Evaluación de Recubrimientos de Perovskita Mediante el Uso de Inteligencia Artificial Para Mejorar la Eficiencia de Celdas Solares

Juan Diego Sierra Villegas

Trabajo de Grado para Optar al Título de Ingeniero Mecánico

Director
Octavio Andrés González Estrada
Ingeniero Mecánico, PhD.

Codirector
Cristian Andrés Hernández Salazar
Ingeniero Mecánico

Universidad Industrial de Santander
Facultad de Ingenierías Físico-Mecánicas
Escuela de Ingeniería Industrial
Bucaramanga
2024

Tabla de Contenido

Pág.

Lista de Tablas

Pág.

Lista de Figuras

Resumen

Título: Evaluación de Recubrimientos de Perovskita Mediante el Uso de Inteligencia Artificial Para Mejorar la Eficiencia de Celdas Solares[*]

Autor: Juan Diego Sierra Villegas[**]

Palabras Clave: Perovskita, inteligencia artificial, machine learning, celdas solares, eficiencia.

Descripción: Las celdas solares de perovskita son de gran interés por sus propiedades fotoeléctricas. En este trabajo se entrenaron 5 diferentes modelos de aprendizaje automático (LSTM, bosques aleatorios, SVM, gradient boosting y XGBoost) en lenguaje Python con un conjunto de datos de entrada categóricos referentes a la fabricación de celdas de perovskitas, como lo son los materiales de las capas de transporte de huecos y de electrones, la perovskita fotovoltaica usada y su proceso de deposición, material contacto electrónico trasero, anti-solvente y solución precursora, para llevar a cabo tareas de regresión con la eficiencia de las celdas como variable objetivo. Luego de la evaluación de los modelos, se identificó el modelo de bosques aleatorios como el más apropiado entre los estudiados debido su capacidad de ajuste a los datos reales y su tiempo. Con este modelo se logró una caracterización satisfactoria de celdas solares de perovskita según su eficiencia de conversión energética teniendo datos de sus parámetros de fabricación, pero con resultados deficientes en la obtención de las mejores combinaciones de parámetros para maximizar dicha eficiencia. Se aconseja llevar a cabo nuevos estudios con diferentes enfoques para obtener modelos capaces de cumplir con esta tarea de forma eficaz.

[*] Trabajo de Grado.

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica. Director: Octavio Andrés González Estrada. Ingeniero Mecánico, PhD. Codirector: Cristian Andrés Hernández Salazar. Ingeniero Mecánico.

Abstract

Title: Evaluation of Perovskite Coatings Using Artificial Intelligence to Improve the Efficiency of Solar Cells [*]

Author: Juan Diego Sierra Villegas[**]

Key Words: Perovskite, artificial intelligence, machine learning, solar cells, efficiency.

Description: Perovskite solar cells are of great interest nowadays due to their photoelectric properties. In this work, five different machine learning models (LSTM, random forests, SVM, gradient boosting and XGBoost) were trained in Python language with a set of categorical input data concerning the manufacture of perovskite cells, such as the electron and hole transport layer materials, the photovoltaic perovskite used and its deposition process, electronic back contact material, anti-solvent, and precursor solution, in order to perform regression tasks with cell efficiency as the target variable. After evaluating the different models, the random forest model was identified as the most appropriate among those studied due to its ability to fit the real data and its run time. With this model, a satisfactory characterisation of perovskite solar cells according to their energy conversion efficiency was achieved using data on their manufacturing parameters, but with poor results in obtaining the best combinations of said parameters to maximise this efficiency. It is advisable to conduct further studies with different approaches to obtain models capable of fulfilling this task efficiently.

[*] Bachelor thesis.

[**] Faculty of Physical-Mechanical Engineering. School of Mechanical Engineering. Director: Octavio Andrés González Estrada. Mechanical Engineer, PhD. Co-Director: Cristian Andrés Hernández Salazar. Mechanical Engineer.

Introducción

Durante los últimos años, las celdas solares de perovskitas han mostrado avances sobresalientes en cuanto a la eficiencia de transformación de energía que pueden llegar a alcanzar, con valores certificados de hasta 26.1% (NREL, 2024). A pesar de esto, sigue presentando graves problemas que los hacen imprácticos a nivel de aplicación comercial, tales como la estabilidad de las celdas, su confiabilidad, desconocimiento de métodos y materiales para su manufactura óptima y falta de métodos de prueba estandarizados (Jacobsson et al., 2022). Se han abordado algunos problemas con herramientas de inteligencia artificial.

Ning et al. (2020) encontraron usando métodos de aprendizaje automático que adicionar una pequeña capa de un compuesto llamado DBTPA a la celda puede llevar a un aumento significativo su desempeño. Encontraron una mejora de eficiencia del 19.9% a 20.6%. Concluyeron que el DBTPA contribuye también a un aumento considerable en la estabilidad térmica de la celda.

Odabaşı y Yıldırım (2020) analizaron los factores con mayor influencia en la estabilidad de las celdas usando reglas de asociación y árboles de decisión en lenguaje R. Hallaron que el uso de tolueno como anti solvente, capas de batiocuproína como inter-capa de transporte de electrones y plata como capa de electrodo negativo tienen efectos beneficiosos en la eficiencia y estabilidad de las celdas, además de disminuir la histéresis y aumentar la reproductibilidad de estas.

Por su parte, Harth et al. (2023) desarrollaron un método de extracción de brecha de energía y calidad de captación en celdas de perovskita fabricadas por el método *blade-coating* mediante regresión con redes neuronales convolucionales. Su trabajo contribuyó con una forma rápida de

caracterización de las celdas por medio de imágenes, con posibilidad de adaptarse a la obtención de otras variables de interés incluso en líneas de producción a gran escala.

Ichwani et al. (2023) utilizaron métodos de aprendizaje automático con el fin de optimizar los parámetros del proceso de *spray coating* en la fabricación de celdas de perovskita, maximizando la eficiencia de conversión de energía. Lograron un modelo de regresión capaz de predecir de forma precisa la eficiencia de las celdas, además de desarrollar una red neuronal convolucional capaz de predecir los defectos superficiales para diferentes parámetros de aplicación.

Este trabajo propone el desarrollo de varios modelos de regresión capaces de predecir el rendimiento de celdas de perovskita teniendo información sobre los materiales y métodos utilizados en su fabricación, buscando identificar el mejor modelo para esta tarea. De esta forma se busca impulsar la adopción de los paneles solares de perovskita como fuente de energía renovable eficiente, sostenible y práctica.

1. Objetivos

1.1 Objetivo General

Evaluar recubrimientos de perovskita mediante el uso de inteligencia artificial para mejorar la eficiencia de celdas solares.

1.2 Objetivos Específicos

Recopilar y analizar datos experimentales de acceso abierto referentes a la fabricación y el rendimiento de los recubrimientos de perovskita, lo cual incluye variables de proceso, composiciones, propiedades ópticas y electrónicas, así como condiciones ambientales. Estos datos servirán como base para el entrenamiento de modelos de inteligencia artificial.

Desarrollar un modelo de inteligencia artificial, como algoritmos de regresión y clasificación, para predecir las propiedades y el rendimiento de los recubrimientos de perovskita. Este modelo debe ser capaz de analizar múltiples variables y encontrar relaciones complejas que ayuden a optimizar la eficiencia de las celdas solares.

Utilizar el modelo de inteligencia artificial desarrollado para realizar iteraciones de diseño y optimización de los recubrimientos de perovskita. A través de la exploración sistemática de diferentes combinaciones de variables y parámetros, buscar mejorar la eficiencia de conversión de energía solar de las celdas solares.

2. Marco Teórico

2.1 Perovskitas Fotovoltaicas

Se trata de combinaciones orgánica-inorgánicas que presentan una estructura cristalina similar a un mineral formado por trióxido de titanio y calcio ($CaTiO_3$) nombrado en honor al minerólogo ruso Lev Perovski (Stoumpos et al., 2013). Fueron introducidas a la fabricación de celdas solares en 2009 con eficiencias de conversión bastante bajas de alrededor de 3.8% (Kojima et al., 2009), pero han alcanzado en últimos años eficiencias superiores al 25% (Yoo et al., 2021), mostrando un gran potencial en el campo de energías alternativas. Sin embargo, su aplicación sigue presentando problemas importantes, como lo son la estabilidad a largo plazo, confiabilidad, mejores combinaciones de materiales y métodos de manufactura, y la falta de métodos de prueba estandarizados (Jacobsson et al., 2022).

La fabricación de estas celdas se lleva a cabo en capas. La primera capa es el electrodo negativo que permite el paso de electricidad al circuito, con materiales como plata u oro. Luego está la capa transportadora de electrones (ETL por sus siglas en inglés), que facilita el transporte de electrones desde la perovskita al electrodo negativo, y se utilizan materiales orgánicos o dióxido de titanio. La tercera es una capa activa de perovskita, seguida de una capa transportadora de huecos (HTL por sus siglas en inglés), que transporta las cargas positivas generadas en la celda. Por último, está la capa que actúa como sustrato, que funciona como estructura de la celda y suele ser de vidrio (Yang et al., 2015).

2.2 Inteligencia Artificial

La inteligencia artificial puede entenderse como la capacidad de una máquina o programa de tomar decisiones en condiciones no preestablecidas, además de ser capaz de aprender y mejorar tal proceso con base en experiencias pasadas (Mellit & Kalogirou, 2018). Incluye el estudio de teorías, tecnologías, métodos y aplicaciones para simular, extender y expandir la inteligencia humana. Hoy en día ha logrado un impacto muy profundo en la vida humana, sobrepasando los límites del cerebro humano en capacidad de volumen y velocidad de procesamiento de datos (Jiang et al., 2022). Entre las disciplinas de aplicación de esta ha destacado el Machine Learning (ML) o Aprendizaje Automático (AA) en español. Corresponde al uso de sistemas computacionales que no requieren de programación explicita para el aprendizaje de la tarea demandada (Morgan & Jacobs, 2020). Suele agruparse en dos grandes categorías: aprendizaje supervisado y no supervisado. El primero permite la capacidad de aprendizaje según estructuras de datos etiquetadas, se busca encontrar relaciones entre variables de salida Y y variables de entrada X definidos. En cambio, el no supervisado aprende propiedades de conjuntos de datos sin intervención humana (Jordan & Mitchell, 2015).

2.3 Algoritmos de Regresión

Son técnicas que se utilizan para modelar y predecir relaciones entre variables. Se enfocan en problemas donde la variable de interés es continua, es decir, utiliza valores numéricos en lugar de etiquetas categóricas. El objetivo principal de los algoritmos de regresión es estimar la función que describe la relación entre las variables de entrada y las variables de salida (Trinchero & Canavero, 2021). Algunos ejemplos aplicados en el aprendizaje automático son LSTM, bosques aleatorios, gradient boosting, máquinas de vectores de soporte y extreme gradient boosting.

