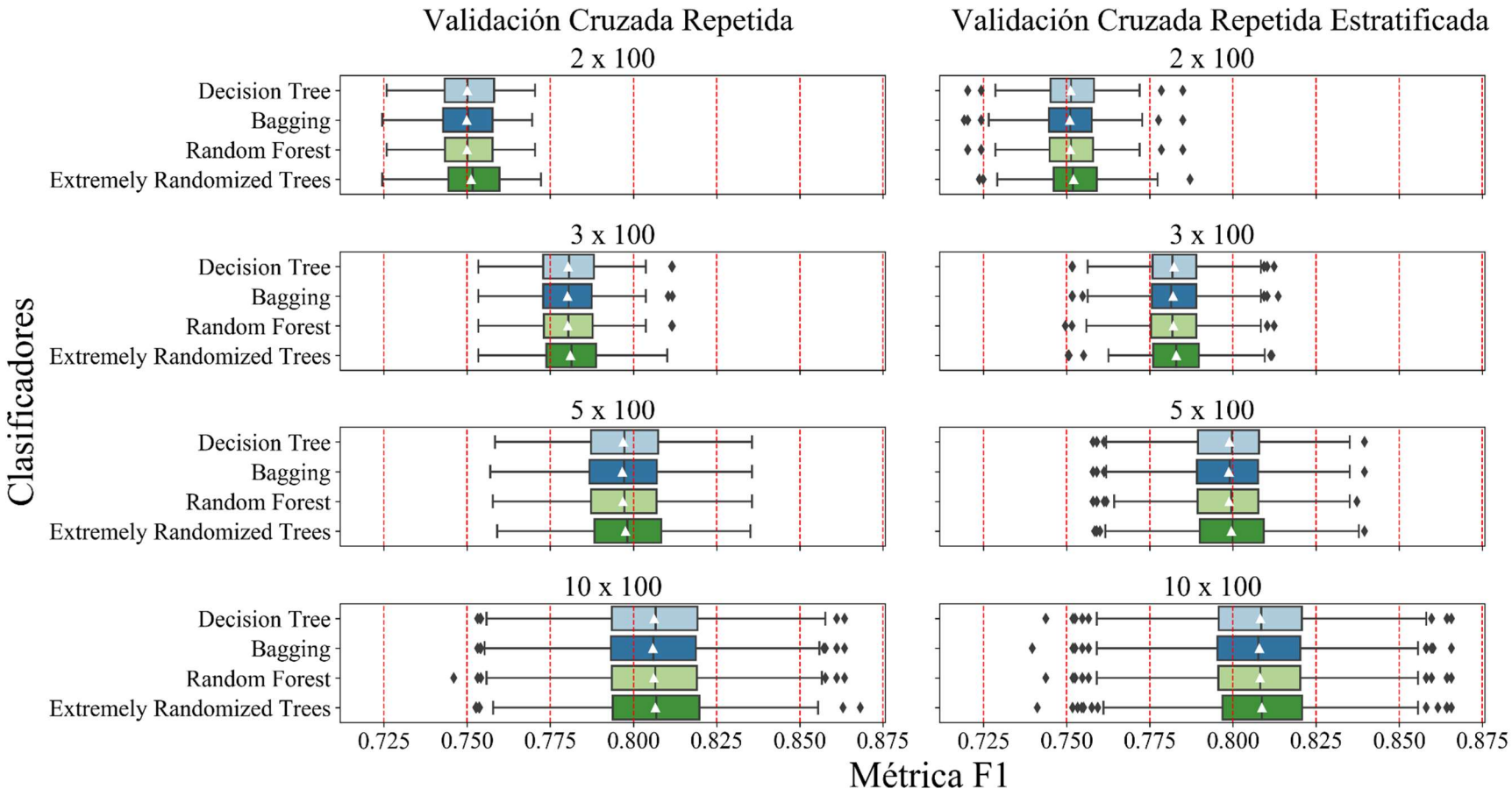


Figura 4.4.2. Grafica de caja y bigotes con validación cruzada repetida y validación cruzada repetida estratificada para los cuatro algoritmos estudiados.

En Figura 4.4.2 se muestra que Extremely Randomized Trees es ligeramente superior en todas las pruebas, mientras que los otros tres algoritmos tienen un rendimiento en apariencia similar. También se puede apreciar que la línea vertical dentro de cada caja (mediana) y el triángulo (media) se superponen, por esta razón se realizó la prueba paramétrica de ANOVA que verifica si las medias de las distribuciones en los modelos son la misma (H0), además de la prueba no paramétrica de Friedman, esta verifica si las medianas de las distribuciones en los modelos son iguales (H0), estos resultados se integran en la Tabla 4.4.4 para un nivel de significancia $\alpha = 0.05$.

Tabla 4.4.4. Comparativa de p-valores en H0 para las pruebas de Friedman y ANOVA.

Experimento	**Friedman p-valor**	**Friedman H0**	**ANOVA p-valor**	**ANOVA H0**
vcr: 2x100	1.85023×10^{-70}	False	0	False
vcr: 3x100	3.43449×10^{-104}	False	0	False
vcr: 5x100	8.22954×10^{-174}	False	0	False
vcr: 10x100	0	False	0	False
vcre: 2x100	1.35025×10^{-69}	False	0	False
vcre: 3x100	5.49523×10^{-104}	False	0	False
vcre: 5x100	7.32834×10^{-173}	False	0	False
vcre: 10x100	0	False	0	False

Se encontró que en ambas pruebas se rechaza H0, es decir, una o más distribuciones en cada caso de validación son diferentes, por lo que el ensamblado de algoritmos se realiza con los cuatro modelos vistos en la Tabla 4.4.3. Con la finalidad de mostrar diferencias sutiles en todos los algoritmos se aplicó la prueba de Nemenyi (Figura 4.4.3) con una significancia $\alpha = 0.05$. En esta prueba se obtiene la posición de predicción para cada muestra y se calcula la distancia crítica (CD por sus siglas en inglés); aquellos modelos que se encuentran dentro del rango de CD se consideran equivalentes en predicción, esto no implica que cada estimador cometa el mismo error de predicción con la misma observación.

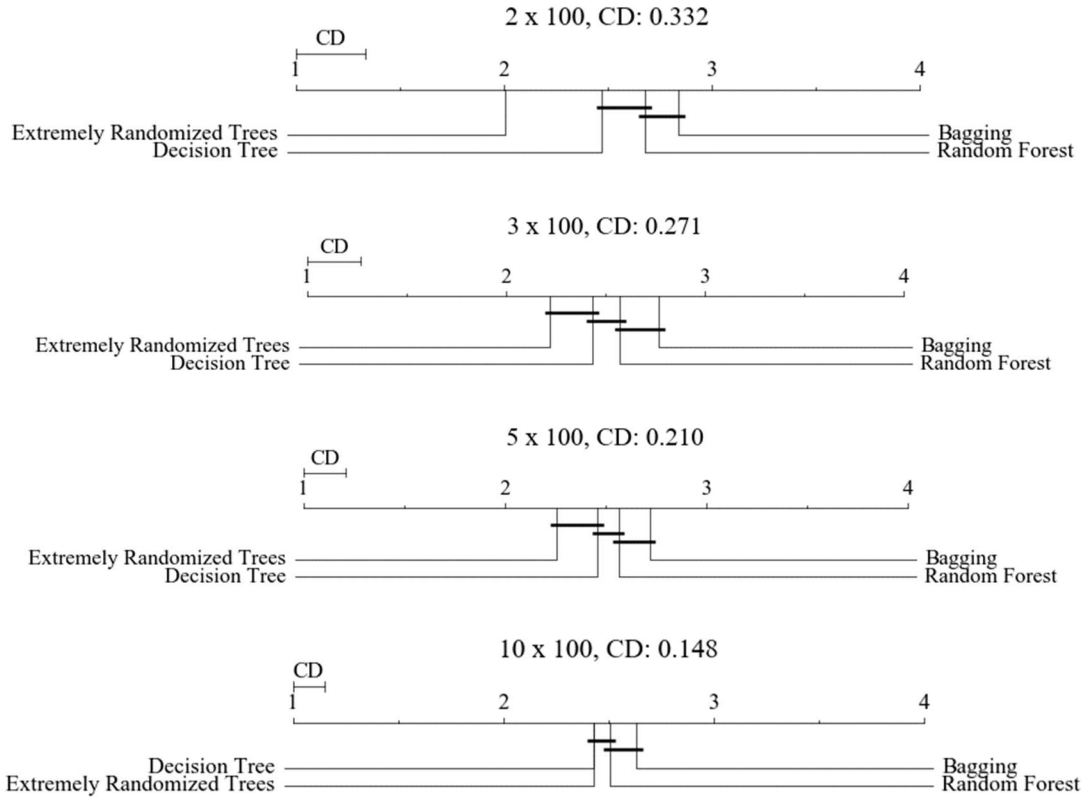


Figura 4.4.3. Prueba de Nemenyi en cuatro casos de validación cruzada.

Se observa en Figura 4.4.3 que conforme aumenta el número de divisiones en la validación cruzada repetida, se reduce el valor de la distancia crítica. También es una constante que Extremely Randomized Trees y Decision Trees aparecen como los mejores algoritmos para cada caso, y que el resto de los modelos tienen pequeñas diferencias entre sí, por lo que al ensamblarlos las predicciones tendrán una métrica F1 mayor.

Al momento ensamblar los algoritmos, cada uno con sus hiperparámetros, se visualiza su efecto en los datos del NIST con entrenamiento y validación cruzada cuando se varía el número de muestras y se mide el rendimiento con el uso de la métrica F1. Así se determina la cantidad de muestras para entrenar el modelo final y si el modelo se ajusta correctamente a los datos. En la Figura 4.4.4 se observa una curva de aprendizaje con el modelo ensamblado por votación suave con los datos del NIST, en esta la curva de validación cruzada tiende a converger con la curva de entrenamiento, sin embargo nunca lo hacen, para esto se requieren más datos, lo que sugiere que deben emplearse todos los datos para entrenar el modelo final. Finalmente, el color verde claro que acompaña la

curva de validación cruzada corresponde a la desviación estándar, esta tiende a decrecer conforme se agregan más datos, como se ve en los dos últimos gráficos. Esto indica que el modelo final predice correctamente los datos con los que es entrenado, y que se requieren todos los datos para integrar el modelo final.

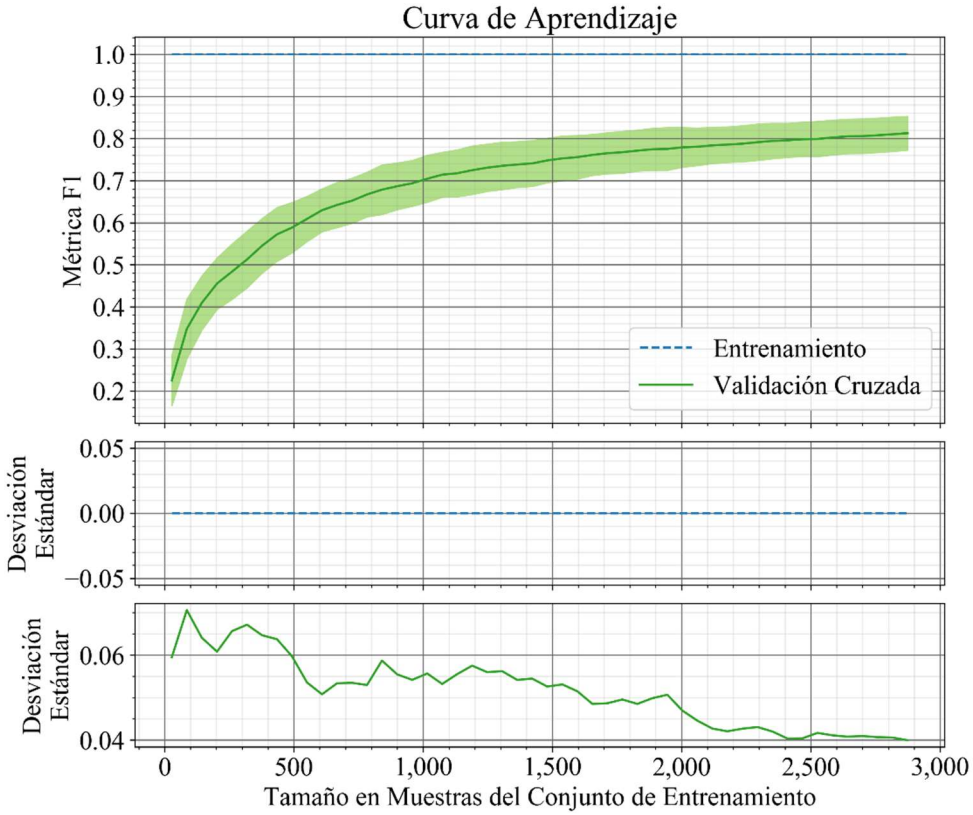


Figura 4.4.4. Curva de aprendizaje para el modelo ensamblado y entrenado con los datos del NIST.

Después de crear el modelo ensamblado, se crearon curvas ROC (Receiver Operating Characteristic, por sus siglas en inglés), estas comparan la tasa de *VP* contra la tasa de *FP*, cuanto más próxima se encuentre la curva a la esquina superior derecha, mejor es el clasificador. Cada gráfica se puede reducir a un valor AUC (Area Under Curve, por sus siglas en inglés). El clasificador perfecto tiene como área una unidad cuadrada, por lo tanto, mientras más se acerque un clasificador a este valor, mejor es su tasa de *VP* [12].