2.3.1 LSTM

Es un tipo de red neuronal recurrente con estructura compuesta por puertas de entrada, salida y olvido, que le permiten retener información importante a lo largo del tiempo y olvidar la menos relevante (Hochreiter & Schmidhuber, 1997). Su importancia en la ciencia de materiales radica en su capacidad de identificar patrones temporales complejos en datos experimentales, permitiendo la predicción de variables objetivo a lo largo del tiempo (Z. Zhang et al., 2018).

2.3.2 Bosques Aleatorios

Es un método de aprendizaje automático que funciona construyendo árboles de decisión, y cada uno de estos toma una muestra aleatoria del conjunto de entrenamiento para identificar la relación entre las variables. Gracias a este muestreo, construye modelos robustos y resistentes al sobreajuste (Breiman, 2001).

2.3.3 Máquinas de Vectores de Soporte (SVM)

Son un tipo de algoritmo de aprendizaje automático que funciona trazando líneas (hiperplanos) entre diferentes variables en un conjunto de datos. Es capaz de encontrar hiperplanos que minimicen la diferencia entre predicciones y valores reales, aproximando así funciones complejas. Son eficaces en el manejo de datos no lineales (Cortes & Vapnik, 1995).

2.3.4 Gradient Boosting

Es una técnica de aprendizaje por conjunto que funciona construyendo una serie de modelos de árbol de decisión débiles, y combina sus predicciones para formar un modelo más robusto. Ajusta a cada nuevo árbol los errores de los modelos previos, reduciendo la pendiente de la función de pérdida. Captura patrones no lineales y relaciones complejas entre datos, ayudando a la precisión de las predicciones (Friedman, 2001).

2.3.5 Extreme Gradient Boosting (XGBoost)

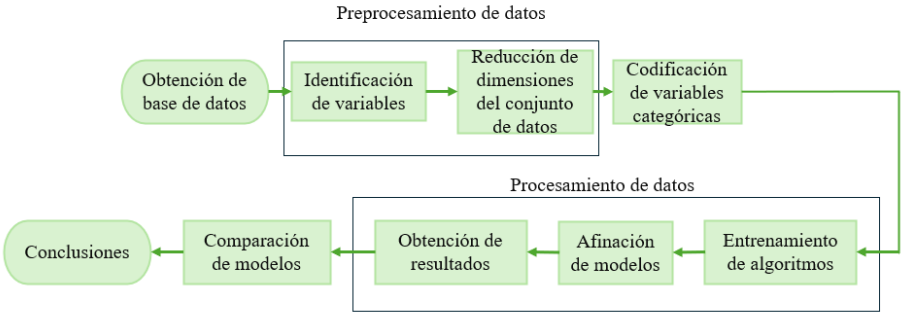
Se trata de una aplicación del algoritmo Gradient Boosting que busca mejorar las capacidades incluyendo técnicas de regularización de datos y manejo de valores faltantes. Tiene capacidad de manejar conjuntos de datos grandes y complejos, y busca mejorar la velocidad y eficiencia del modelo computacional (Chen & Guestrin, 2016).

3. Metodología

Se utilizó la base de datos de acceso abierto *The Perovskite Database Project* (Jacobsson et al., 2022) que cuenta con datos de más de 43000 celdas solares de perovskitas extraídos de diferentes artículos a partir del año 2009. A partir de estos datos se llevó a cabo un preprocesamiento, seguido de desarrollo y entrenamiento de algoritmos, y finalmente una evaluación de estos, para así alcanzar los objetivos del proyecto. En la Figura 1 se muestra la estructura metodológica del proyecto.

Figura 1

Metodología general del proyecto



3.1 Preprocesamiento de Datos

Con la ayuda de tablas dinámicas de Excel se categorizó los datos experimentales de la base de datos. Se seleccionaron aquellos datos referentes a la fabricación y el rendimiento de los recubrimientos de perovskita según variables de proceso, composiciones y eficiencia.

Como variables de entrada se tomaron aquellas referentes a procesos y materiales de fabricación de las diferentes capas de las celdas solares de perovskita. De esta forma se obtuvieron 10 columnas con datos categóricos. En cuanto a variable objetivo se tomó la columna de eficiencia de celda de perovskita (PCE) como indicador del rendimiento.

Se implementó un criterio de exclusión con el que eliminaron los datos de celdas construidas que no contaran con información suficiente en las variables, usando como criterio contar con datos en al menos 7 de las 10 columnas de datos de entrada.

De esta forma se obtuvo un conjunto de datos con 11 columnas y 114 filas, con datos de parámetros de fabricación de celdas solares de perovskita con eficiencias entre 4,6% y 20,4% para la respectiva aplicación de algoritmos de ML.

La codificación de datos se llevó a cabo utilizando la librería de scikit-learn por medio de la técnica *label* encoder. Esta asigna valores numéricos a las variables categóricas para permitir su uso efectivo en los diferentes algoritmos de aprendizaje. Se optó las librerías Keras para la implementación del modelo LSTM, y se utilizó scikit-learn para los demás modelos.

3.2 Entrenamiento de Algoritmos

Para abordar los problemas de regresión se seleccionaron diferentes modelos de aprendizaje automático según aplicaciones exitosas anteriores en el área de la ciencia de

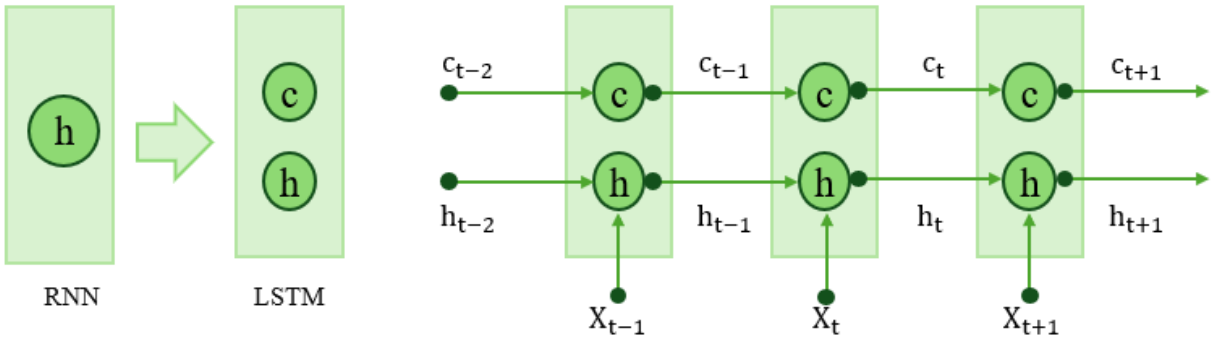
materiales. Se escogieron los modelos LSTM, Bosques Aleatorios, Máquinas de Vectores de Soporte, Gradient Boosting y XGBoost (Ichwani et al., 2023; Odabaşı & Yıldırım, 2020b, 2020a).

3.2.1 LSTM

El algoritmo LSTM incorpora un estado de celda basado en redes neuronales recurrentes (RNN) para la preservación del estado a largo plazo. En la Figura 2se ilustra la entrada a la red LSTM en el tiempo *t*, así como la salida y el estado de la celda de la red LSTM en el tiempo *t−1*. Estos tres datos representan la entrada, salida y estado de la celda de la red LSTM en el tiempo *t*, donde *X*, *h* y *c* son vectores (Wu et al., 2021).

Figura 2

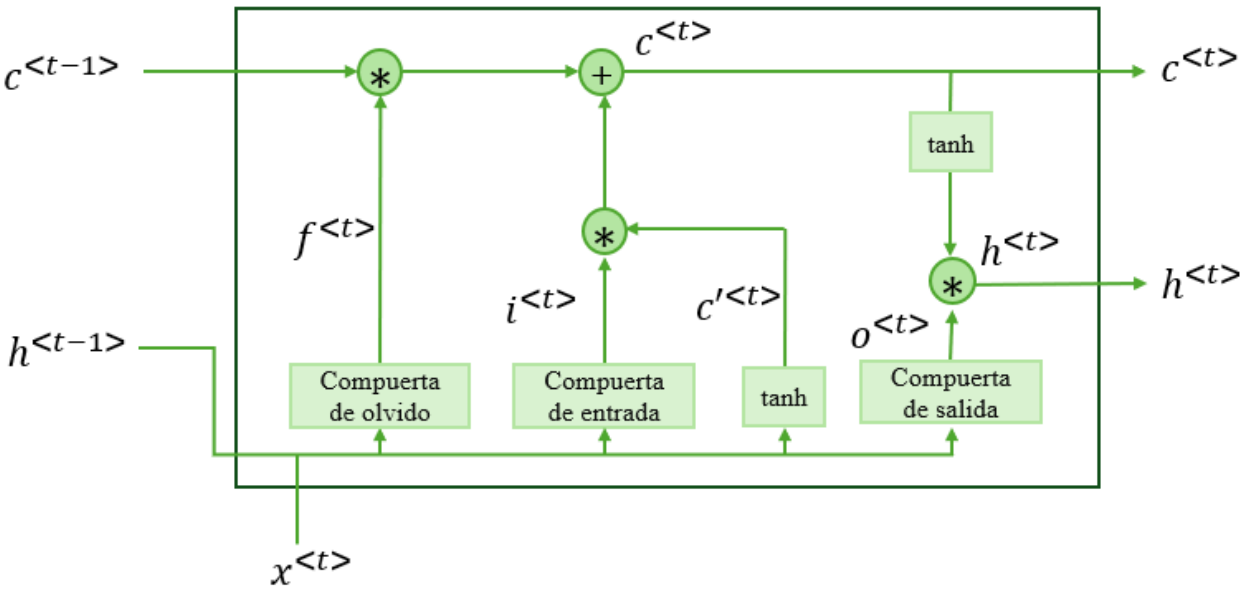
Estructura general LSTM



La implementación de LSTM se realiza a través de la preservación, actualización e incorporación del estado a largo plazo (*c*) mediante la puerta de olvido interna, la puerta de entrada y la puerta de salida, como se muestra en la Figura 3. La puerta de olvido determina la información preservada en el estado de la celda, mientras que la puerta de salida preserva la información de la entrada en el momento *t*. La puerta de salida controla las partes del estado de la celda que se desean incluir en la salida (Wu et al., 2021).

Figura 3

Estructura interna LSTM



Matemáticamente, las compuertas se representan de la siguiente forma:

$$g(x) = \sigma(W_x + b) \quad \textbf{(1)}$$

Donde:

g : Compuerta

σ : Función sigmoide

W : Peso de la compuerta

b : Vector de error

De esta forma, las compuertas de olvido (f), de entrada (i) y de salida (o) están representadas por las ecuaciones (2), (3) y (4) respectivamente.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad \textbf{(2)}$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad \textbf{(3)}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad \textbf{(4)}$$

Donde:

h: Salida del nodo de la capa oculta.

x: Entrada.

Por su parte, la unidad de estado de entrada en un tiempo *t* es calculada según la salida de la red en el tiempo $t-1$ y la entrada en el tiempo t, tal que:

$$c_t' = \tanh\,(W_c[h_{t-1}, x_t] + b_c) \quad \textbf{(5)}$$

Así, es posible obtener el estado de celda para un tiempo *t* utilizando multiplicaciones elemento por elemento:

$$c_t = f_t \circ c_{t-1} + i_t \circ c_t' \quad \textbf{(6)}$$

La salida de la red neuronal de memoria a largo plazo está determinada según la puerta de salida y el estado de celda, tal que:

$$h_t = o_t \circ \tanh\,(c_t) \quad \textbf{(7)}$$

3.2.2 Bosques Aleatorios

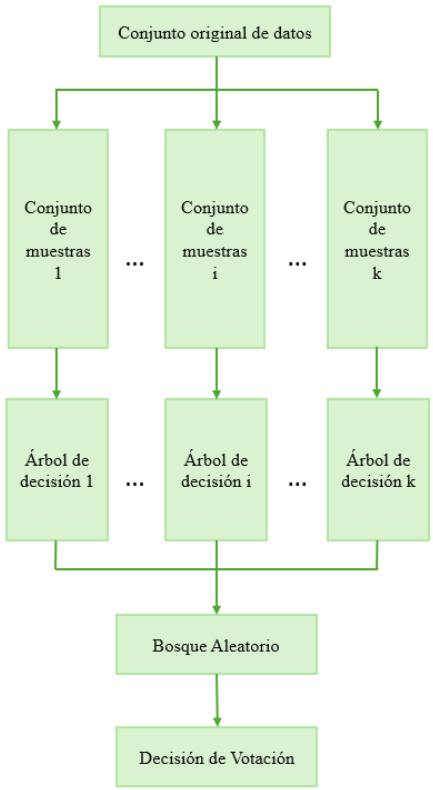
El primer paso para el funcionamiento del algoritmo es la selección de k conjuntos de datos diferentes utilizando un método de muestreo aleatorio. Cada uno de estos conjuntos se utiliza para construir un árbol de decisión que abarca todos los datos de entrenamiento. El número de muestras en cada conjunto es igual al tamaño del conjunto de datos original.

Luego, los k conjuntos se emplean para construir k árboles de decisión sin podar. Para formar los nodos del árbol se seleccionan m atributos, donde m es menor o igual a M (número total de atributos en el conjunto de datos original), como candidatos para los atributos de características. El árbol de decisión se construye utilizando m atributos candidato seleccionados aleatoriamente, y no se lleva a cabo un crecimiento completo ni una poda. Cada uno de los árboles se clasifica como un árbol de decisión completo, generando k resultados de la clasificación.

Finalmente, se utiliza un método de votación. Con base en los k resultados de clasificación obtenidos se realiza una votación para determinar la categoría final de la variable de salida. La categoría final se determina mediante la mayoría de los votos obtenidos entre los k resultados de clasificación. El flujo del modelo se muestra en la Figura 4 (Liu & Wu, 2017).

Figura 4

Flujo del modelo Bosques Aleatorios



3.2.3 SVM

El SVM construye un modelo lineal para estimar la función de decisión utilizando límites de clase no lineales basados en vectores de soporte. Cuando los datos están separados linealmente, el SVM entrena máquinas lineales para encontrar un hiperplano óptimo que separe los datos sin error y maximice la distancia entre el hiperplano y los puntos de entrenamiento más cercanos. Estos puntos más cercanos se llaman vectores de soporte, y los demás ejemplos de entrenamiento son irrelevantes para determinar los límites de clase binarios (Shin et al., 2005).

Definiendo un conjunto de entrenamiento $[x_i, y_i]$, con vector de entrada $x_i \in R^n$, valor de clase $y_i \in \{-1,1\}; i = 1, \dots, l$.

Para el caso linealmente separable, las reglas de decisión definidas por un hiperplano óptimo que separa las clases de decisión binarias se expresan mediante una ecuación en términos de los vectores de soporte, tal que:

$$Y = sign\left(\sum_{i=1}^{N} y_i \alpha_1 (\boldsymbol{x} \cdot \boldsymbol{x_i}) + b\right) \quad \textbf{(8)}$$

Donde:

Y: Salida.

$\boldsymbol{x} = (x_1, x_2, \dots, x_n)$: Vector de entrada.

$\boldsymbol{x_i}, i = 1, \dots, N$: Vectores de soporte.

α_i, b: Parámetros que determinan el hiperplano.

En el caso no linealmente separable, la ecuación se generaliza a una versión de alta dimensión utilizando una función de *kernel*. Así:

$$Y = sign\left(\sum_{i=1}^{N} y_i \alpha_1 K(\boldsymbol{x} \cdot \boldsymbol{x_i}) + b\right) \quad \textbf{(8)}$$

En caso de máquinas polinomiales, la función *kernel* es:

$$K(\boldsymbol{x}, \boldsymbol{x_i}) = (\boldsymbol{x} \cdot \boldsymbol{x_i} + \boldsymbol{1})^{\boldsymbol{d}} \quad \textbf{(9)}$$

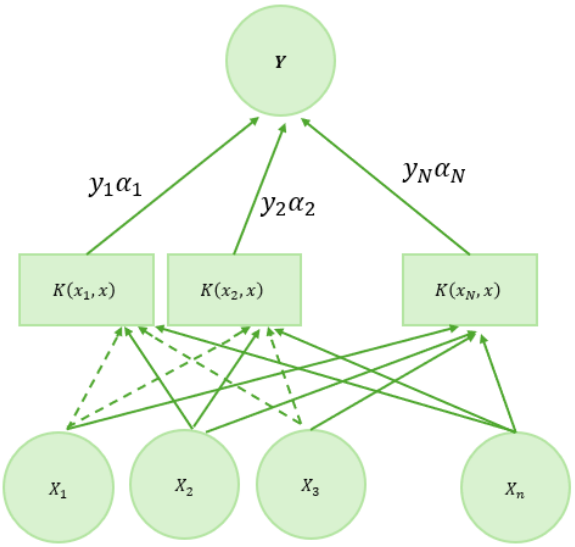
Donde:

d: Grado del *kernel* polinomial.

El proceso de aprendizaje para construir funciones de decisión del SVM se basa en la estructura de redes neuronales de dos capas y utiliza teoría de optimización que minimiza la clasificación errónea basada en la teoría del aprendizaje estadístico. Como se muestra en la Figura 5, la primera capa de las SVM selecciona la base *K(xi,x)*, y el número de vectores de soporte a partir de un conjunto de bases definido para el kernel. La segunda capa construye el hiperplano óptimo en el espacio de características correspondientes (Shin et al., 2005).

Figura 5

Estructura algoritmo SVM



3.2.4 Gradient Boosting

El árbol de regresión en "boosting" construye el modelo de manera incremental y lo actualiza minimizando el valor esperado de cierta función de pérdida. Con muchos árboles agregados al modelo, éste puede lograr un error de entrenamiento arbitrariamente pequeño. Sin embargo, ajustar el modelo demasiado a los datos de entrenamiento puede llevar a una capacidad

de generalización deficiente. Para prevenir el sobreajuste, es necesario determinar el número óptimo de iteraciones (o el número de árboles) M para minimizar los riesgos futuros asociados con la predicción (Wang et al., 2008). El algoritmo general utilizado se define de la siguiente forma:

$$f_m(x) = f_{m-1}(x) + J \cdot \sum_{j=1}^{J} \rho_{jm} I\left(x \in R_{jm}\right) \quad ; \quad I\left(x \in R_{jm}\right) = \begin{cases} 1, si\ x \in R_{jm} \\ 0 \end{cases} \qquad \textbf{(10)}$$

Donde:

ρ: Correlación entre árboles.

R_{jm}: Regiones disjuntas.

La tasa de aprendizaje J controla la contribución de cada modelo base. Existe un equilibrio entre el número de iteraciones y la tasa de aprendizaje. Con el mismo número de iteraciones, un valor más pequeño de la tasa de aprendizaje tiende a conducir a un mayor riesgo de entrenamiento. Esto es, un valor más pequeño de tasa de aprendizaje requiere un mayor número de iteraciones para obtener el mismo riesgo de entrenamiento. En general, es preferible una tasa pequeña con un mayor número de iteraciones (Wang et al., 2008).

Otro parámetro, la complejidad del árbol, también influye en el rendimiento del algoritmo. La complejidad del árbol se refiere al número de nodos en un árbol. El tamaño óptimo de cada árbol puede estimarse por separado al construir los conjuntos. Al asumir simplemente que cada árbol es el último en el modelo, generalmente se esperan árboles muy grandes, especialmente durante las primeras iteraciones, aumentado la complejidad computacional. Por lo tanto, para todo el proceso, se determina un valor de complejidad único para todos los árboles. Las ganancias de incrementar esta complejidad son mayores con conjuntos de datos más grandes, permitiendo la identificación de interacciones complejas (Wang et al., 2008).

3.2.5 XGBoost

El funcionamiento de XGBoost se basa en la construcción de un conjunto de árboles de decisión de forma secuencial. Cada árbol se ajusta para corregir los errores del modelo existente. La salida final del modelo es la suma ponderada de las predicciones de cada árbol.

Para entrenar el modelo, se minimiza una función de pérdida que mide la discrepancia entre las predicciones y las etiquetas reales. Además, se utiliza un término de regularización para evitar que el modelo se vuelva demasiado complejo y se sobreajuste a los datos de entrenamiento (Y. Zhang & Haghani, 2015).

Durante el proceso de construcción de cada árbol, se utiliza un enfoque de aumento (boosting). En cada iteración, se agrega un nuevo árbol para mejorar el rendimiento del modelo. Este árbol se enfoca en corregir las predicciones incorrectas del modelo existente (Y. Zhang & Haghani, 2015).

Además, al dividir un árbol durante su construcción, se utiliza un criterio que considera la mejora en la función de pérdida. Esto asegura que cada división se realice de manera que minimice la pérdida total del modelo (Y. Zhang & Haghani, 2015).

3.3 Afinación de Modelos

Para la afinación y mejoramiento de los modelos de regresión desarrollados se llevó a cabo la búsqueda de parámetros durante su entrenamiento, usando algoritmos de *Random Search* y *Grid Search* integrados en la librería scikit-learn. El primero se usó para exploraciones amplias y computacionalmente más eficientes, mientras que el segundo en refinamientos más precisos, ajustando rangos de manera más.

Se incorporaron gráficas descriptivas y métricas de evaluación a cada algoritmo de predicción. Esto para facilitar la visualización de los resultados obtenidos por cada uno de estos, y permitir una comparación cualitativa y cuantitativa de su rendimiento.

Los detalles y resultados específicos de cada modelo se presentarán de manera detallada en las secciones siguientes.

3.4 Evaluación de Modelos

La comparación del rendimiento de los modelos se hizo con métricas estándar en problemas de regresión, como el error absoluto medio (MAE), el error cuadrático medio (MSE), el coeficiente de determinación (R^2) y el valor p asociado (Willmott, 1981). Este último proporciona una evaluación estadística de la significancia de las métricas, mejorando la confiabilidad de la comparación entre modelos (Wasserstein & Lazar, 2016).