Se crearon curvas ROC con validación cruzada para observar la tendencia de las tasas de *VP* y *FP* cuando se entrena el modelo por votación con una fracción de los datos del NIST y se usa el resto para probarlo, en la Figura 4.4.5 se aprecia que la menor área es para O

II con 0.82, el área máxima es para O I con 0.99, mientras que el área bajo la curva para la métrica F1 es de 0.94.

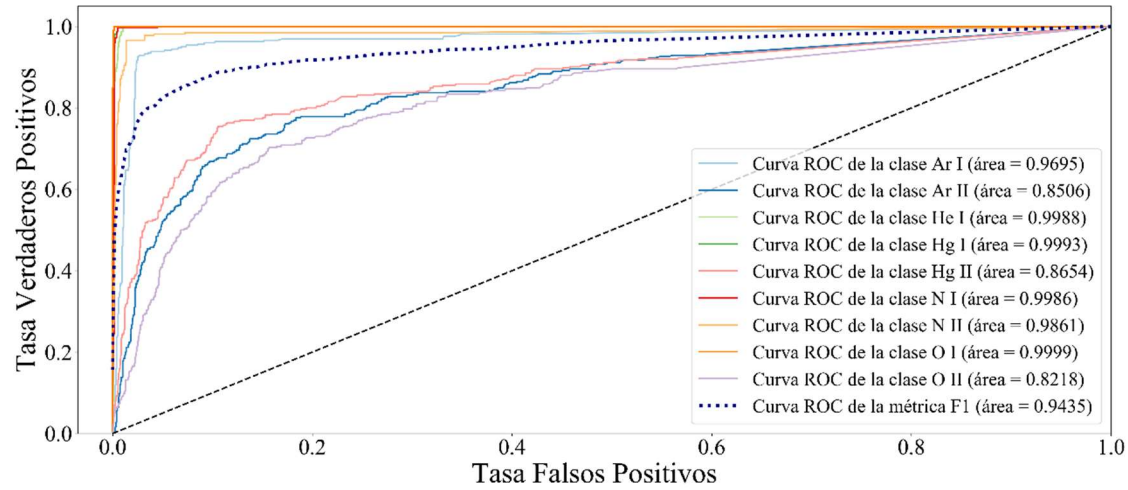


Figura 4.4.5. Curvas ROC con validación cruzada para el modelo por votación.

En el acercamiento mostrado en Figura 4.4.6, se aprecia con dificultad a causa de la superposición de curvas ROC que las clases He I, Hg I, N I y O I tienen una tasa de *VP* próxima a 1.0, esto se puede confirmar al buscar en la leyenda del gráfico el área de estas clases. La curva ROC que representa a todas las clases es F1 con un área de 0.94, y al usarla como referencia, se observa que las clases Ar II, Hg II y O II tienen un área menor.

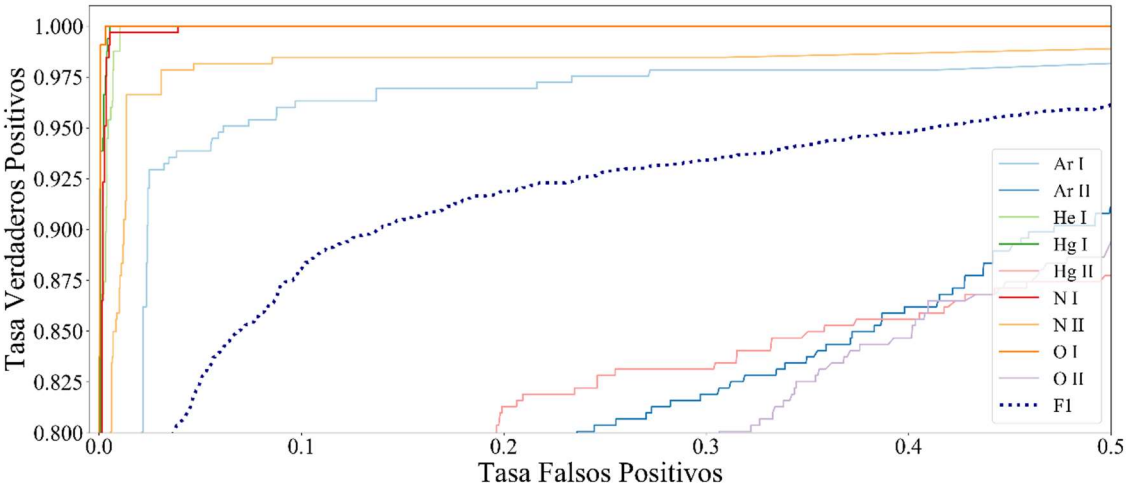


Figura 4.4.6. Acercamiento en la esquina superior derecha en curvas ROC con validación cruzada para el modelo por votación

4.5 Fronteras de Decisión.

Una gráfica de frontera de decisión permite visualizar la forma en que un modelo separa las clases respecto a un par de variables de entrada en función de los datos de entrenamiento. En la Figura 4.5.1 se muestran las fronteras de decisión para el modelo por votación, el eje X corresponde a la longitud de onda y el eje Y al tipo, donde 0 es vacío y 1 con aire; el acercamiento a) muestra las fronteras de todo el modelo entrenado, los siguiente tres acercamientos b), c) y d) corresponden a diferentes secciones a partir de los 200 nm con saltos de 300 nm y un ancho de 4 nm para apreciar las fronteras y las especies contenidas en intervalos simétricos. Cabe mencionar que para poder apreciar las fronteras de decisión la Figura 4.5.1 se genera un objeto de imagen con una dimensión de 1000 in de largo por 2.2 in de alto, esto ocupa en memoria RAM alrededor de 30 GB, al ser una figura extensa, el criterio por el que se eligen esos intervalos es mostrar regiones de los extremos y el centro. Cada elemento se representa con una forma particular, y para saber si una clasificación es correcta, el color de cada forma debe corresponder con el color de su región de decisión.

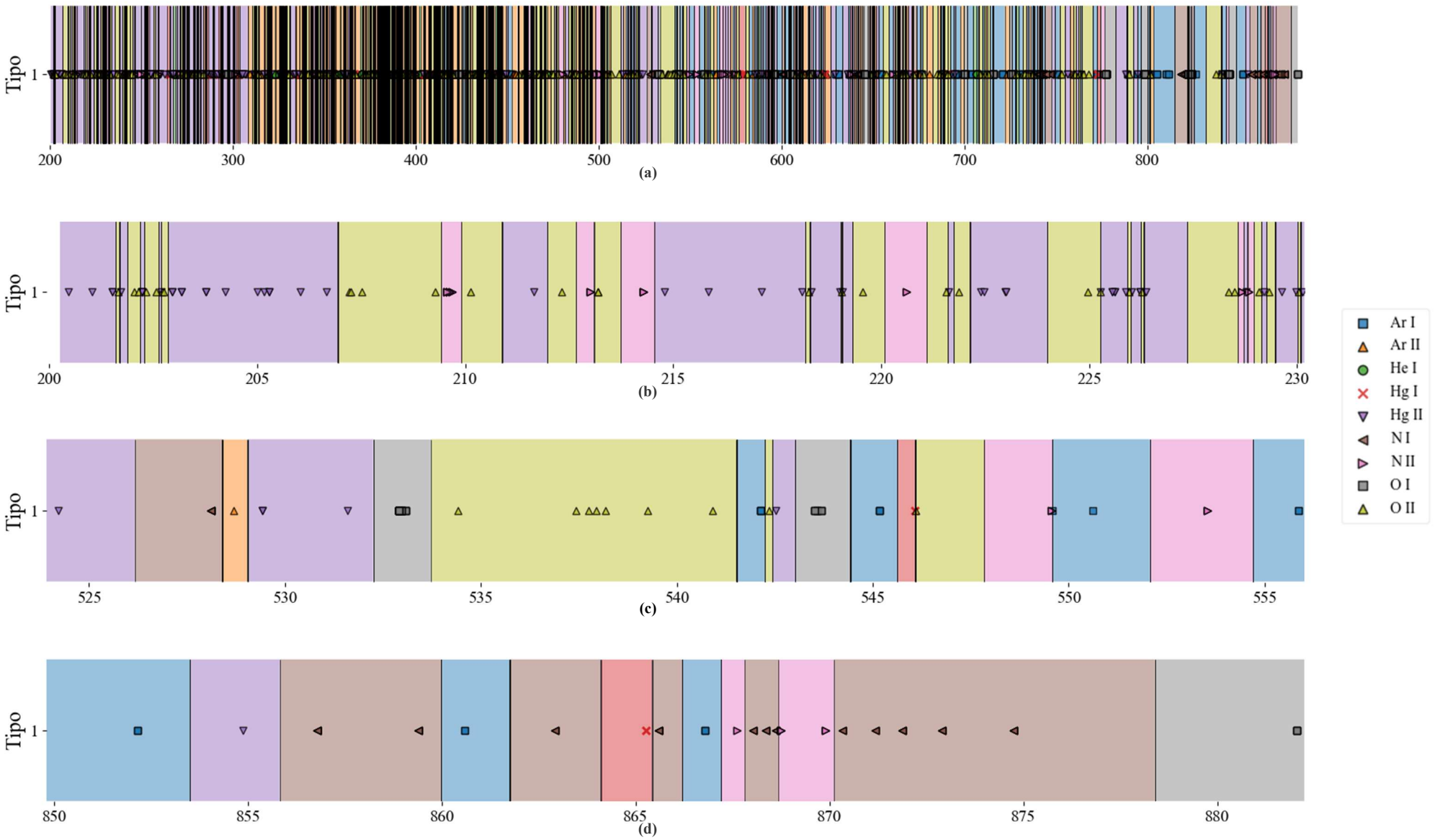


Figura 4.5.1. Fronteras de decisión para el modelo ensamblado por votación en un rango de longitud de onda para (a) 200 nm a 890 nm, (b) 200 nm a 230 nm, (c) 524 nm a 556 nm, y (d) 850 nm a 883 nm.

En la Figura 4.5.1 en (c) a una longitud de onda de 546 nm una especie de Hg I 546.0750 nm se encuentra próxima a una especie de O II 546.1040 nm, y con el nivel de zoom mostrado parecen superponerse. En caso de solapamiento en una frontera de decisión se tiene preferencia por la clase de la región de decisión que se encuentra a la izquierda, y si el solapamiento ocurre sobre una región de decisión se asigna su clase correspondiente.

Se destaca que la corrección del desplazamiento óptico y la resolución a la que fue capturado o generado el espectro determinan la posición en la que cada pico se corresponde con su frontera y por tanto elemento de predicción. Para observar el rendimiento del modelo ensamblado por votación se utilizan datos de validación que hasta el momento el algoritmo no ha usado. Este procedimiento se detalla en la siguiente sección.

5 RESULTADOS Y DISCUSIÓN

Dada la contingencia sanitaria presentada en el año en curso 2020, existen limitaciones para regresar a las actividades laborales en toda la república mexicana basadas en un semáforo COVID-19 de cuatro colores (rojo, naranja, amarillo y verde) y cuando una región se encuentra en los colores amarillo y verde se permiten todas las actividades laborales pero con precaución, hasta el mes de Agosto el semáforo COVID-19 se mantiene en color naranja por lo que no se pueden recolectar datos experimentales de las especies de estudio en el Laboratorio de Física de Plasmas del ININ, y se plantea como alternativa de origen de datos utilizar el generador de espectros sintéticos, siendo este una mejora y refinamiento al trabajo de [32], con la característica adicional de generar espectros en condiciones variables de ***Paso*** (espacio en nanómetros entre cada par de puntos adyacentes de longitud de onda), ***Anchura*** (anchura a media altura) y ***Temperatura*** (en grados Kelvin), y se realizan dos análisis, uno general en que se propone un conjunto de opciones basadas en el conocimiento empírico obtenido hasta el momento, y un análisis específico basado en las características de los espectros de la lámpara de calibración HG-1 utilizada en el Laboratorio de Física de Plasmas del ININ.

5.1 Análisis General.

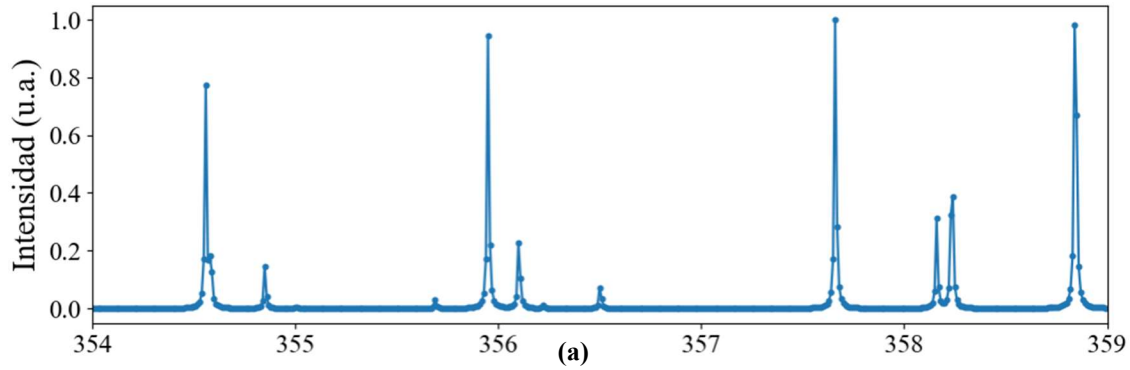
El procedimiento para generar espectros sintéticos se detalla en la sección 3.1.3, en esta sección se muestra el procedimiento para generar espectros sintéticos que servirán como datos de validación. A continuación, se trata la cantidad de espectros sintéticos generados y la nomenclatura usada.

Por cada clase de la Figura 3.2.2.1 se generaron 108 espectros sintéticos dando un total de 972, los cuales son resultado de alguna de las combinaciones mostradas en la Tabla 5.1.1 en el supuesto caso de ser observadas con aire en el rango de longitud de onda de los 200 nm a 890 nm. Eso implica que por cada una de las 9 clases se calculan 6 pasos, por cada paso 6 anchuras y por cada anchura 3 temperaturas.

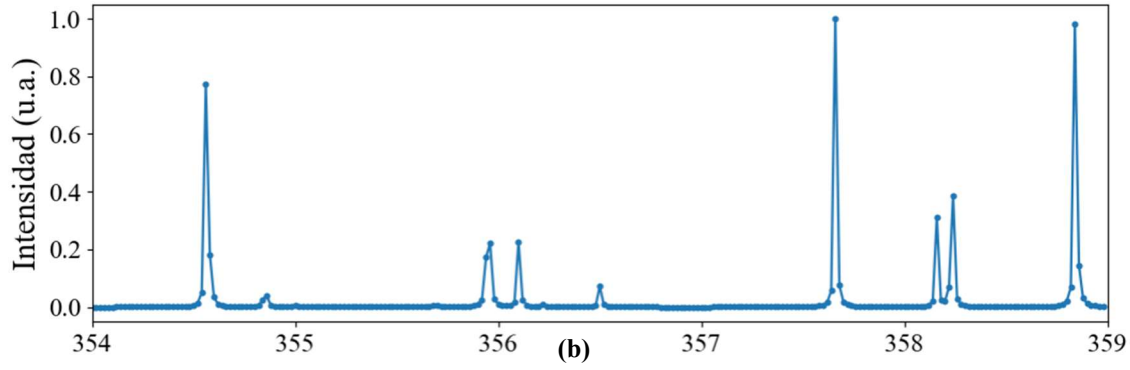
Tabla 5.1.1. Opciones de variación para los espectros sintéticos.

Atributo	Opciones	Cantidad
Clase	Ar I, Ar II, He I, Hg I, Hg II, N I, N II, O I, O II	9
Paso	0.01, 0.02. 0.03, 0.05, 0.1, 0.2	6
Anchura	0.01, 0.03, 0.05, 0.1, 0.3, 0.5	6
Temperatura	1,000K, 10,000K, 20,000K	3

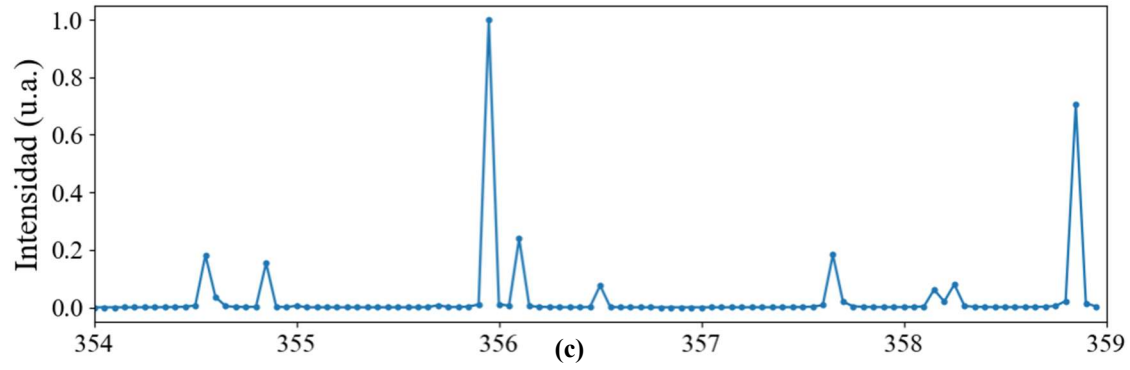
El ***Paso*** es la variable λ utilizada en la ecuación (3.2) y corresponde a la distancia en nanómetros entre cada punto en longitud de onda del espectro sintético, entre mayor es este valor, menor es la resolución del espectro y aparecen desplazamientos y deformaciones en los picos, como se aprecia en la Figura 5.1.1



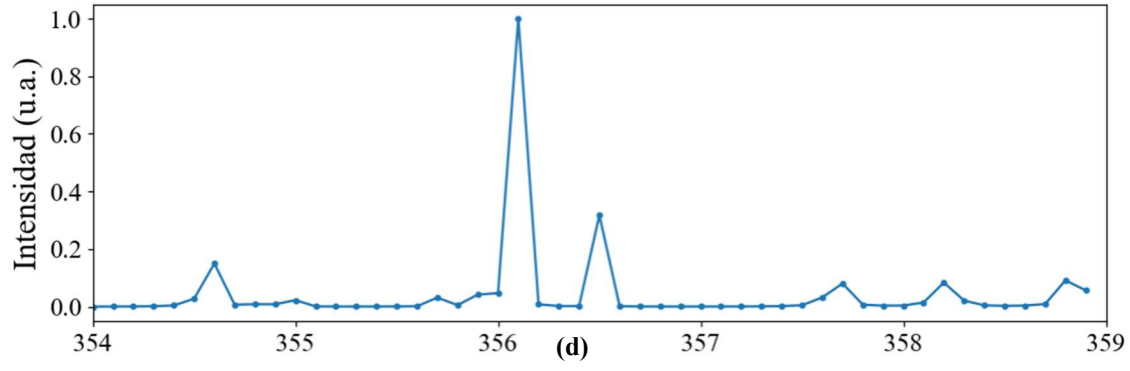
(a)



(b)

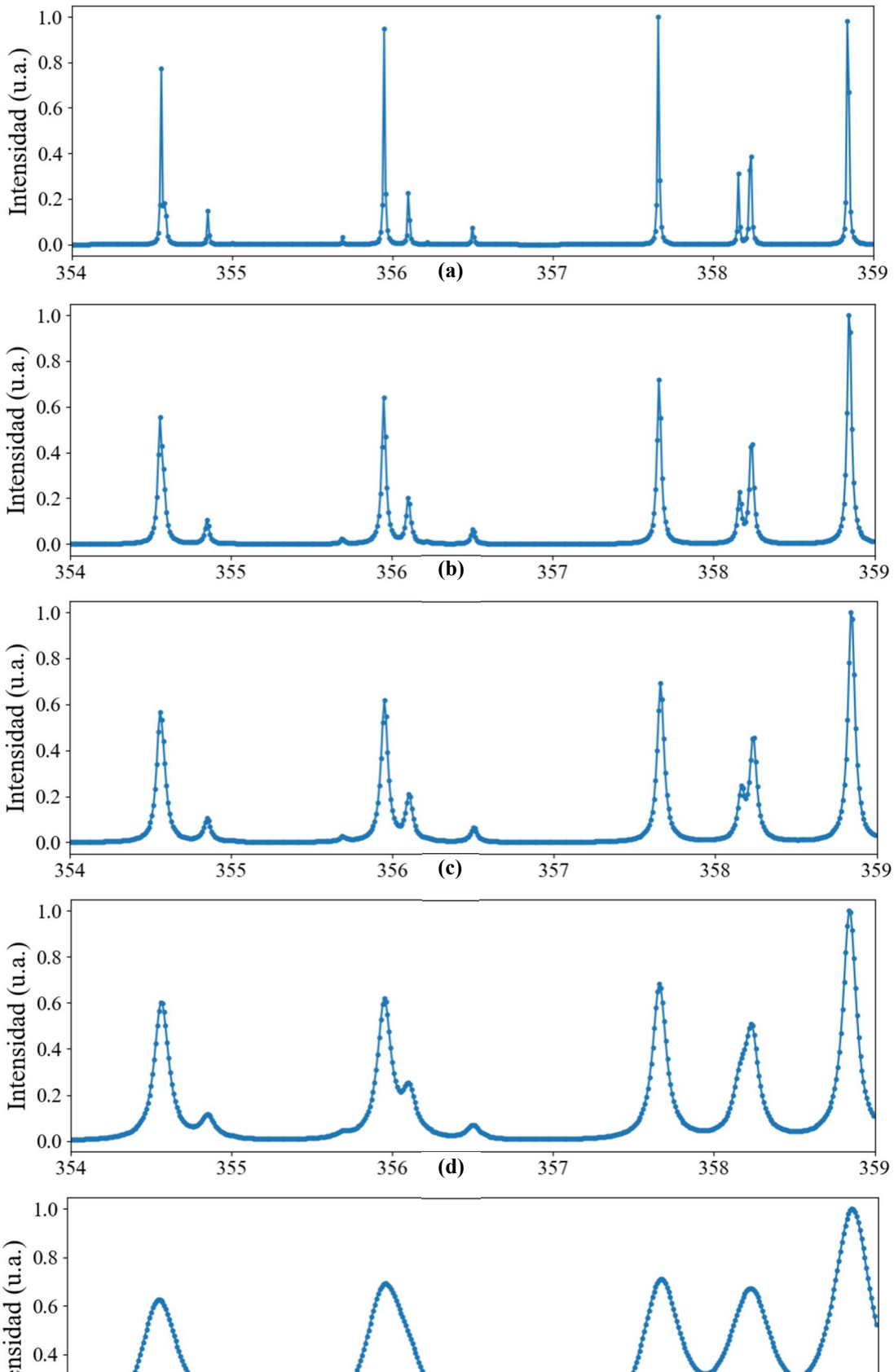


(c)

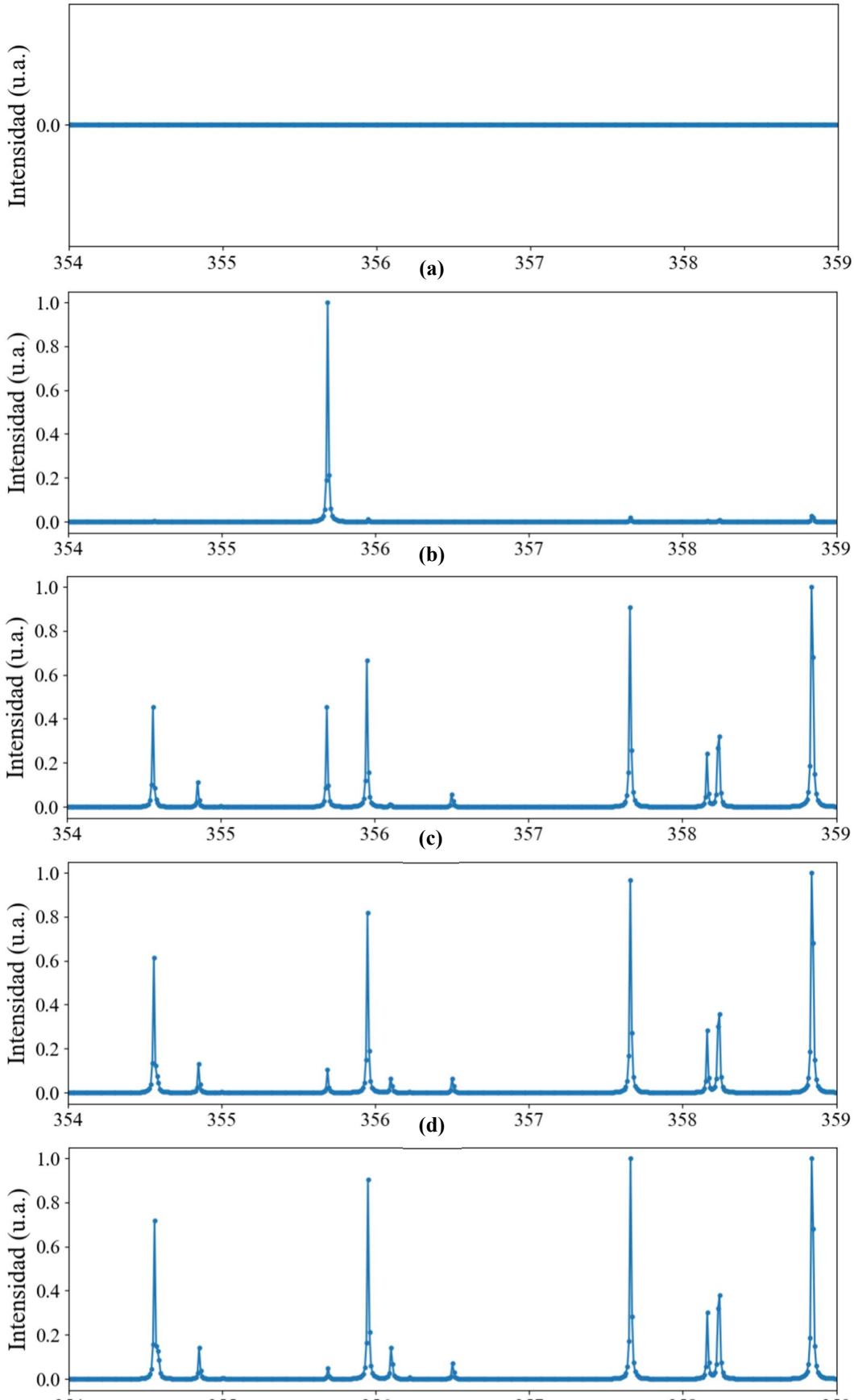


(d)

La ***Anchura*** es la variable w utilizada en la ecuación (3.2) y define la anchura a media altura de cada pico, valores pequeños permiten la observación de picos contiguos, mientras que valores grandes suman y distorsionan picos, el efecto de esta variable se observa en la Figura 5.1.2.



La ***Temperatura*** es la variable T de la ecuación (3.2) y define la altura de cada pico, conforme se incrementa este valor se aprecia la aparición y variación de picos como se observa en la Figura 5.1.3.