Se incluyó también un análisis del tiempo de ejecución de cada uno de los modelos, permitiendo evaluar la eficiencia computacional y determinar la aplicabilidad de cada algoritmo en problemas similares en el futuro.

Los gráficos comparativos y tablas que presenten el comportamiento de cada modelo se incluirán en las secciones subsiguientes. El enfoque metodológico del proyecto respalda la elección del mejor modelo en cuanto al rendimiento de la predicción, y también considerando aspectos prácticos fundamentales de cada algoritmo implementado en el proyecto.

4. Resultados

La Tabla 1 presenta las combinaciones de variables de entrada para las 4 celdas con mayores valores de eficiencia. Todos tienen en común la aplicación de la perovskita en un solo paso mediante *spin coating*, la presencia de dimetilsulfóxido (DMSO) como parte de la mezcla de precursor de solución y uso de [6,6]-fenil-C61-butilo metano (PCBM) en la capa de transporte de electrones. Además, 3 de 4 combinaciones incluyen el uso de una única capa de transporte de huecos, y presencia de dimetilformamida **(**DMF) en la mezcla de precursor de solución, además de una segunda capa de transporte de electrones de bloques copolímeros (BCP).

Tabla 1

Combinaciones con mejores eficiencias según datos reales

HTL	HTL-2	Perovskite	Deposition procedure	Depositio n method	Anti-solvent threatment	Precursor solution	ETL	ETL-2	Back contact
PTAA	no	FA0.83MA 0.17Pb(Br0. 4I0.6)3\xa0	one-step	spin	toluene	DMF+ DMSO	PCBM+ C60	BCP	Cu
NiOx-Cu	mNiOx-Cu	MAPbI3	one-step	spin	chlorobenzene	DMF+ DMSO	PCBM	no	Ag
PTAA	no	FA0.83MA 0.17Pb(Br0. 4I0.6)3\xa0	one-step	spin	toluene	DMF+ DMSO	PCBM+ C60	BCP	Cu
NiMgL iO	no	Cs0.05FA0. 15MA0.8Pb I3	one-step	spin 2-3	chlorobenzene	DMSO+ GBL	PCBM	BCP	Ag

4.1 LSTM

Una vez procesados los datos se entrenó el modelo. Los parámetros iniciales se escogieron arbitrariamente, y con la ayuda de los algoritmos *Random Search* y *Grid Search* integrados en la librería scikit-learn se afinaron sus rangos. Se optó por los parámetros más conservativos en cuanto a requerimientos computacionales que mantuvieran un rango de calidad aceptable, según un coeficiente de determinación (R^2) no menor al 95% del valor máximo alcanzado en la búsqueda. La Tabla 2 muestra los parámetros obtenidos para el modelo LSTM.

Tabla 2

Hiperparámetros del modelo LSTM

Parámetro	Valor
Unidades LSTM 1	200
Unidades LSTM 2	200
Optimizador	Adam
Función de pérdida	Mean squared error
Épocas	1000
Tamaño de lote	23

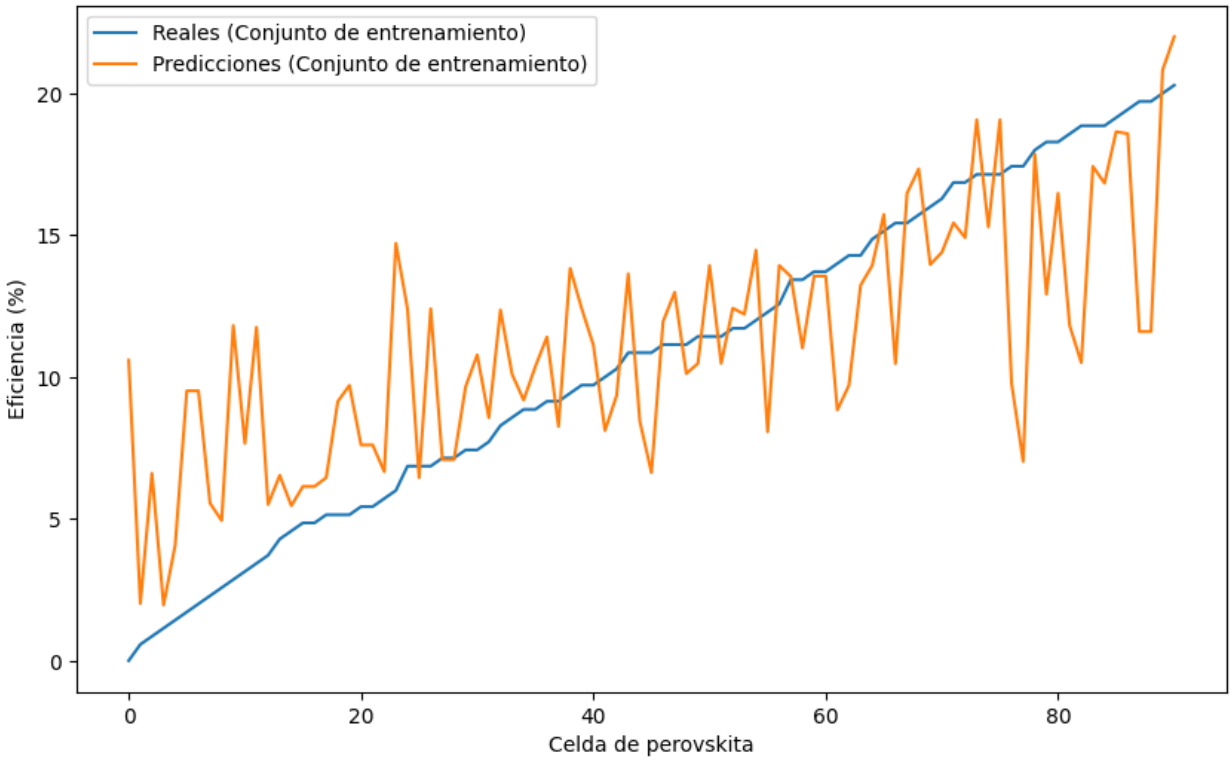
Se definieron dos capas LSTM con 200 unidades cada una, y utilizando método de optimización Adam y error medio cuadrado como función de pérdida. Se especificaron además 1000 iteraciones, con un entrenamiento de lote completo, tomando el tamaño de lote como el mismo tamaño del conjunto de entrenamiento.

La Figura 6 muestra el comportamiento de la eficiencia predicha por el modelo LSTM en el conjunto de entrenamiento para cada combinación de variables y sus valores reales. Esta muestra

un error considerable a nivel general, además de fallas importantes en la capacidad de predecir el comportamiento de la eficiencia para diferentes configuraciones de celdas de perovskita.

Figura 6

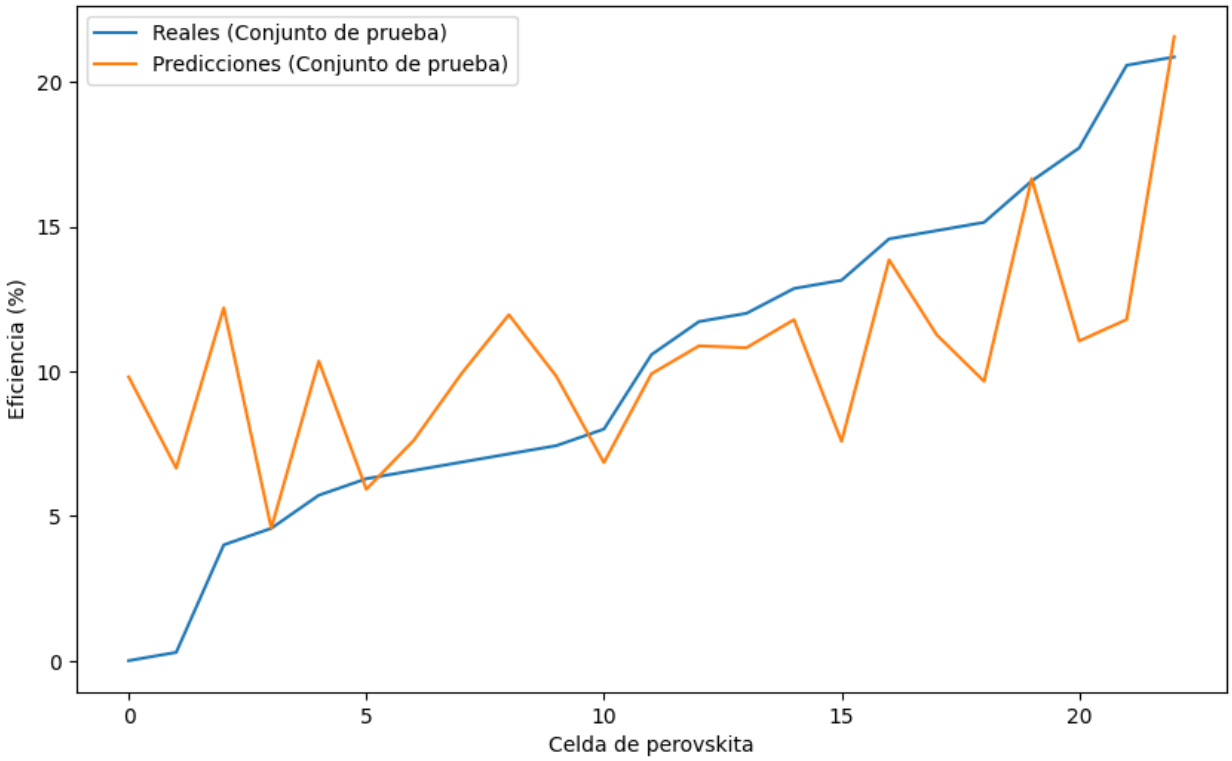
Eficiencias de celdas de perovskita reales y predichas en conjunto de entrenamiento (LSTM)



La Figura 7 muestra razgos similares en el comportamiento a lo largo del conjunto de datos de prueba. La diferencia entre las eficiencias reales y obtenidas por el modelo es apreciable a simple vista, y las excepciones a esta regla son relativamente pocas.

Figura 7

Eficiencias de celdas de perovskita reales y predichas en conjunto de pruebas (LSTM)



En cuanto a la obtención de variables optimas de manufactura, en la Tabla 3 se muestra que el modelo LSTM encontró que las mejores opciones cuentan siempre con una única capa de transporte de huecos (HTL) y un solo paso de deposición de perovskita (Deposition procedure) mediante el método *spin coating*. Como precursor de solución, la DMF y el DMSO en tándem aparecen como excelente opción. Como tratamiento anti solvente, parece tener inclinación hacia el tolueno. Para el transporte de electrones, presenta siempre doble capa, la primera con PCBM o fullereno (C60), usados independientemente o en tándem, mientras que en la segunda capa siempre

propone el uso de BCP. Para el electrodo negativo sugiere uso de cobre (Cu) o plata (Ag). Los resultados de composición de capa de perovskita no son convergentes.