Cabe mencionar que cada archivo de espectro sintético generado se nombró con la nomenclatura **Elemento_NivelDeEnergía_Tipo_Anchura_Rango_Temperatura**, agrupados en directorios de acuerdo con el tamaño del ***Paso***, en la Figura 5.1.4 se observan 162 espectros sintéticos agrupados en un directorio por ***Paso*** de 0.01.

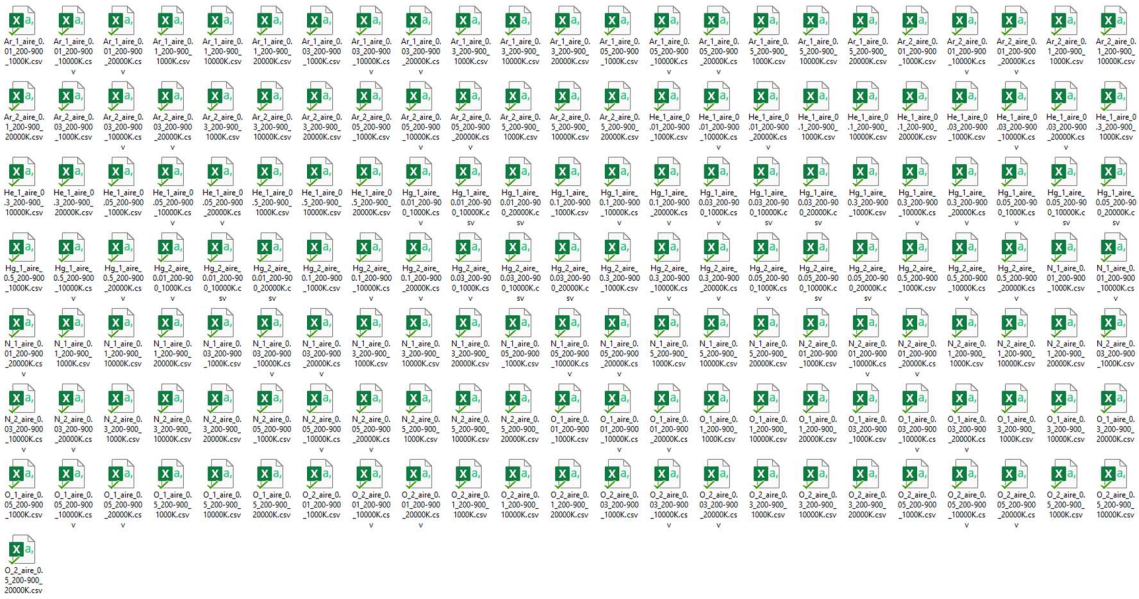


*Figura 5.1.4. 162 archivos de espectros sintéticos generados para un **Paso** de 0.01.*

La finalidad de estos espectros sintéticos es evaluar la exactitud (se utiliza *accuracy* mostrado en Tabla 4.3.1) del modelo final generado con los mejores hiperparámetros encontrados. Los resultados de este experimento con diferentes condiciones de ***Paso***, ***Anchura*** y ***Temperatura*** se muestran en las Figura 5.1.5 a Figura 5.1.10.

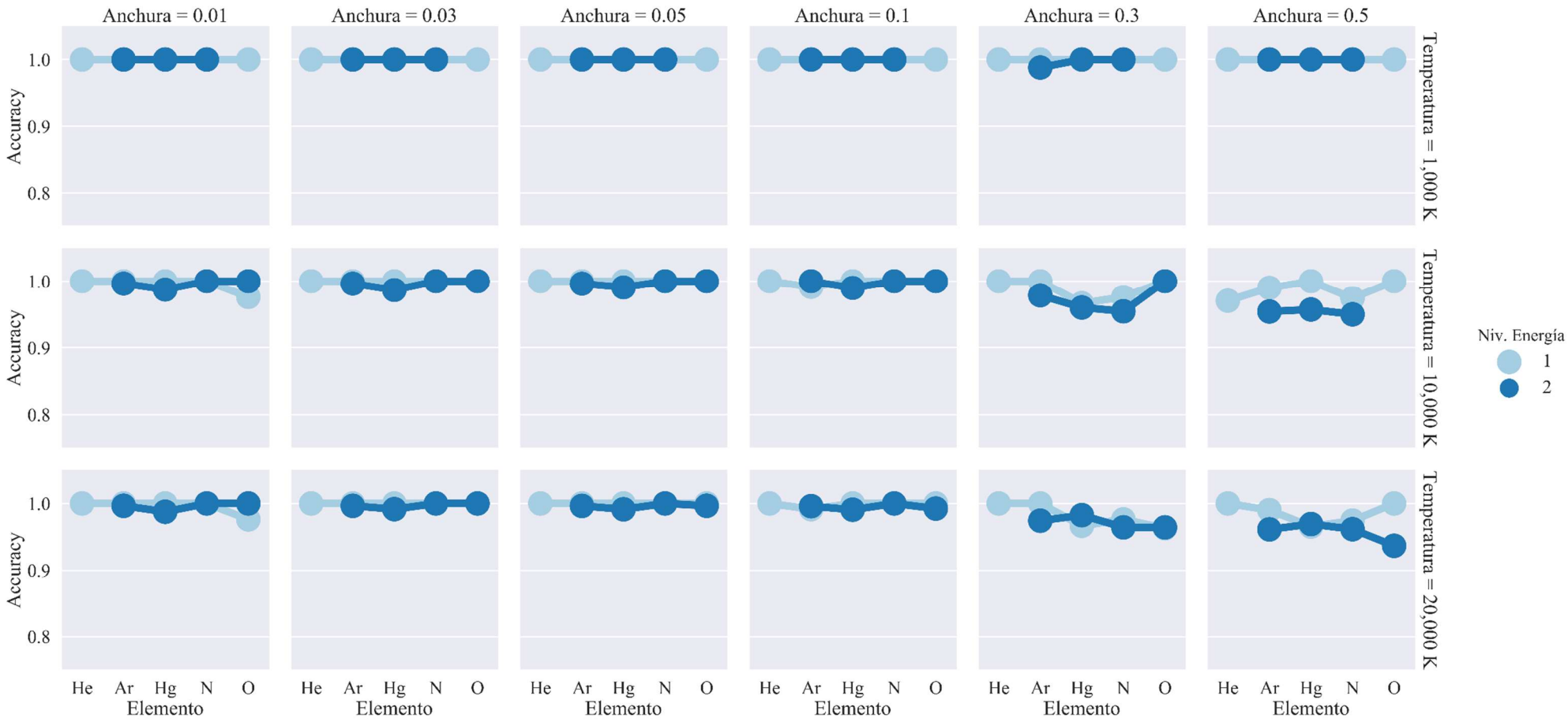


*Figura 5.1.5. Exactitud en las predicciones del modelo ensamblado con un **Paso** de 0.01 nm en distintas condiciones de **Anchura** y **Temperatura**.*

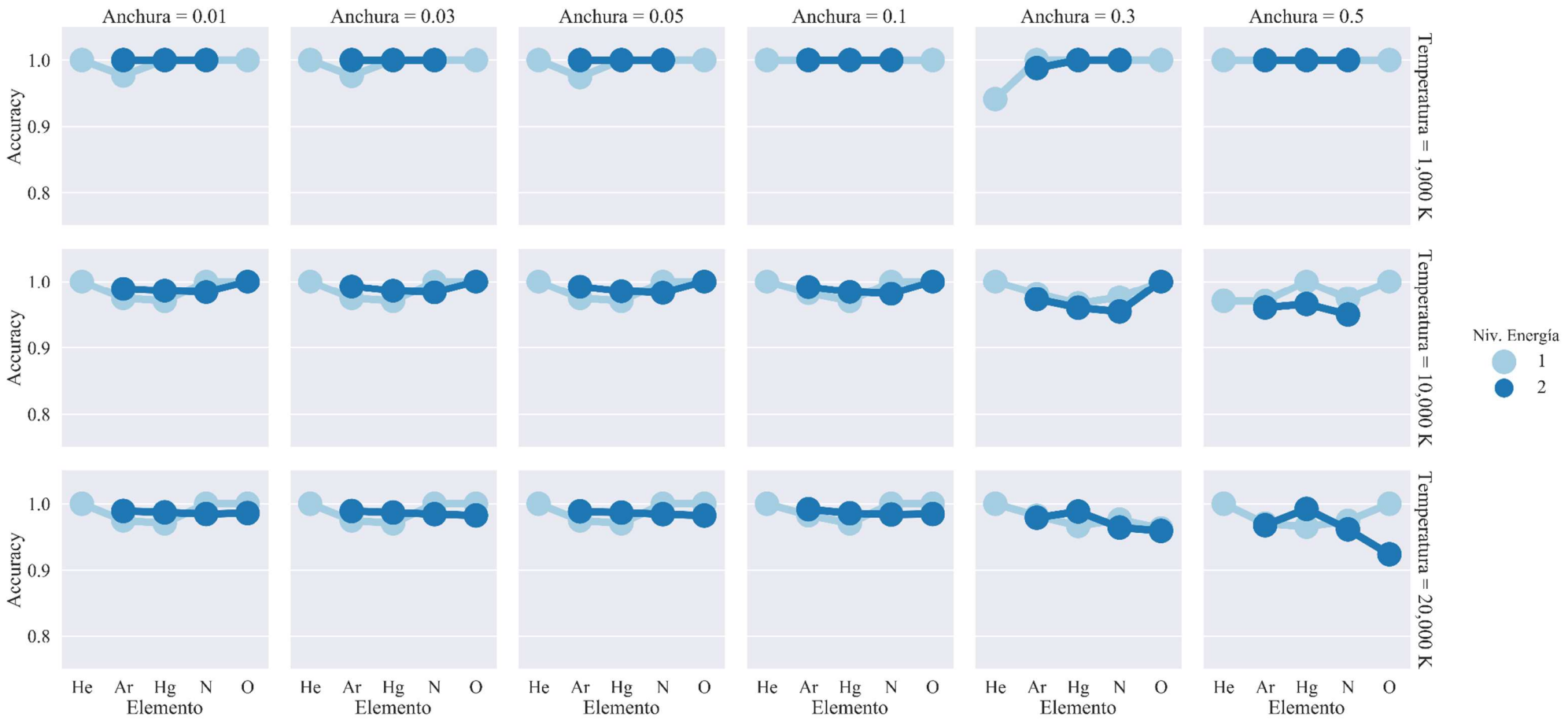


*Figura 5.1.6. Exactitud en las predicciones del modelo ensamblado con un **Paso** de 0.02 nm en distintas condiciones de **Anchura** y **Temperatura**.*

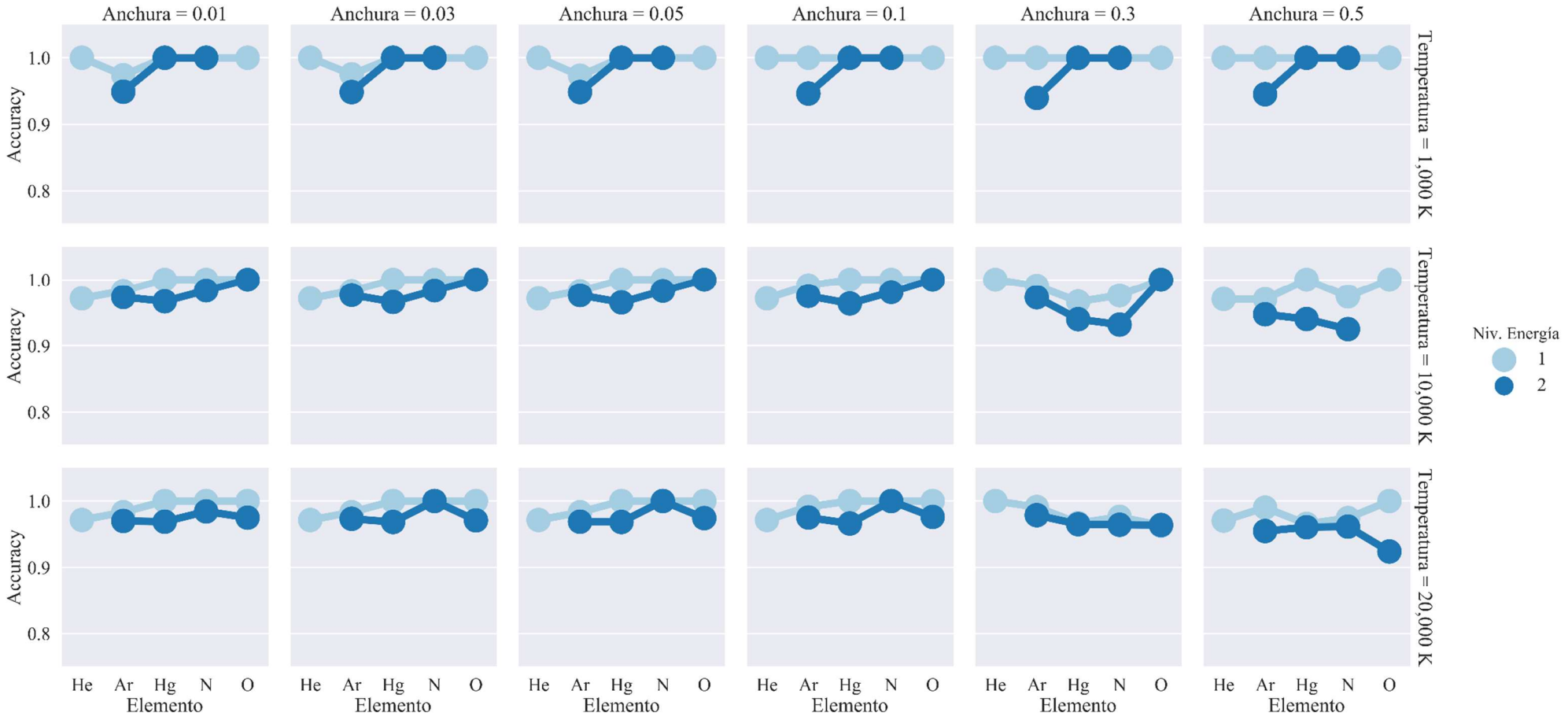


*Figura 5.1.7. Exactitud en las predicciones del modelo ensamblado con un **Paso** de 0.03 nm en distintas condiciones de **Anchura** y **Temperatura**.*

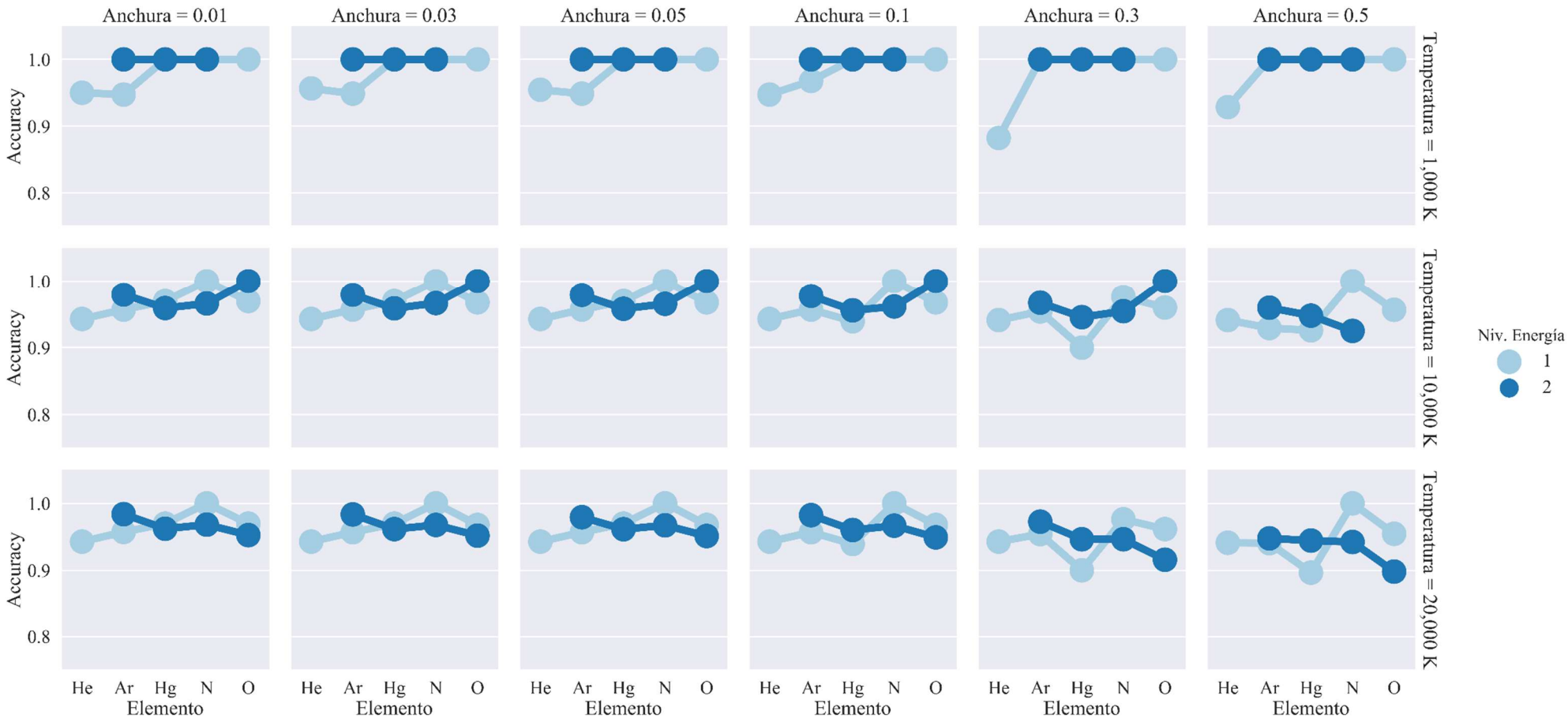


*Figura 5.1.8. Exactitud en las predicciones del modelo ensamblado con un **Paso** de 0.05 nm en distintas condiciones de **Anchura** y **Temperatura**.*

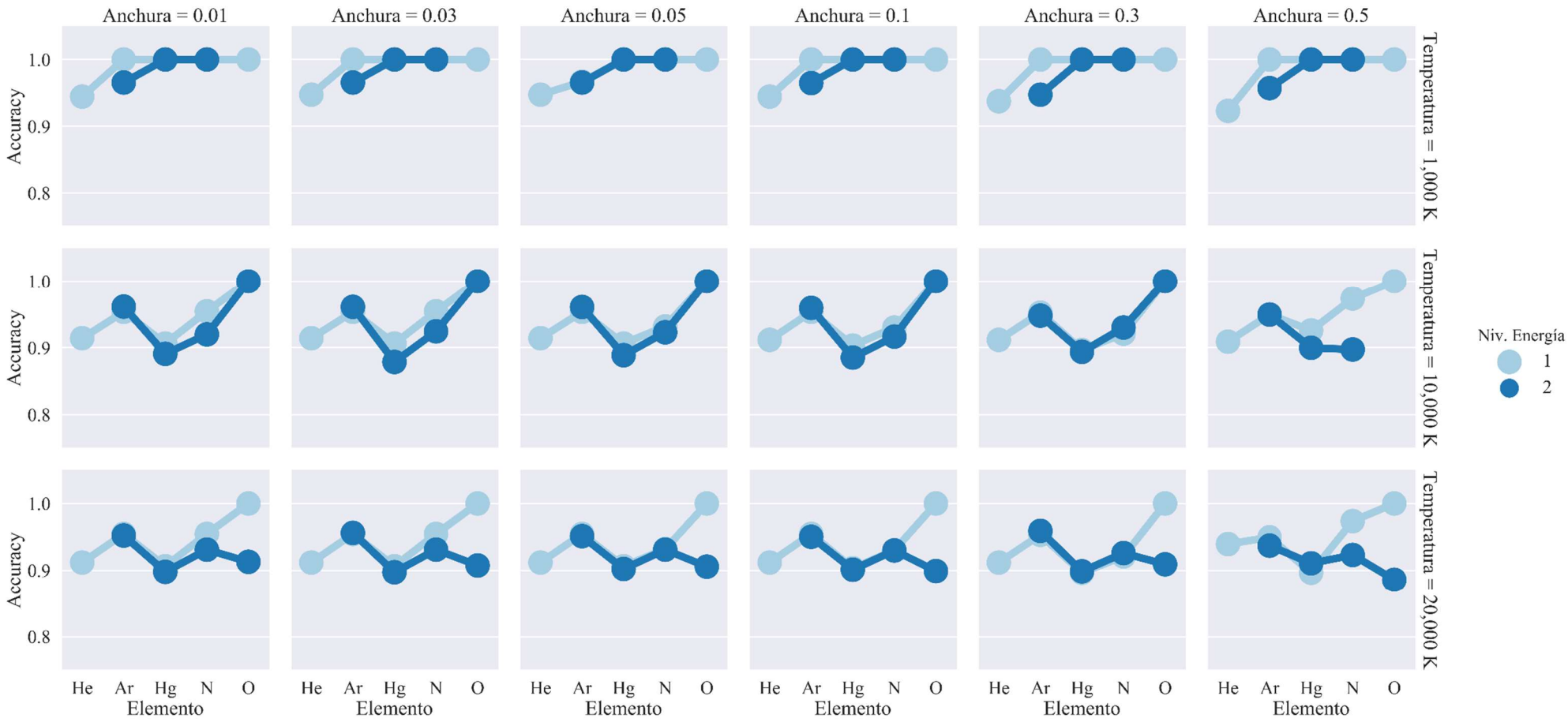


*Figura 5.1.9. Exactitud en las predicciones del modelo ensamblado con un **Paso** de 0.1 nm en distintas condiciones de **Anchura** y **Temperatura**.*

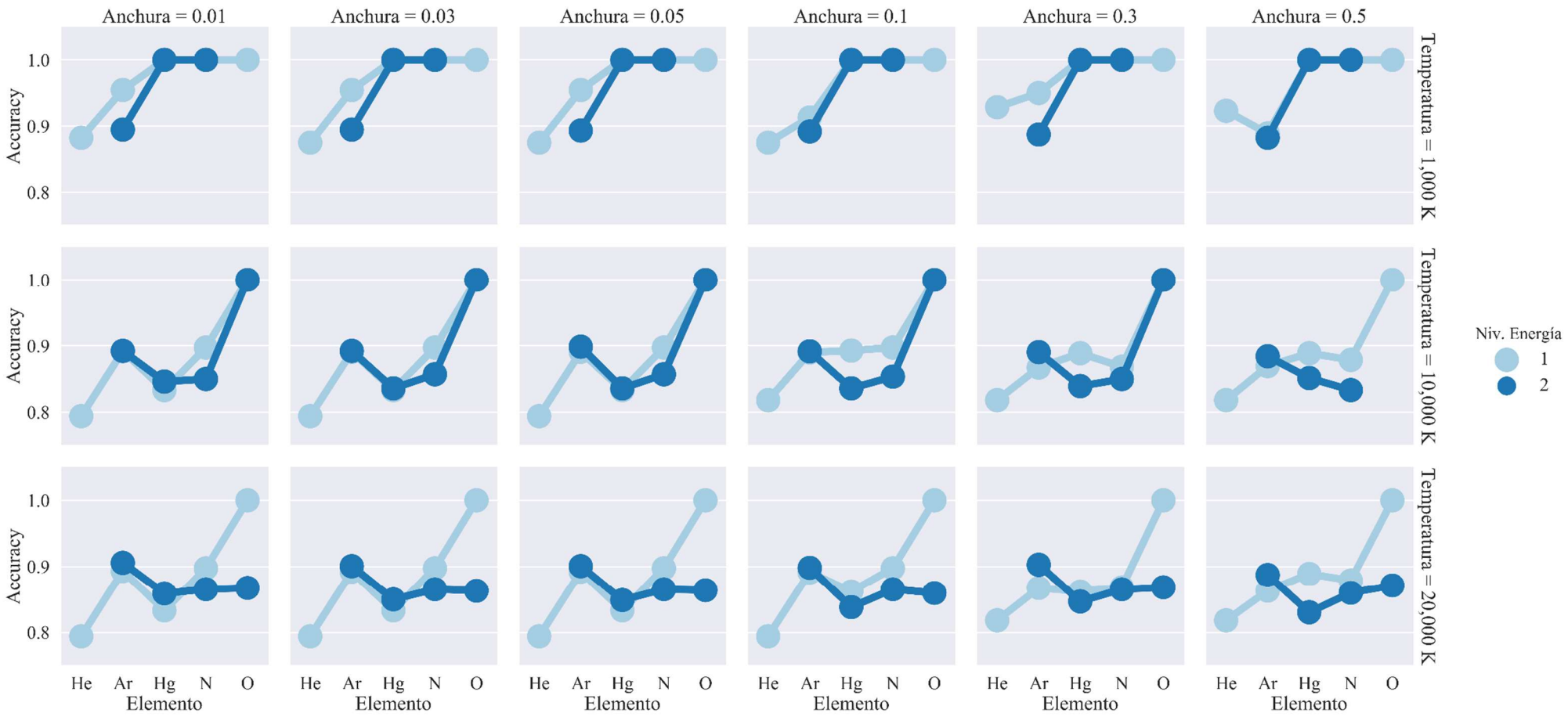


*Figura 5.1.10. Exactitud en las predicciones del modelo ensamblado con un **Paso** de 0.2 nm en distintas condiciones de **Anchura** y **Temperatura**.*

El efecto en el espectro sintético de variar de manera individual los parámetros: ***Paso***, ***Anchura*** y ***Temperatura***, se explican de la siguiente manera: a) se observa a partir de la Figura 5.1.1 que conforme se incrementa el valor del ***Paso*** se induce desplazamiento de picos en ambas direcciones, así como una deformación y ocultamiento de picos porque disminuye la cantidad de puntos que integran el espectro, b) la Figura 5.1.2 muestra que el incremento de ***Anchura*** provoca el ensanchamiento horizontal de los picos, que gradualmente se suman adyacentemente y distorsionan el espectro, hasta el punto que los 11 picos iniciales (ver Figura 5.1.2 **a**) se convierten en 5 picos (ver Figura 5.1.2 **f**), c) con la Figura 5.1.3 se observa que el incremento de ***Temperatura*** promueve la aparición de picos, esto sugiere que la intensidad mostrada en el espectro es proporcional a la energía presente en el espectro.

Los resultados de variar conjuntamente ***Paso***, ***Anchura*** y ***Temperatura*** en los espectros sintéticos con efecto en las predicciones es el siguiente: a) en las Figura 5.1.5,Figura 5.1.6 y Figura 5.1.7 se observan predicciones con una exactitud mayor a 0.9 para los niveles de energía 1 y 2, para todos los casos de ***Anchura*** y ***Temperatura***, para temperaturas menores o iguales a 10,000 K algunas veces se muestra la exactitud de O II, lo que sugiere que estos picos tienen poca intensidad y no siempre es posible detectarlos, b) en las Figura 5.1.8, Figura 5.1.9 y Figura 5.1.10 se aprecian oscilaciones en la exactitud de las predicciones que va desde 1.0 hasta 0.8, c) finalmente en la Figura 5.1.10 aún se consiguen puntuaciones de 1.0 para algunos elementos con nivel de energía 1 y 2.