Tabla 3

Predicción de mejores combinaciones de variables (LSTM)

HTL	HTL-2	Perovskite	Deposition procedure	Deposition method	Anti-solvent threatment	Precursor solution	ETL	ETL-2	Back contact
PTAA	no	FA0.83MA0.17 Pb(Br0.4I0.6) 3\xa0	one-step	spin	toluene	DMF+ DMSO	PCBM +C60	BCP	Cu
PTAA	no	FA0.83MA0.17 Pb(Br0.4I0.6) 3\xa0	one-step	spin	toluene	DMF+ DMSO	PCBM +C60	BCP	Cu
NiMgLiO	no	Cs0.05FA0.15 MA0.8PbI3	one-step	spin 2-3	chloro-benzene	DMSO+ GBL	PCBM	BCP	Ag
polyTPD	no	MAPbI3	one-step	spin	toluene	DMF+ DMSO	C60	BCP	Ag

Las métricas de evaluación del modelo LSTM se consignaron en la Tabla 4. El coeficiente de determinación de 0,356863 indica una capacidad moderada para explicar la variabilidad, pero aún hay una cantidad significativa de esta que no está siendo capturada por el modelo. El error cuadrático medio relativamente alto (259,84443) sugiere discrepancias sustanciales entre las predicciones y los valores reales. Se presenta por último el tiempo total de ejecución del modelo de alrededor de 57 segundos. En este caso específico resulta un modelo impráctico, cuyo desempeño mediocre no justifica su largo tiempo de ejecución. Sin embargo, no por eso debe ser descartado su uso en problemáticas de ciencia de materiales, pues su carácter temporal es útil al momento de análisis de variables que dependen del tiempo, como lo puede ser en el estudio de fenómenos de fatiga (Heng et al., 2023).

Tabla 4

Métricas de evaluación (LSTM)

Métrica	Valor
R^2	0,356863
MSE	259,84443
P value	0,002603
Time	57,559218

4.2 Bosques Aleatorios

La Tabla 5 muestra los parámetros obtenidos para el modelo de bosques aleatorios. Se definieron 31 árboles y el uso del criterio de error cuadrado medio durante su construcción. No se limitó la profundidad máxima de los árboles, y se definió un número mínimo de muestras para división de 2 y número de muestras para nodo hoja de 1.

Tabla 5

Hiperparámetros del modelo Bosques Aleatorios

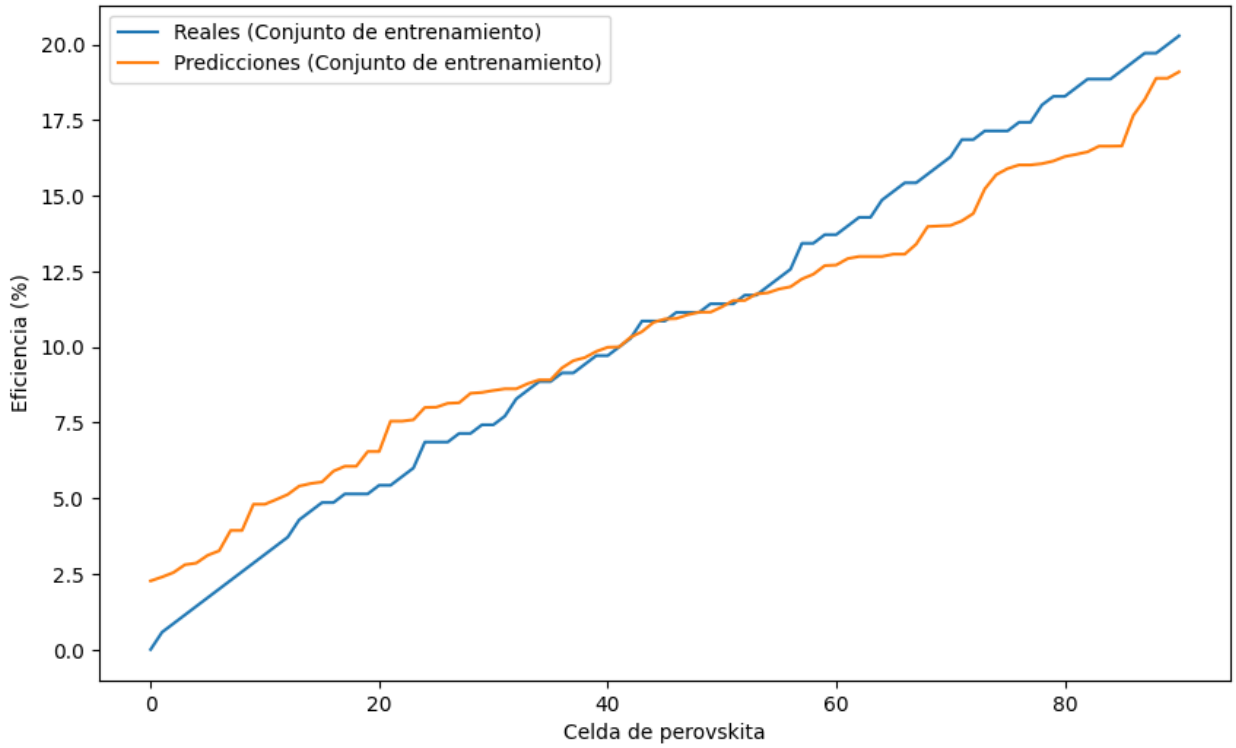
Parámetro	Valor
# Estimators	31
Criterion	Squared error
Random State	42
Max depth	None
Min. Samples split	2
Min. Samples leaf	1

La Figura 8 muestra la predicción de eficiencia del modelo de bosques aleatorios en el conjunto de entrenamiento para cada celda y sus valores reales. Se observa que las predicciones logran ajustarse a la tendencia de los datos de entrenamiento. Además, los valores de eficiencias

predichos no se alejan mucho de los valores reales, con especial acercamiento a estos últimos para los valores de eficiencia medios (entre 8% y 11%)

Figura 8

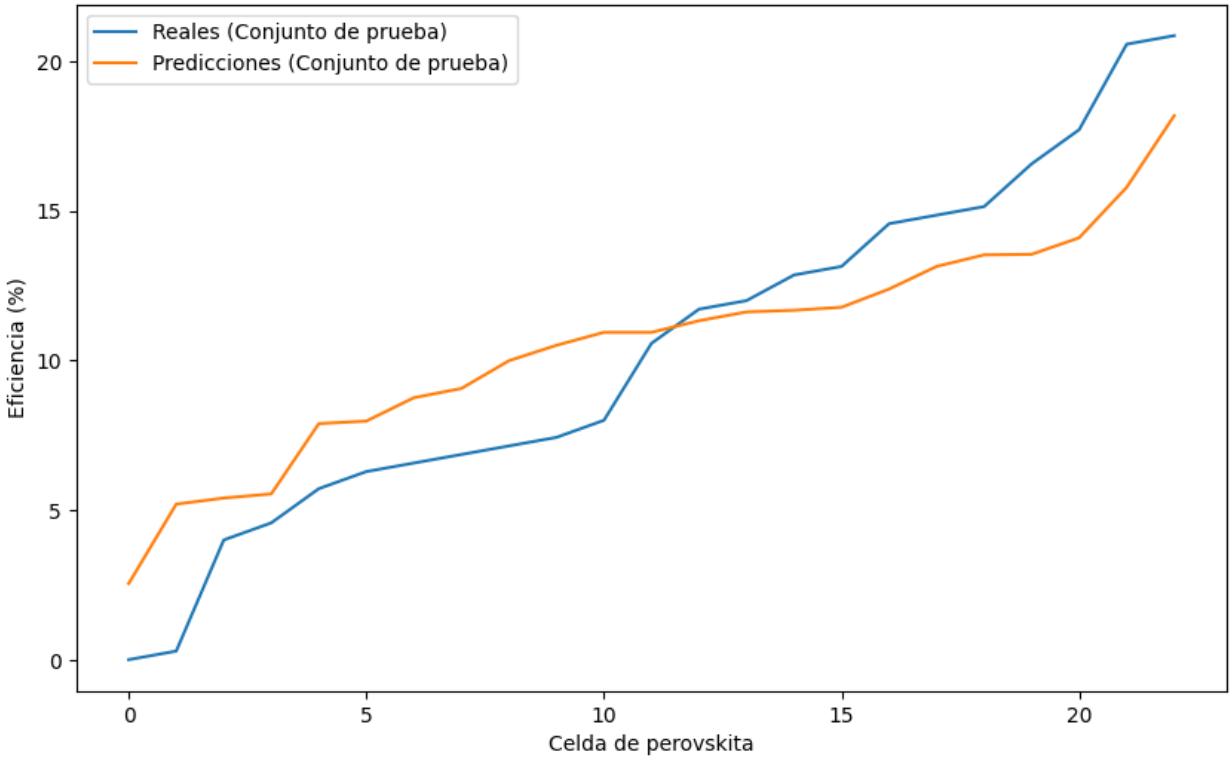
Eficiencias de celdas de perovskita reales y predichas en conjunto de entrenamiento (Bosques Aleatorios)



La Figura 9 muestra la comparación de las predicciones del modelo de bosques aleatorios con los datos reales en el conjunto de prueba. La línea de datos predichos muestra una tendencia cercana a la de los datos reales. Sin embargo, muestra desfases relativamente altos en los valores obtenidos respecto a los datos reales.

Figura 9

Eficiencias de celdas de perovskita reales y predichas en conjunto de pruebas (Bosques Aleatorios)



La Tabla 6 muestra las predicciones de las celdas con los mejores resultados de eficiencia según el modelo de bosques aleatorios. La mayor convergencia se ve en el uso de una sola capa de transporte de huecos, deposición de la perovskita en un solo paso mediante método de *spin coating*, el uso de DMSO y DMF como precursores de solución, y la presencia de PCBM en la primera capa de transporte de electrones. Para el electrodo negativo toma el uso de plata o cobre, y se presenta el tolueno como común para tratamiento anti solvente.

Tabla 6

Predicción de mejores combinaciones de variables (Bosques Aleatorios)

HTL	HTL-2	Perovskite	Deposition procedure	Deposition method	Anti-solvent threatment	Precursor solution	ETL	ETL-2	Back contact
NiMgLiO	no	Cs0.05FA0.15MA0.8PbI3	one-step	spin 2-3	chloro-benzene	DMSO+GBL	PCBM	BCP	Ag
NiOx	no	MAPbI3	one-step	spin 2-3	toluene	DMF+DMSO	PCBM	no	Ag
NiOx	no	MAPbI3	one-step	spin 2-3	toluene	DMF+DMSO	PCBM	no	Ag
PTAA	no	FA0.83MA0.17Pb (Br0.4I0.6) 3\xa0	one-step	spin	toluene	DMF+DMSO	PCBM+C60	BCP	Cu

La Tabla 7 muestra las métricas de evaluación obtenidas para este modelo. El modelo presenta un coeficiente de determinación R^2 de 0.39, ligeramente superior al LSTM, pero aún con un MSE considerable (246,07776). Muestra cierta capacidad para explicar la variabilidad, pero aún hay una cantidad significativa de esta que no está siendo capturada por el modelo. Se obtuvo un MSE alto, mostrando una elevada varianza. El tiempo total de ejecución del modelo fue de alrededor de medio segundo. Este modelo demuestra una buena eficiencia en términos de tiempo de ejecución, y buena capacidad de predicción en cuanto a la tendencia de la eficiencia de las celdas según sus parámetros de fabricación. Sin embargo, en cuanto a la cercanía de los valores de eficiencia presentados no alcanza niveles muy satisfactorios.

Tabla 7

Métricas de evaluación (Bosques Aleatorios)

Métrica	Valor
R^2	0,390937
MSE	246,07776
P value	0,001393
Time	0,543135

4.3 Máquinas de Vectores de Soporte (SVM)

Los parámetros del modelo SVM se muestran en la Tabla 8. Se escogió un parámetro de decisión C de 1 para operar en un *kernel* polinomial de grado 3. Además, se escogió un épsilon o error permitido de 0,2 con tolerancia de 0,01, y no se limitaron las iteraciones.

Tabla 8

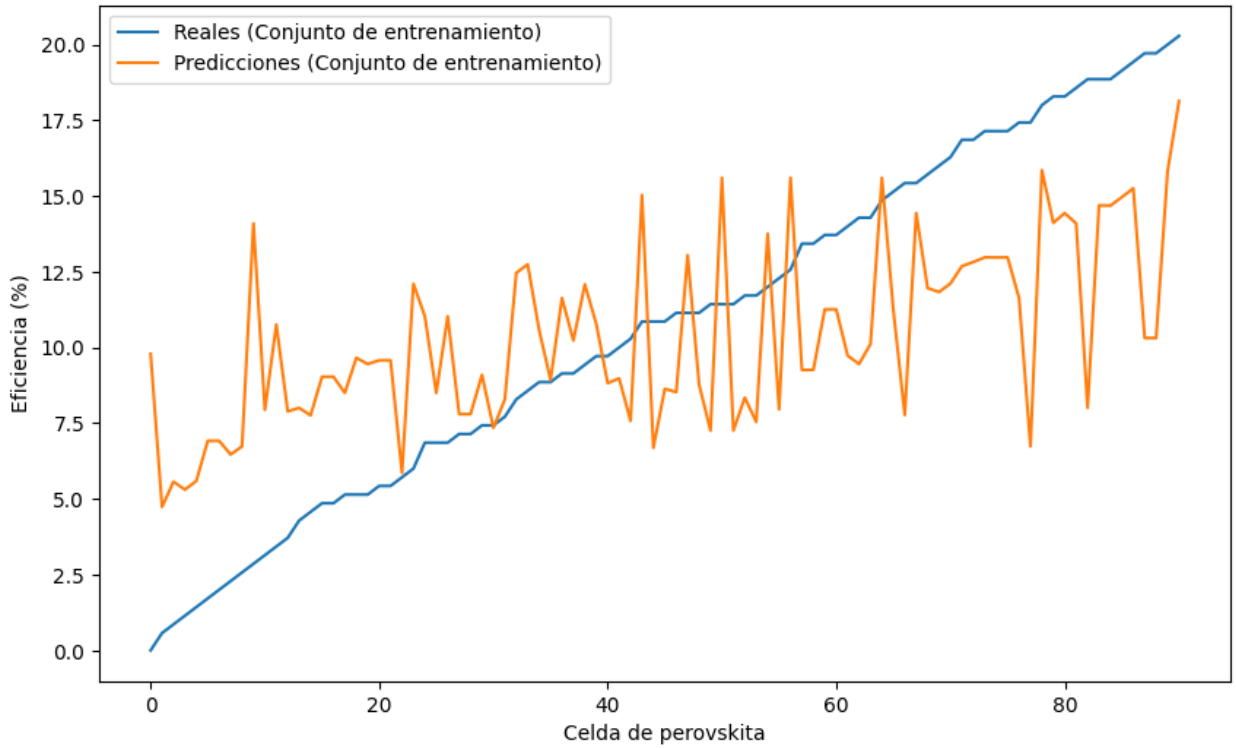
Hiperparámetros del modelo SVM

Parámetro	Valor
C	1
Kernel	Poly
Degree	3
Epsilon	0,2
Tolerance	0,01
Max iter.	-1

La comparación de la predicción de eficiencias del modelo SVM con las eficiencias reales vistas en la Figura 10 no permite ver un ajuste confiable del modelo al comportamiento de los datos del conjunto de entrenamiento. Se aprecian un comportamiento caótico, sin ninguna tendencia aparente.

Figura 10

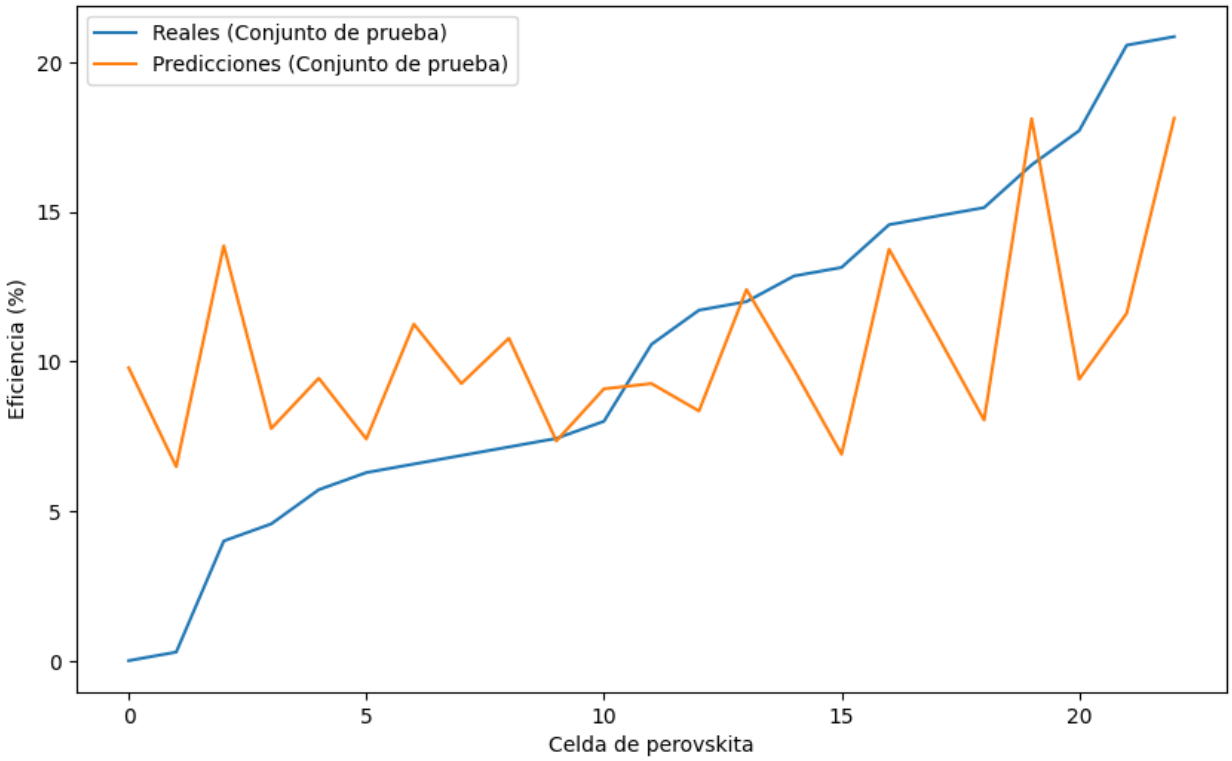
Eficiencias de celdas de perovskita reales y predichas en conjunto de entrenamiento (SVM)



Al observar el comportamiento del modelo en el dataset de prueba en la Figura 11 se evidencia incapacidad del modelo de captar la tendencia de los datos. No se obtuvo una predicción que permita acercarse al comportamiento de la eficiencia en las celdas evaluadas.

Figura 11

Eficiencias de celdas de perovskita reales y predichas en conjunto de prueba (SVM)



Respecto a la predicción de mejores combinaciones de variables para las celdas, el modelo SVM en la Tabla 9 muestra inclinación hacia celdas de una sola capa de transporte de huecos que usan Poly[bis(4-phenyl)(2,4,6-trimethylphenyl)amine] (PCAA) y procesos físicos de deposición de perovskita en un paso, sin uso de tratamiento anti solvente. Como precursor de solución presenta gamma-butirolactona (GBL) y DMF como buenas opciones, junto a capas de transporte de electrones compuestas de PCBM, C60 y BCP. Para el electrodo negativo presenta el cobre.

Tabla 9

Predicción de mejores combinaciones de variables (SVM)

HTL	HTL-2	Perovskite	Deposition procedure	Deposition method	Anti-solvent threatment	Precursor solution	ETL	ETL-2	Back contact
NiOx	mNiOx-Li	Cs0.05 (FA0.83MA0.17Pb (I0.83Br0.17))0.95	one-step	spin 2-3	chloro-benzene	DMF+DMSO	IZO	no	Al
PTAA	no	MAPbI3	one-step	spin	no	GBL	PCBM+C60	BCP	Cu
PTAA	no	MAPbI3	one-step	spin	no	GBL	PCBM+C60	BCP	Cu
PTAA	no	FA0.4MA0.6PbI3	one-step	doctor blade	no	DMF	ICBA+C60	BCP	Cu

La Tabla 10 muestra las métricas de evaluación del modelo SVM. El coeficiente de determinación muestra que el modelo tiene un ajuste muy limitado a los datos reales, con baja capacidad para modelar su variabilidad, lo que podría ser atribuido a la elección del kernel lineal. No obstante, en pruebas anteriores se entrenó el modelo con diferentes parámetros, y el uso de kernel polinomiales no mostró una mejora considerable en su coeficiente de determinación, pero sí incrementó su tiempo de ejecución en valores de hasta 300%. Presenta un error cuadrático bastante elevado para el rango de eficiencias manejado, mostrando baja confiabilidad. El tiempo total de ejecución fue de alrededor de 0.8 segundos. Se asume que el modelo tiene problemas para ajustarse a los datos específicos utilizados en este proyecto.

Tabla 10

Métricas de evaluación (SVM)

Métrica	Valor
R^2	0,110562
MSE	359,35638
P value	0,079911
Time	0,777897

4.4 Gradient Boosting

La Tabla 11 muestra los parámetros escogidos para el modelo Gradient Boosting. Se escogieron 17500 árboles con tasa de aprendizaje de 0,01. Se definió una profundidad máxima de 3, un número mínimo de muestras para división de 2 y número de muestras para nodo hoja de 1, con una submuestra de filas de 1.