5.2 Análisis Específico.

Una vez que se realizaron pruebas en distintas condiciones de ***Paso***, ***Anchura*** y ***Temperatura***, se determinaron las condiciones de ***Paso*** y ***Anchura*** en que se encuentran los espectros del Laboratorio de Física de Plasmas del ININ. Para determinar el ***Paso*** se calcularon las diferencias entre cada par de puntos adyacentes de longitud de onda de los espectros de la lámpara de calibración HG-1. En la Figura 5.2.1 se muestra un espectro experimental integrado por 33 archivos, en color azul fuerte se resaltan los archivos impares, y en azul claro los archivos pares. En base con esta figura se determina que los extremos del ***Paso*** van de 0.012 nm a 0.028 nm con centro en 0.02 nm.

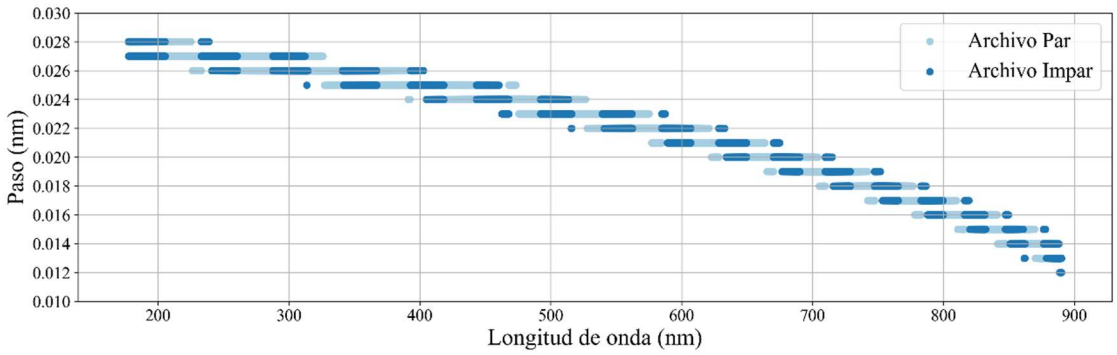


*Figura 5.2.1. Diferencia en **Paso** para cada par adyacente de longitudes de onda para los espectros experimentales de la lámpara de calibración HG-1.*

En lo respectivo al cálculo de la ***Anchura***, se ocuparon los mismos tres espectros experimentales de la lámpara de calibración HG-1, y se siguió el siguiente procedimiento por cada espectro para descartar ***Anchuras*** atípicas:

- Calcular todas las ***Anchuras*** a media altura del espectro.
- Calcular los estadísticos: media, mediana, primer (q_1) y tercer cuartil (q_3).
- Calcular rango intercuartílico: $iqr = q_3 - q_1$.
- Calcular: $umbral = iqr \cdot 1.5$.
- Calcular límites: $lim_{inf} = q_1 - umbral$, $lim_{sup} = q_3 + umbral$.
- Ordenar vector de ***Anchuras*** a media altura y guardar el resultado en vector_a.
- Calcular frontera de valores atípicos: $f_{sup} = vector\_a[vector\_a \leq lim_{sup}][0]$, $f_{inf} = vector\_a[vector\_a \geq lim_{inf}][-1]$
- Identificar valores atípicos: $posición\_atípico = [(vector\_a < f_{inf}) \mid (vector\_a > f_{sup})]$.

En la Figura 5.2.2 se muestran a la izquierda los gráficos de dispersión con puntos rojos los valores atípicos para la ***Anchura*** a media altura, y en color azul los valores no atípicos, así como una línea punteada que representa el valor de la mediana. En (b), (d) y (f) se observan gráficos de caja y bigotes que sirven para verificar si la detección de valores atípicos es correcta, el triángulo blanco se representa el valor de la media, y los colores rojo y azul tienen significado equivalente a los gráficos de dispersión (a), (c) y (e).

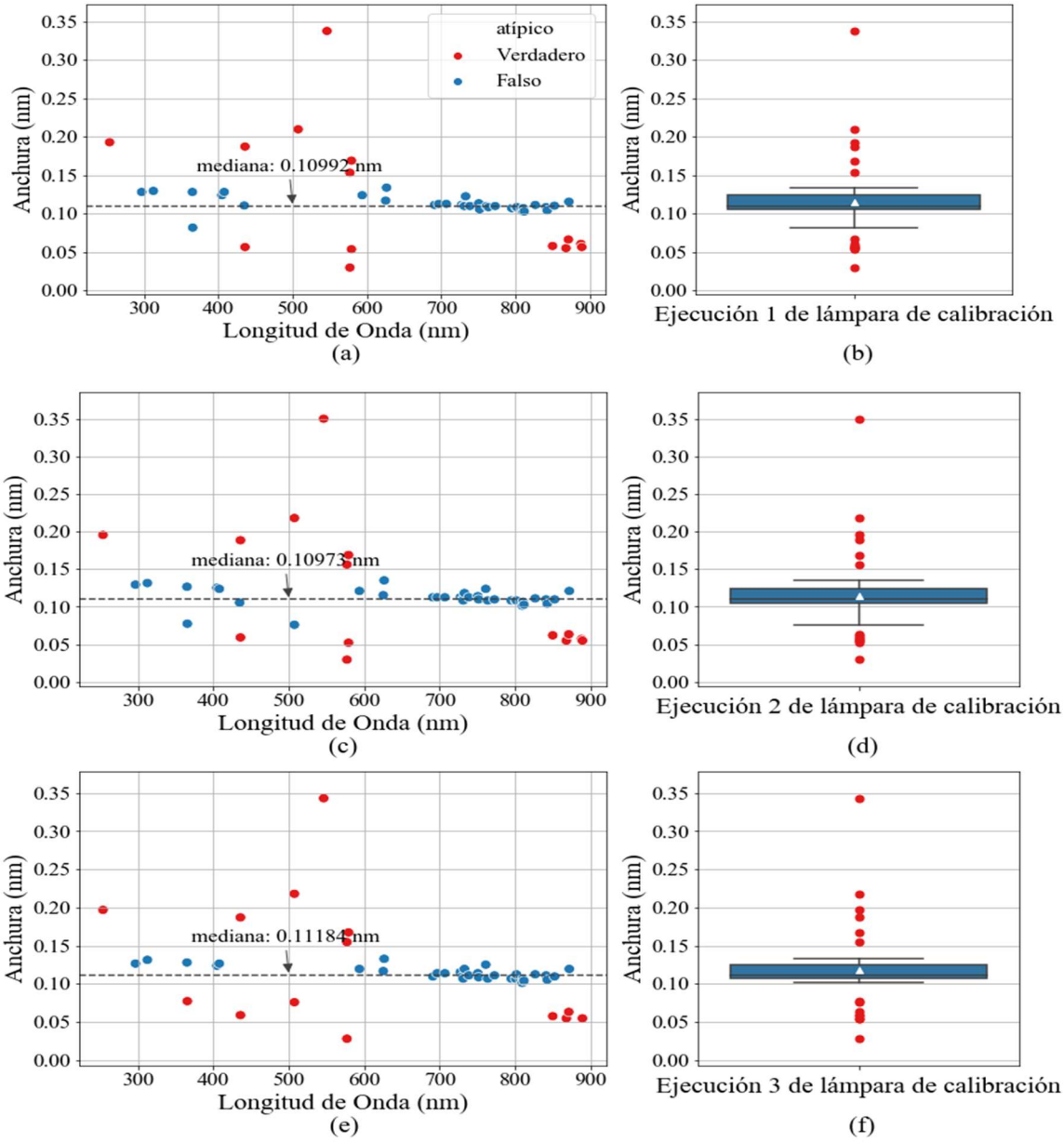


*Figura 5.2.2. Detección de valores atípicos en la **Anchura** a media altura para cada espectro experimental de la lámpara de calibración HG-1, las medianas para la **Anchura** a media altura son: (a) 0.10992 nm, (c) 0.10973 nm y (e) 0.11184 nm.*

Se observa en la Figura 5.2.2, que los valores de la mediana para ***Anchura*** en los tres espectros experimentales de la lámpara de calibración HG-1 tienen un valor próximo a 0.1 nm. Con los valores de ***Paso*** y ***Anchura*** encontrados se crearon 2,700 espectros sintéticos que siguen la configuración de la Tabla 5.2.1.

Tabla 5.2.1. Opciones para determinar rango de exactitud en predicciones.

Atributo	Opciones	Cantidad
Clase	Ar I, Ar II, He I, Hg I, Hg II, N I, N II, O I, O II	9
Paso	0.012, 0.02. 0.028	3
Anchura	0.11	1
Temperatura	Inicio: 200K, Fin: 20,000K, paso: 200K	100

Posteriormente, en cada espectro sintético generado se detectaron picos y realizaron predicciones. En la Figura 5.2.3 se observa la exactitud agrupada por ***Paso*** para todas las clases, el punto central representa la media, la longitud vertical de las líneas colores muestra la desviación estándar, mientras que las líneas internas de color negro son los intervalos de confianza al 95% utilizando *bootstrapping* y calculando los límites con los percentiles 0.025 y 0.975 respectivamente [48].

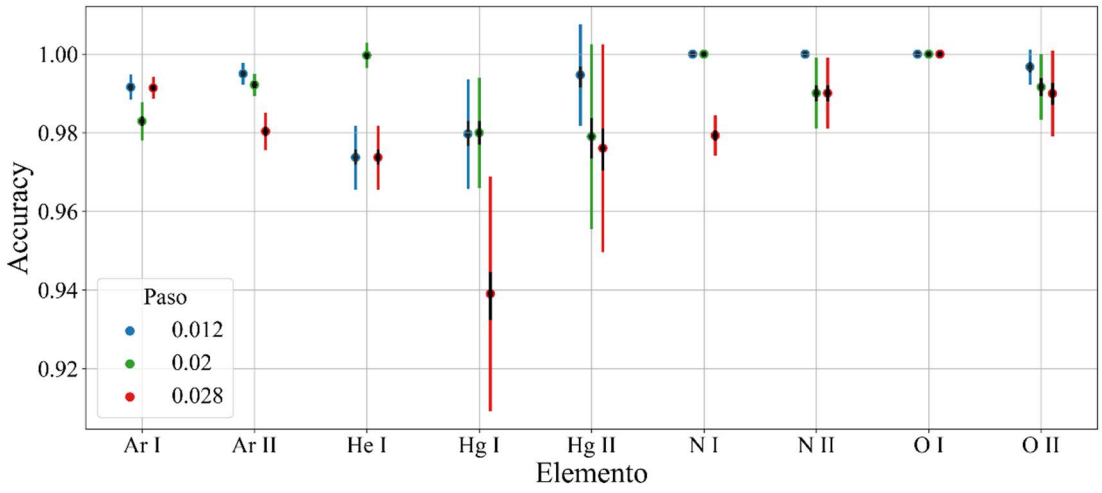


*Figura 5.2.3. Exactitud de las predicciones agrupadas por **Paso** para cada clase.*

Se aprecia en la Figura 5.2.3 que las longitudes de las líneas para los intervalos de confianza para cada clase son menores a su desviación estándar. Por otra parte, He I, N I, N II y O I presentan una exactitud del 1.0 en al menos un tamaño de paso, mientras que Hg I presenta la mayor desviación estándar para un paso de 0.028 nm, esto indica que hay valores que se alejan de su media en 0.93905, sin embargo, su intervalo de confianza indica que el 95% de las predicciones tienen una exactitud entre 0.93273 y 0.94423. Los datos que componen la Figura 5.2.3 se muestran en la Tabla 5.2.2.

*Tabla 5.2.2. Estadísticos generados a partir de la exactitud en las predicciones con los espectros sintéticos para las condiciones de **Paso** y **Anchura** encontrados.*

Elemento	Paso (nm)	Cantidad	Media	Desviación Estándar	Intervalo de Confianza Bootstrapping 95% Alto	Intervalo de Confianza Bootstrapping 95% Bajo	Intervalo de Confianza Bootstrapping 95% Diferencia
	0.012	99	0.99161	0.00286	0.99215	0.99103	0.00111
Ar I	**0.02**	99	0.98293	0.00453	0.98382	0.98201	0.00181
	0.028	99	0.99141	0.00239	0.99189	0.99091	0.00097
	0.012	99	0.99500	0.00242	0.99547	0.99450	0.00096
Ar II	**0.02**	99	0.99219	0.00247	0.99268	0.99173	0.00094
	0.028	99	0.98035	0.00438	0.98127	0.97956	0.00170
	0.012	99	0.97372	0.00782	0.97518	0.97229	0.00288
He I	**0.02**	99	0.99971	0.00287	1	0.99913	0.00086
	0.028	99	0.97372	0.00782	0.97518	0.97229	0.00288
	0.012	100	0.97970	0.01367	0.98235	0.97705	0.00529
Hg I	**0.02**	100	0.98	0.01378	0.98294	0.97735	0.00558
	0.028	100	**0.93905**	**0.02970**	**0.94423**	**0.93273**	**0.01149**
	0.012	100	0.99471	0.01256	0.99644	0.99180	0.00463
Hg II	**0.02**	100	0.97903	0.02329	0.98307	0.97402	0.00905
	0.028	100	0.97607	0.02625	0.98073	0.97065	0.01008
	0.012	100	1	0	1	1	0
N I	**0.02**	100	1	0	1	1	0
	0.028	100	0.97931	0.00484	0.98029	0.97843	0.00185
	0.012	100	1	0	1	1	0
N II	**0.02**	100	0.99008	0.00866	0.99182	0.98836	0.00346
	0.028	100	0.99007	0.00867	0.99181	0.98834	0.00347
	0.012	100	1	0	1	1	0
O I	**0.02**	100	1	0	1	1	0
	0.028	100	1	0	1	1	0
	0.012	72	0.99675	0.00413	0.99758	0.99583	0.00175
O II	**0.02**	72	0.99164	0.00810	0.99333	0.98979	0.00353

Se observa en la Tabla 5.2.2 que Ar I, Ar II, He I y O II no completan los 100 espectros sintéticos, esto se debe a que no se detectaron picos en sus espectros porque no se alcanzó suficiente intensidad, por tanto, de los 2,700 espectros sintéticos generados a partir de la Tabla 5.2.1 solo se ocuparon 2,607 en la predicciones. En la Figura 5.2.4 se aprecia el efecto del incremento de temperatura en la exactitud.