Tabla 11

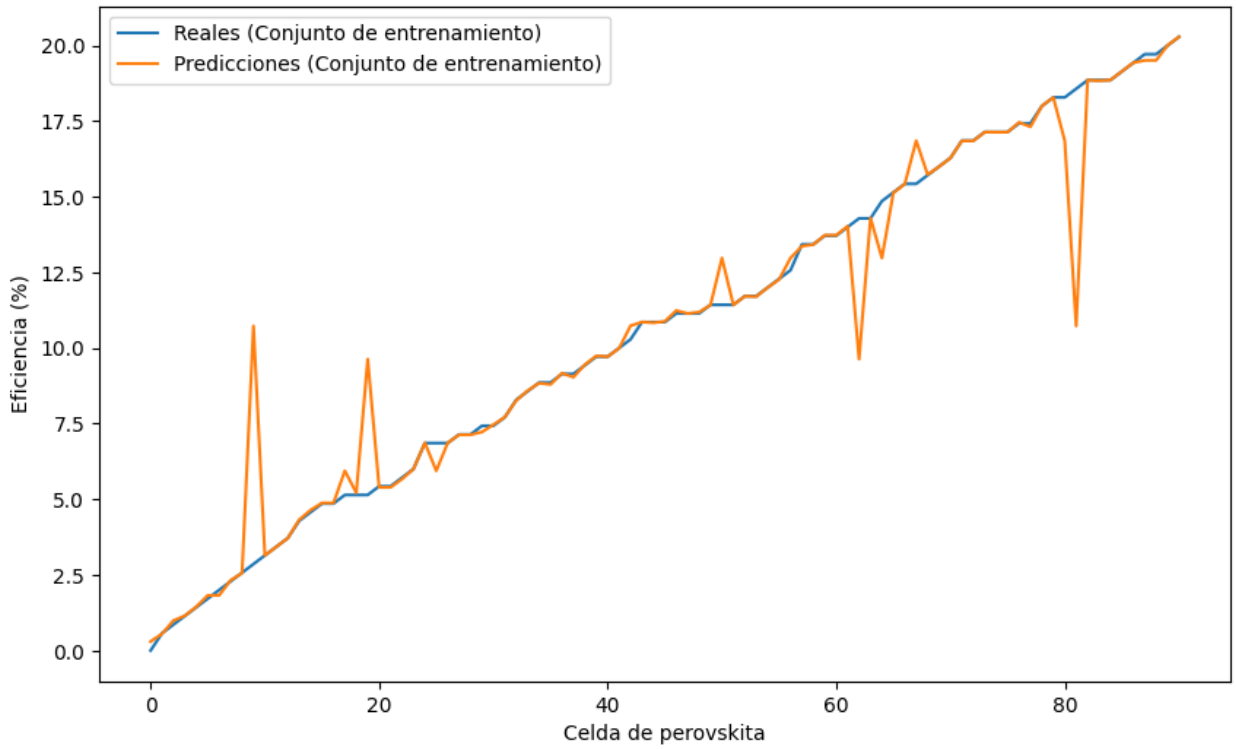
Hiperparámetros del modelo Gradient Boosting

Parámetro	Valor
# Estimators	17500
Learning rate	0,01
Random State	42
Max depth	3
Min. Samples split	2
Min. Samples leaf	1
Subsample	1

En la Figura 12 se aprecia la predicción de eficiencia del modelo de gradient boosting en el conjunto de entrenamiento para cada celda y sus valores reales. Se observa sobreajuste durante el entrenamiento del modelo, con la mayoría de los valores de predicción demasiado cercanos a los datos reales. Debido a ello, en este caso las gráficas presentan tendencias similares.

Figura 12

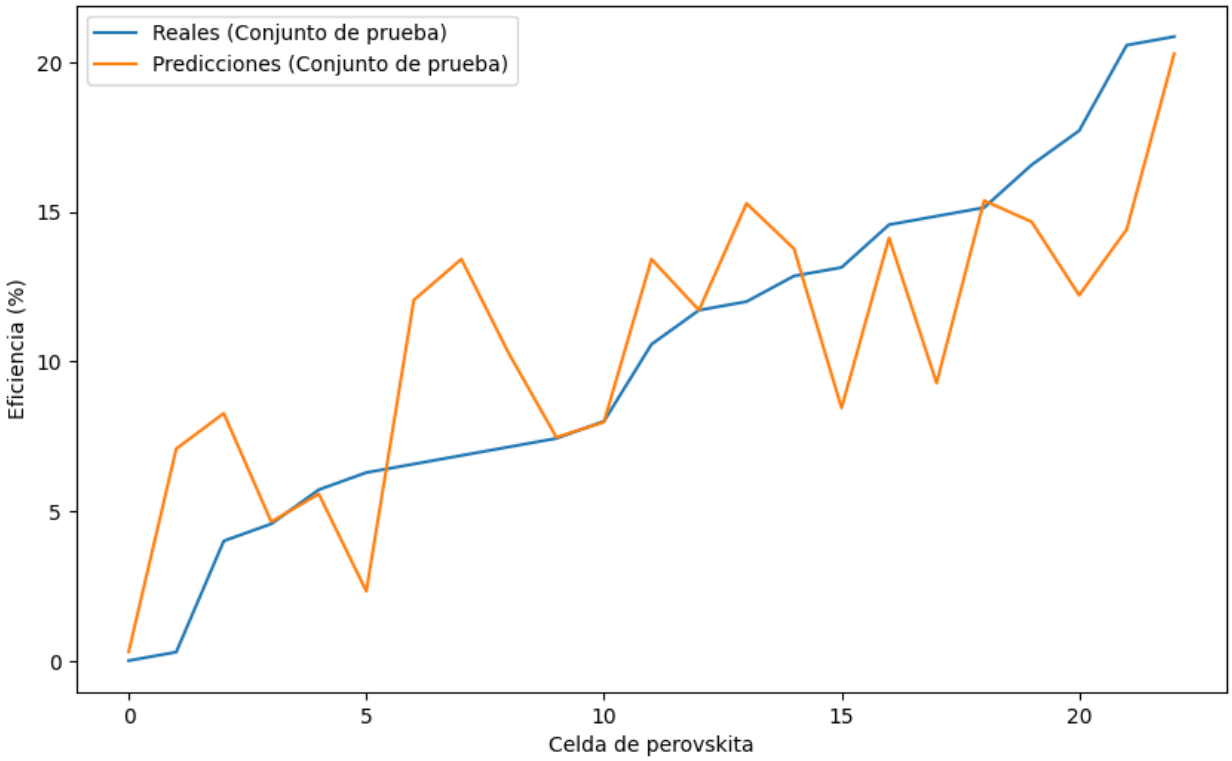
Eficiencias de celdas de perovskita reales y predichas en conjunto de entrenamiento (Gradient Boosting)



La Figura 13 muestra la comparación de las predicciones del modelo de gradient boosting con los datos reales en el conjunto de prueba. Las eficiencias predichas no muestran una tendencia clara, además de presentar desfases bastante claros en muchas de las celdas evaluadas.

Figura 13

Eficiencias de celdas de perovskita reales y predichas en conjunto de prueba (Gradient Boosting)



La predicción del modelo gradient boosting de mejores combinaciones de variables para las celdas se muestra en la Tabla 12. Esta presenta como variables comunes el uso de una única capa de transporte de huecos, deposición de perovskita por método *spin coating* en un solo paso, uso de tolueno para tratamiento anti solvente, presencia de DMF y DMSO como precursores de solución, y uso de PCBM, C60 y BCP en sus respectivas capas de transporte de electrones.

Tabla 12

Predicción de mejores combinaciones de variables (Gradient Boosting)

HTL	HTL-2	Perovskite	Deposition procedure	Deposition method	Anti-solvent threatment	Precursor solution	ETL	ETL-2	Back contact
PTAA	no	FA0.83MA0.17 Pb(Br0.4I0.6) 3\xa0	one-step	spin	toluene	DMF+ DMSO	PCBM+ C60	BCP	Cu
PTAA	no	FA0.83MA0.17 Pb(Br0.4I0.6) 3\xa0	one-step	spin	toluene	DMF+ DMSO	PCBM+ C60	BCP	Cu
niMgLiO	no	Cs0.05FA0.15 MA0.8PbI3	one-step	spin 2-3	chloro-benzene	DMSO+ GBL	PCBM	BCP	Ag
NiOx	no	MAPbI3	one-step	spin 2-3	toluene	DMF+ DMSO	PCBM	no	Ag

En la Tabla 13 se muestran las métricas de evaluación del modelo gradient boosting. El coeficiente de determinación de 0,592 sugiere una capacidad de capturar una cantidad sustancial de la variabilidad en los datos, especialmente teniendo en cuenta el carácter complejo del conjunto de datos de entrada. Sin embargo, este valor se ve incrementado debido a la predicción de eficiencias muy cercanas a las reales causadas por sobreajuste durante el entrenamiento del modelo, afectando la veracidad de esta métrica. El valor del error cuadrático indica que aún existen discrepancias considerables en la regresión. El tiempo total de ejecución fue de alrededor de 10 segundos.

Tabla 13

Métricas de evaluación (Gradient Boosting)

Métrica	Valor
R^2	0,592209
MSE	164,75838
P value	0,000016
Time	10,36

4.5 XGBoost

La Tabla 14 consigna los parámetros escogidos para el modelo XGBoost. Se escogieron 30000 árboles con tasa de aprendizaje de 0,001. Se definió una submuestra de filas de 7, con objetivo de regresión usando error cuadrado como medida de evaluación.

Tabla 14

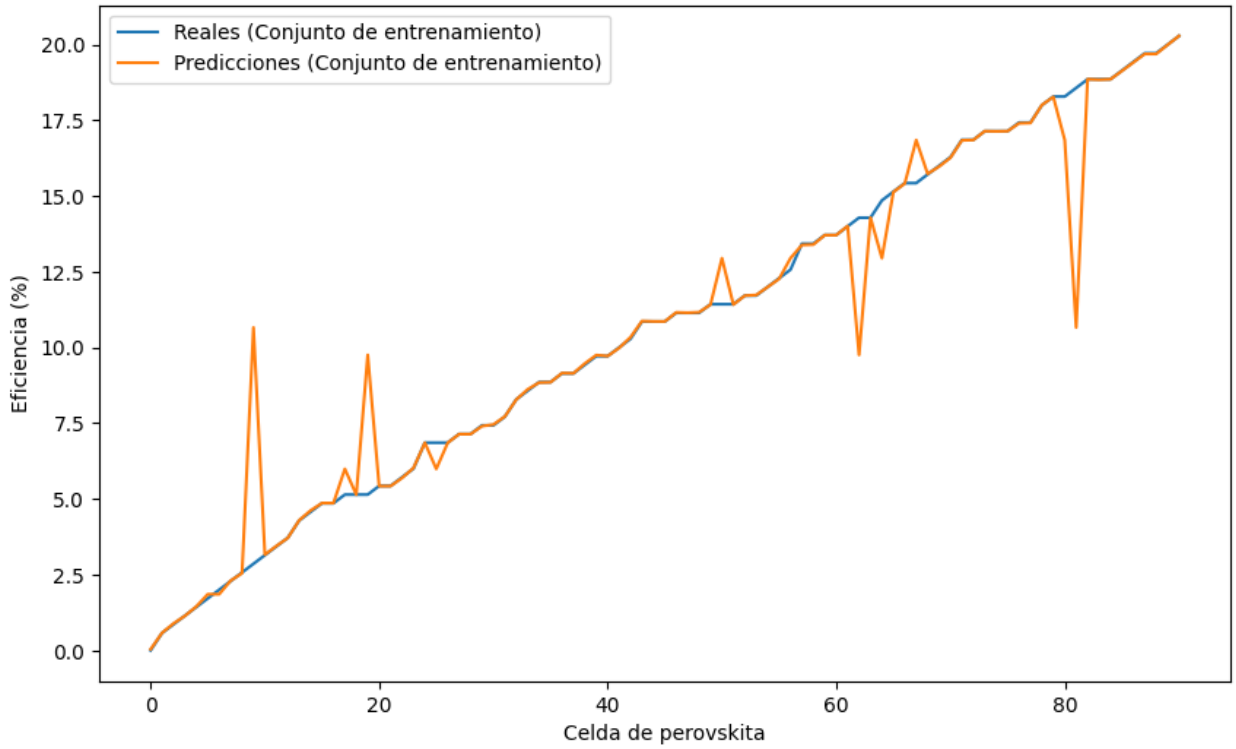
Hiperparámetros del modelo XGBoost

Parámetro	Valor
# estimators	30000
Learning rate	0,001
Max. Depth	7
Subsample	0,9
Objective	reg: squarederror