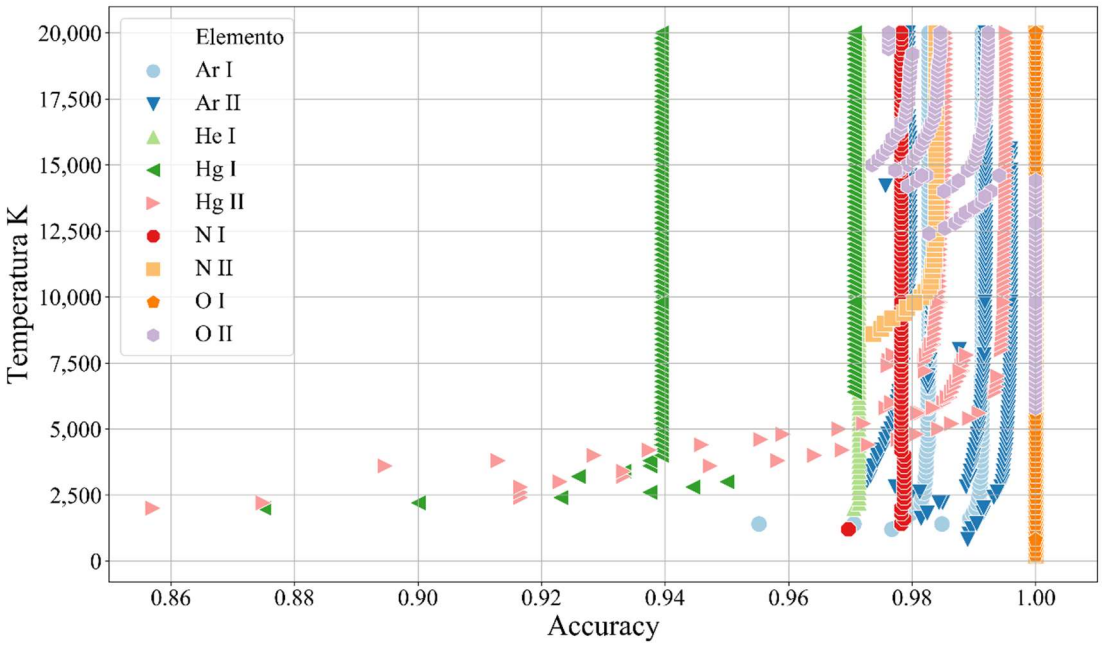


*Figura 5.2.4. Efecto del incremento de **Temperatura** sobre la exactitud en las predicciones.*

Se infiere con la Figura 5.2.4 que a partir de una ***Temperatura*** de 5,000 K se obtienen predicciones con una exactitud mayor a 0.94, también se observa que el incremento de temperatura tiene un efecto positivo en la exactitud de las predicciones, donde es notorio en las clases: Ar I, Ar II, Hg I, Hg II.

A partir de los intervalos de confianza al 95% y con exactitud en las predicciones comprendidas entre 0.93905 y 1.0, como se muestra en la Figura 5.2.3 y en la Tabla 5.2.2, se acepta la hipótesis nula que establece: mediante la incorporación de técnicas de Machine Learning en una interfaz de usuario se podrán caracterizar automáticamente especies de plasma frío en espectroscopia de emisión óptica con una precisión mayor o igual al 70%, en un intervalo de longitud de onda de 200 nm a 890 nm.

Cabe resaltar que, desde el planteamiento de este trabajo de tesis, hasta al momento no se encontró en la literatura estudios sobre el uso de técnicas de Machine Learning para la caracterización automática de especies en plasma no térmico, específicamente mediante la combinación de técnicas de clasificación mediante Árboles de Decisión. Por otra parte, tampoco se encontró en la literatura un análisis predictivo con técnicas de Machine con variación de ***Paso***, ***Anchura*** y ***Temperatura*** con el uso de espectros sintéticos, por tanto, se espera que este trabajo sea un referente para futuros trabajos en el área de la espectroscopia de emisión óptica en plasma no térmico con técnicas de Machine Learning.

CONCLUSIONES

La conclusión con base en la hipótesis de trabajo es la siguiente: A partir de los intervalos de confianza al 95% y con exactitud en las predicciones comprendidas entre 0.93 y 1.0, como se muestra en la Figura 5.2.3 y en la Tabla 5.2.2, se acepta la hipótesis nula que establece: mediante la incorporación de técnicas de Machine Learning en una interfaz de usuario se podrán caracterizar automáticamente especies de plasma frío en espectroscopia de emisión óptica con una precisión mayor o igual al 70%, en un intervalo de longitud de onda de 200 nm a 890 nm.

Las conclusiones en base a los objetivos establecidos en esta tesis son los siguientes:

1) Se implementaron técnicas y algoritmos de Machine Learning para determinar la posibilidad de caracterizar automáticamente las líneas de átomos excitados resultantes de la espectroscopia óptica de emisión de un plasma frío generado por descarga de barrera dieléctrica en un gas puro o en la degradación de compuestos recalcitrantes, con datos experimentales proporcionados por el Laboratorio de Física de Plasmas del ININ y datos sintéticos, para observar la variación en ***Paso***, ***Anchura*** y ***Temperatura***.
2) Se implementaron técnicas de Machine Learning para la caracterización automática de las siguientes especies: He, N, O, Ar y Hg, resultantes de la espectroscopia óptica de emisión de plasma frío en sus niveles I y II, dónde el Hg solo es considerado porque forma parte de la lámpara de calibración HG-1 y no es de interés para los experimentos y estudios realizados en el Laboratorio de Física de Plasmas del ININ.
3) Se identificó en qué medida se pueden caracterizar especies de plasma frio en espectroscopia de emisión óptica, en la Tabla 5.2.2 se muestra la exactitud media para todas las clases con sus respetivos intervalos de confianza al 95%, dónde en un extremo se tiene la menor exactitud con una media de 0.93905 para Hg I con un ***Paso*** de 0.028 nm, y en el otro extremo N I, N II y O I, con una exactitud media máxima de 1.0 para al menos un tamaño de ***Paso***.
4) Se usaron las líneas de especies reportadas en NIST como datos de entrenamiento para los algoritmos de Machine Learning, en primer lugar, se integraron en un repositorio local, y en segundo lugar se convirtió en un *DataFrame* con optimización de tipos. Finalmente se convirtió en un objeto almacenado en disco y cargado en memoria RAM

desde la interfaz gráfica diseñada en QT5, esto para recortar tiempos en la generación de modelos predictivos y en la creación datos sintéticos.

5) Por otra parte, se midió la efectividad de predicciones en espectros con distintas condiciones de ***Paso***, ***Anchura*** y ***Temperatura***, para lograrlo se creó un generador de espectros sintéticos detallado en 3.1.3 y se crearon dos conjuntos de datos, el primero estaba compuesto por 972 espectros sintéticos, su función fue establecer una idea general del rendimiento del clasificador (como se muestra en la sección en 5.1). El segundo conjunto de datos estaba integrado por 2,700 espectros sintéticos ajustados a las características de los espectros experimentales del Laboratorio de Física de Plasmas del ININ, para tener un referente del rendimiento del clasificador (ver Tabla 5.2.2), este procedimiento se detalla en la sección 5.2. Además, se precisa que la métrica utilizada se desprende de la matriz de confusión (ver Tabla 4.3.1) y corresponde a *Accuracy* ya que cada espectro generado corresponde a un elemento y nivel de energía, más no a una combinación de elementos y niveles de energía.
6) Se corrigió desplazamiento óptico del espectro con espectros obtenidos de lámpara de calibración HG-1 de la empresa Ocean Optics™, utilizando regresiones lineales (ver 3.2.11) sin valores atípicos dónde se obtuvo un valor máximo de 0.805007 para el coeficiente de determinación (Tabla 3.2.11.1), los coeficientes de la regresión polinomial de grado dos que corresponden a este valor se utilizaron para corregir el desplazamiento óptico y su efecto se muestra en la Figura 3.2.11.3. Cabe resaltar que este ajuste es parte fundamental de este trabajo porque permite obtener una aproximación de la posición real del pico para que el modelo del clasificador lo ubique en su correspondiente región de decisión como se observa en la Figura 4.5.1.
7) Se presenta el ajuste de hiperparámetros (ver Tabla 3.2.12.1) en algoritmos de Machine Learnig con la técnica *GridSearch* y validación cruzada repetida estratificada (ver sección 4.1) con tiempos de búsqueda variables alcanzado un máximo de 25.967 días (Tabla 4.4.2). Posteriormente se creó una validación cruzada repetida y validación cruzada repetida estratificada que mostraron un rendimiento similar, por lo que se aplicaron las pruebas de Friedman y ANOVA. Los resultados obtenidos muestran que las distribuciones para cada modelo son diferentes, así para obtener la distancia crítica y mostrar sus diferencias y ranking se aplicó la prueba de Nemenyi (Figura 4.4.3), con

base a esto se determina que el mejor algoritmo es Extremely Randomized Trees, dónde además se observó que conforme aumenta el número de divisiones en la validación cruzada, disminuyen las diferencias en los modelos generados. Posteriormente se creó una curva de validación, de esta se infiere que es necesario ocupar todos los datos de entrenamiento para crear el modelo final, por lo tanto, este modelo no tendría sobreajuste al no existir los estados de convergencia y divergencia entre las curvas de entrenamiento y validación cruzada. Finalmente se crearon curvas ROC con validación cruzada respecto a los datos de entrenamiento de las que se obtuvo que la media armónica F1 del clasificador es 0.9435 (Figura 4.4.5).

8) Se validó el modelo generado de Machine Learning como se mostró en la sección 5.2. Para hacerlo se calcularon los valores de ***Paso*** (Figura 5.2.1) y ***Anchura*** (Figura 5.2.2) con ***Temperatura*** variable desde los 200 K hasta los 20,000 K, para generar 2,700 espectros sintéticos que se utilizaron para medir la exactitud de las predicciones, dónde se obtuvo con un intervalo de confianza del 95% como se observa en el gráfico de la Figura 5.2.3 y la Tabla 5.2.2 que indican la exactitud de en las predicciones ubicada entre 0.93905 y 1.0. Adicionalmente se encontró que, el incremento de ***Temperatura*** tiende a mejorar la exactitud de las predicciones (Figura 5.2.4).
9) Mediante una interfaz gráfica de usuario (ver sección 3.4) se realizó la carga espectros para su graficación, caracterización y generación de reporte utilizando los lenguajes de programación Python y QT5 en función de los requerimientos funcionales (Figura 3.4.1.1) y no funcionales (Figura 3.4.1.2) del Laboratorio de Física de Plasmas del ININ. Los tiempos del software para la predicción automática de especies están en promedio en los 3 segundos y en la estimación de temperatura en 1 segundo. También que se generó una interfaz gráfica con Python y Jupyter para el generar espectros sintéticos (ver secciones 3.1.3 y 3.1.4) mediante la cual se generaron datos de validación que se usaron en el análisis de la variación de ***Paso***, ***Anchura*** y ***Temperatura*** (ver sección 5.1) así como la caracterización automática de especies y validación del modelo final (ver sección 5.2).
10) Otro de los objetivos concluidos fue la estimación de la temperatura electrónica de las especies de elementos detectadas (ver sección 3.3) la cual requirió de un proceso específico de recolección y tratamiento de datos. Para su cálculo se utilizaron los

valores en los extremos del espectro de un mismo elemento a los que posteriormente se aplica una regresión lineal o polinomial según elija el usuario. Este proceso representó una característica adicional como un primer bosquejo para la estimación automática de temperatura, dado que esta característica requiere de la pericia y experiencia del usuario para elegir las especies detectadas de interés.

En esta tesis se consideran como aportaciones principales

1) Se realizó un clasificador automático mediante técnicas de Machine Learning, para especies de plasma frío obtenidas mediante análisis de espectroscopía de emisión óptica.
2) Se creo un generador de espectros sintéticos en Jupyter Notebook que puede ser codificado por cualquier programador que utilice Python si se siguen las pautas descritas en la sección 3.1.3.
3) Se creó una interfaz gráfica para la caracterización automática de especies en QT5 cuya implementación se muestra en la sección 3.4.
4) Se realizó un primer análisis de las variables ***Paso***, ***Anchura*** y ***Temperatura*** que hasta el momento no se ha encontrado en la literatura un análisis igual o similar.

La importancia de dar una solución a la problemática planteada en este trabajo es:

1) Agilizar las investigaciones del Laboratorio de Física de Plasmas del ININ, potencialmente mediante la reducción de tiempos con el uso de la interfaz gráfica para realizar caracterización automática de especies, estimar la temperatura electrónica y generar un reporte que permite utilizar la información generada y hacer comprobaciones adicionales y correcciones puntuales si es necesario.
2) El reporte generado por la interfaz gráfica está integrado por las especies identificadas con sus respectivas probabilidades, áreas bajo la curva, posiciones de cada pico, así como el inicio y fin de cada anchura a media altura, y las temperaturas estimadas-

Como trabajos a futuro se plantea:

1) Utilizar todos los datos reportados en el NIST para trabajar con todos los elementos y sus niveles de energía para construir un modelo de Machine Learning.

2) Revisar y mejorar la implementación de la función de partición dependiente de la temperatura (3.1) porque durante la elaboración de este trabajo se realizaron observaciones sobre las altas intensidades que se alcanzaban en los picos de los espectros sintéticos generados.
3) Implementar la corrección automática de desplazamiento óptico para que trabaje con lámparas de calibración como la HG-1 y HG-2 de Ocean Optics™, con solo introducir los datos de longitud de onda, elemento y nivel de energía.
4) Recolectar espectros experimentales para todos los elementos y niveles de energía de interés desde un inicio y así contar con datos para contrastar así los resultados en las predicciones. La cantidad de espectros experimentales no se encuentra bien definido en la literatura, en [14] para espectroscopia de rayos gamma se ocupan 409 conjuntos de datos para el entrenamiento y 5 para prueba, mientras que en [15] para la caracterización de espectros PIXE se utilizan dos subconjuntos de datos, el primero está compuesto por 22 conjuntos de muestras orgánicas, de las cuales 18 se ocupan para entrenamiento y 4 para prueba, el segundo está compuesto por 37 conjuntos de datos de aerosoles, de estos 29 son para entrenamiento y 8 para prueba. Hay que recordar que en [17] se ocupan 4.4 millones de registros para clasificar estrellas y galaxias por medio de su espectro. Se observa que la cantidad de espectros es variable en la literatura, por tanto, una opción es consultar con el experto y determinar en conjunto la cantidad de espectros a utilizar.

REFERENCIAS

[1] N. Tkachenko, *Optical Spectroscopy: Methods and Instrumentations*, First. Tampere, Finland: Elsevier, 2006.

[2] A. Mercado-Cabrera *et al.*, "Simultaneous degradation of toxic organic pollutants by thin-falling-water-film DBD reactor," *Desalin. WATER Treat.*, vol. 101, pp. 157–169, 2018, doi: 10.5004/dwt.2018.21762.

[3] A. Mercado-Cabrera *et al.*, "Chlorobenzene Degradation in Simultaneous Gas–Liquid Phases Assisted by DBD Plasma," *IEEE Trans. Plasma Sci.*, vol. 47, no. 1, pp. 86–94, Jan. 2019, doi: 10.1109/TPS.2018.2877057.

[4] P. H. C. Eilers and H. F. M. Boelens, "Baseline Correction with Asymmetric Least Squares Smoothing," *Leiden Univ. Med. Cent. Rep.*, pp. 1–24, 2005.

[5] V. Van Asch, "Macro-and micro-averaged evaluation measures," *Univ. Antwerp*, vol. 49, pp. 1–27, 2013.

[6] E. Restrepo and A. Devia, "Caracterización de materiales utilizando la espectroscopía óptica de emisión," *Rev. Colomb. Física*, vol. 34, no. 2, pp. 478–483, 2002.

[7] A. Sáinz, M. C. García, and M. D. Calzada, "Spectroscopic determination of the electron temperature in non-LTE argon and neon plasmas," *32nd EPS Conf. Plasma Phys. 2005, EPS 2005, Held with 8th Int. Work. Fast Ignition Fusion Targets - Europhys. Conf. Abstr.*, vol. 29C, pp. 1842–1845, 2005.

[8] S. S. Hamed, "Spectroscopic Determination of Excitation Temperature and Electron Density in Premixed Laminar Flame," *Egypt. J. Solids*, vol. 28, no. 2, pp. 349–357, 2005.

[9] W. Wang, S. Wang, F. Liu, W. Zheng, and D. Wang, "Optical study of OH radical in a wire-plate pulsed corona discharge," *Spectrochim. Acta - Part A Mol. Biomol. Spectrosc.*, vol. 63, no. 2, pp. 477–482, 2006, doi: 10.1016/j.saa.2005.05.033.

[10] A. Sarani, A. Y. Nikiforov, and C. Leys, "Atmospheric pressure plasma jet in Ar and Ar/H2O mixtures: Optical emission spectroscopy and temperature measurements," *Phys. Plasmas*, vol. 17, no. 6, pp. 1–8, 2010, doi: 10.1063/1.3439685.

[11] A. Garduño Aparicio, "Adquisición de espectros ópticos para la estimación de temperatura electrónica," M.S. Thesis, Universidad Autónoma del Estado de México, Centro Universitario UAEM Atlacomulco, 2015.

[12] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2014.

[13] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, Second. O'Reilly Media, 2019.

[14] E. Yoshida, K. Shizuma, S. Endo, and T. Oka, “Application of neural networks for the analysis of gamma-ray spectra measured with a Ge spectrometer,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 484, no. 1–3, pp. 557–563, 2002, doi: 10.1016/S0168-9002(01)01962-3.

[15] R. Correa Deves, “Redes Neuronales Artificiales en Ingeniería y Física Nuclear . Caracterización de espectros PIXE,” Ph.D. dissertation, Universidad de Granada, 2006.

[16] O. Miettinen, “Protostellar classification using supervised machine learning algorithms,” *Astrophys. Space Sci.*, vol. 363, no. 9, pp. 1–17, 2018, doi: 10.1007/s10509-018-3418-7.

[17] Y. Bai, J. Liu, S. Wang, and F. Yang, “Machine Learning Applied to Star–Galaxy–QSO Classification and Stellar Effective Temperature Regression,” *Astron. J.*, vol. 157, no. 1, p. 9, 2018, doi: 10.3847/1538-3881/aaf009.

[18] J. R. Quinlan, “Induction of Decision Trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/bf00116251.

[19] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[20] M. Sebban, R. Nock, J. H. Chauchat, and R. Rakotomalala, “Impact of Learning Set Quality and Size,” *Int. J. Comput. Syst. Signal*, vol. 1, no. 1, pp. 85–105, 2000.

[21] D. A. Cieslak and N. V. Chawla, “Learning decision trees for unbalanced data,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5211 LNAI, no. PART 1, pp. 241–256, 2008, doi: 10.1007/978-3-540-87479-9_34.

[22] N. Patel and S. Upadhyay, “Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA,” *Int. J. Comput. Appl.*, vol. 60, no. 12, pp. 20–25, 2012, doi: 10.5120/9744-4304.

[23] W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla, “A Robust Decision Tree Algorithm for Imbalanced Data Sets,” *Proc. 10th SIAM Int. Conf. Data Mining, SDM 2010*, pp. 766–777, 2010, doi: 10.1137/1.9781611972801.67.

[24] J. Gou, L. Du, Y. Zhang, and T. Xiong, “A New Distance-weighted k-nearest Neighbor Classifier,” *J. Inf. Comput. Sci.*, vol. 9, no. 6, pp. 1429–1436, 2012.

[25] S. A. Dudani, “The Distance-Weighted k-Nearest-Neighbor Rule,” *IEEE Trans. Syst. Man Cybern.*, vol. 6, no. 4, pp. 325–327, 1976.

[26] T. E. Oliphant, “Python for Scientific Computing,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 10–20, 2007.

[27] R. Kumar, “Future For Scientific Computing Using Python,” *Int. J. Eng. Technol. Manag. Res.*, vol. 2, no. 1, pp. 30–41, 2015.

[28] W. Yu, M. Carrasco Kind, and R. J. Brunner, “Vizic: A Jupyter-based Interactive Visualization Tool for Astronomical Catalogs,” *Astron. Comput.*, vol. 20, pp. 128–

139, 2017, doi: 10.1016/j.ascom.2017.06.004.

[29] P. Podrzaj, “A brief demonstration of some Python GUI libraries,” 2019, pp. 1–6.

[30] I. Guyon, J. Makhoul, and R. Schwartz, “Design of experiments for the NIPS 2003 variable selection benchmark Isabelle Guyon – July 2003,” *Test*, no. July, 2003.

[31] L. Breiman, “Bagging Predictors,” *Mach. Learn.*, vol. 24, no. 421, pp. 123–140, 1996, doi: 10.1007/BF00058655.

[32] D. J. Flannigan, “Spreadsheet-Based Program for Simulating Atomic Emission Spectra,” *J. Chem. Educ.*, vol. 91, no. 10, pp. 1736–1738, 2014, doi: 10.1021/ed500479u.

[33] J. Barnes, *Azure Machine Learning Microsoft Azure Essentials*. Microsoft Press, 2015.

[34] R. Barga, V. Fontama, and W. H. Tok, *Predictive Analytics with Microsoft Azure Machine Learning*, Second. Apress, 2015.

[35] O. Irsoy, O. T. Yildiz, and E. Alpaydin, “Design and Analysis of Classifier Learning Experiments in Bioinformatics: Survey and case studies,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 9, no. 6, pp. 1663–1675, 2012, doi: 10.1109/TCBB.2012.117.

[36] A. Halevy, P. Norving, and F. Pereira, “The Unreasonable Effectiveness of Data,” *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 8–12, 2009.

[37] B. M. Randles, I. V. Pasquetto, M. S. Golshan, and C. L. Borgman, “Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study,” in *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Jun. 2017, pp. 1–2, doi: 10.1109/JCDL.2017.7991618.

[38] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification Algorithms and Regression Trees*. Taylor & Francis Ltd, 1984.

[39] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.

[40] R. C. Tausworthe, *Standardized Development of Computer Software. Part {II}. Standards*. Pasadena, California: National Aeronautics and Space Administration, 1979.

[41] J. P. Mueller and L. Massaron, *Python for Data Science For Dummies*, Second. 2019.

[42] L. De Galan, R. Smith, and J. D. Winefordner, “The electronic partition functions of atoms and ions between 1500 K and 7000 K,” *Spectrochim. Acta Part B At. Spectrosc.*, vol. 23, no. 8, pp. 521–525, 1968.

[43] J. Ingle, J. D. and S. R. Crouch, *Spectrochemical Analysis*. Upper Saddle River, NJ: Prentice Hall, 1988.

[44] S. Vluymans, “Dealing with imbalanced and weakly labelled data in machine learning using fuzzy and rough set methods,” Ph.D. dissertation, Universidad de Granada, Belgium, 2019.

[45] D. H. Wolpert and W. G. Macready, “No Free Lunch Theorems for Search,” *Tech. Rep. SFI-TR-95-02-010*, pp. 1–38, 1996, doi: 10.1145/1389095.1389254.

[46] N. Yadah, A. Yadah, J. C. Bansal, K. Deep, and J. H. Kim, Eds., *Harmony Search and Nature Inspired Optimization Algorithms*, vol. 741. Singapore: Springer, 2019.

[47] S. A. Agudelo Estrada, “Determinación de la Concentración Elemental del Plasma de Aire Atmosférico Producido por la Técnica Laser-Induced Breakdown Spectroscopy,” B.S. Thesis, Universidad Tecnológica de Pereira, 2017.

[48] J. Robertson and M. Kaptein, Eds., *Modern Statistical Methods for HCI*. Cham: Springer International Publishing, 2016.

GLOSARIO

Concepto	Definición
Na	Sodio, número atómico 11.
Si	Silicio, número atómico 14.
Ar	Argón, número atómico 18.
He	Helio, número atómico 2.
N	Nitrógeno, número atómico 7.
Hg	Mercurio, número atómico 80.
O	Oxígeno, número atómico 8.
AUROC	Area Under the Receiver Operative Characteristics.
CCD	Charge-Coupled Device.
CSV	Comma Separated Values.
DAT	Data File.
DataFrame	Estructura de datos que maneja valores como si se tratará de una tabla. Esta estructura está incluida en la biblioteca de funciones Pandas para Python.
EEO	Espectroscopia de Emisión Óptica.
Características	
Clases	
F1	Métrica de clasificación utilizada en *Machine Learning* para determinar las predicciones correctas con una matriz de confusión.
H_2O	Agua.
Hiperparámetro	Variable de configuración externa al modelo cuyo valor no puede ser estimado a partir de los datos. Un valor que forme parte de la solución se obtiene por ensayo y error, o por búsqueda combinatoria, por ejemplo, en un árbol de decisión un hiperparámetro es la profundidad máxima.
ININ	Instituto Nacional de Investigaciones Nucleares.
ML	Machine Learning.
N_2	Nitrógeno molecular o dinitrógeno.
NIST	National Institute of Standards and Technology.

Concepto	Definición
NO_2	Dióxido de Nitrógeno.
O_2	Oxígeno o Dioxígeno.
O_3	Ozono.
OH	Hidróxido.
Parámetro	Variable de configuración interna al modelo cuyo valor puede ser estimado a partir de lo datos. En estadística se puede estimar la media y la desviación estándar a partir de los datos que integran una distribución Gaussiana. En el contexto de programación, un parámetro es el tipo de variable que se define como entrada en una función, mientras que un argumento es el valor que recibe una función cuando esta es invocada.
PIXE	Proton Induced X-Ray Emissions.
SMOTE	Synthetic Minority Over-sampling Technique.
SVM	Support Vector Machine.
UV	Ultravioleta.
Vis	Visible