En la Figura 14 se muestra la predicción de eficiencia del modelo de XGBoost en el conjunto de entrenamiento para cada celda y sus valores reales. Presenta sobreajuste durante el entrenamiento del modelo, con valores de predicción y tendencias muy cercanos a los datos reales.

Figura 14

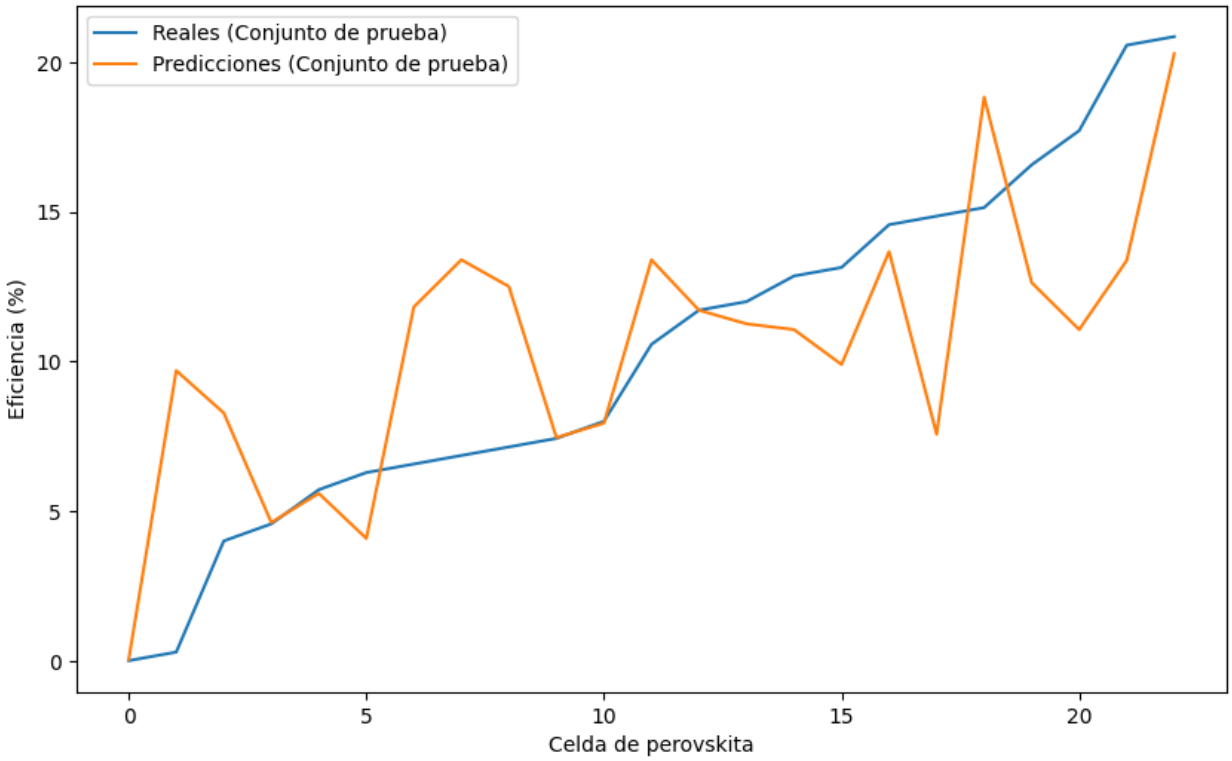
Eficiencias de celdas de perovskita reales y predichas en conjunto de entrenamiento (XGBoost)



La comparación de las predicciones del modelo XGBoost con los datos reales en el conjunto de prueba se muestra en la Figura 15. Las eficiencias predichas no muestran una tendencia clara, además de presentar desfases bastante claros en muchas de las celdas evaluadas.

Figura 15

Eficiencias de celdas de perovskita reales y predichas en conjunto de prueba (XGBoost)



La Tabla 15 muestra las predicciones de las celdas con los mejores resultados de eficiencia según el modelo de XGBoost. El modelo presenta convergencia en uso de una sola capa de transporte de huecos, especialmente polietilenglicol tereftalato/polietilenglicol (PEDOT:PSS), estructura de perovskita MAPbI3 depositada mediante método de *spin coating*, el uso DMF como precursor de solución, la presencia de PCBM en la primera capa de transporte de electrones y uso de plata para la capa de electrodo negativo.

Tabla 15

Predicción de mejores combinaciones de variables (XGBoost)

HTL	HTL-2	Perovskite	Deposition procedure	Deposition method	Anti-solvent threatment	Precursor solution	ETL	ETL-2	Back contact
NiOx	no	MAPbI3	one-step	spin 2-3	diethyl ether	DMF+ DMSO	PCBM	no	Ag
PEDOT: PSS	no	MAPbI3	one-step	spin	no	DMF+ CHP	PCBM	BCP	Ag
PEDOT: PSS	no	MAPbI3	two-step	spin-dip	no	DMF	PCBM	BCP	Ag
PEDOT: PSS	no	MAPbI3	one-step	spin 2-3	toluene	DMSO	PCBM	no	Ag-Al

Las métricas de evaluación del modelo XGBoost se muestran en la Tabla 16. El coeficiente de determinación indica capacidad relativa del modelo para explicar la variabilidad, pero con fallas significativas, que además son mitigadas por la predicción de eficiencias muy cercanas a las reales causadas por sobreajuste durante el entrenamiento del modelo, afectando la veracidad de esta métrica de evaluación de manera similar a lo ocurrido con el modelo Gradient Boosting. El error medio cuadrático indica una varianza apreciable. El tiempo total de ejecución del modelo fue de alrededor de 15 segundos.

Tabla 16

Métricas de evaluación (XGBoost)

Métrica	Valor
R^2	0,456223
MSE	219,7004
P value	0,000339
Time	15,3299

4.6 Comparación de Métricas de Evaluación

La Tabla 17 muestra la comparación de las métricas de evaluación de los diferentes modelos entrenados. El mayor coeficiente de determinación corresponde al modelo de gradient boosting, presentando un ajuste a los datos analizados considerablemente superior al resto de modelos. Por otra parte, el desempeño del modelo SVM resultó bastante pobre, con R^2 cercano al 10%, lo que muestra grandes fallas en la percepción de la variabilidad de los datos.

En cuanto a tiempos de ejecución, el modelo LSTM presentó el mayor tiempo de ejecución entre todos los modelos, llegando a casi un minuto para su ejecución completa. El tiempo más bajo fue el del modelo de bosques aleatorios, con cerca de medio segundo para su ejecución.

Tabla 17

Comparación de métricas entre los modelos

Modelo	R2	P Value	Tiempo de ejecución (s)
LSTM	0,356863	0,002603	57,559218
Bosques Aleatorios	0,390937	0,001393	0,543135
SVM	0,110562	0,079911	0,777897
Gradient Boosting	0,592209	0,000016	10,36
XGBoost	0,456223	0,000339	15,3299

Las métricas favorecen la elección del modelo gradient boosting como mejor opción para la predicción del desempeño de perovskitas con el dataset empleado, pero evaluando el comportamiento de las gráficas resulta más acertado escoger el modelo de bosques aleatorios como el más acertado.

En general, la predicción de mejores combinaciones de variables según sus eficiencias presentó problemas. Esta tarea resulta muy compleja y dependiente de múltiples factores

interrelacionados (Odabaşı & Yıldırım, 2020b). El único modelo que acertó a alguna de las mejores combinaciones reales fue el gradient boosting, que solo acertó a una de estas. Estos resultados podrían mejorarse con el uso de bases de datos más extensas, pero esta solución podría no ser muy viable, pues la limitación en volúmenes de datos disponibles es uno de los grandes problemas recurrentes en la ciencia de materiales (Morgan & Jacobs, 2020).

5. Conclusiones

Se logró una evaluación exhaustiva de diferentes celdas solares de perovskita. Este enfoque permitió analizar datos complejos de tipo categórico y descubrir patrones que podrían pasar desapercibidos mediante métodos de regresión convencionales.

Se implementaron varios modelos de inteligencia artificial LSTM, Bosques Aleatorios, SVM, Gradient Boosting y XGBoost para predecir propiedades y eficiencias de los recubrimientos. Cada modelo demostró capacidades únicas en la interpretación de datos, destacando el modelo de Bosques Aleatorios como el más efectivo en la predicción de la tendencia de los datos para el dataset empleado, pero el coeficiente de determinación cercano a 0.4 indica que debería ser optimizado para ser utilizado de forma confiable.

A pesar de la capacidad de los modelos para predecir eficiencias, la predicción de las mejores combinaciones de variables presentó desafíos. Esto podría solucionarse empleando modelos más robustos que permitan mejores resultados de regresión con volúmenes de datos pequeños, o emplear conjuntos de datos de mayor volumen.

En suma, las recomendaciones generadas por estos modelos podrían contribuir a la mejora continua de la eficiencia de las celdas solares, además de poder aplicarse a otras variables objetivo de interés, impulsando así la adopción de tecnologías más eficientes y sostenibles. Por ello, se evidenció un impacto potencial significativo en la industria energética solar.

Referencias

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Harth, M., Vesce, L., Kouroudis, I., Stefanelli, M., Di Carlo, A., & Gagliardi, A. (2023). Optoelectronic perovskite film characterization via machine vision. *Solar Energy*, *262*. https://doi.org/10.1016/j.solener.2023.111840

Heng, F., Gao, J., Xu, R., Yang, H., Cheng, Q., & Liu, Y. (2023). Multiaxial fatigue life prediction for various metallic materials based on the hybrid CNN-LSTM neural network. *Fatigue & Fracture of Engineering Materials & Structures*, *46*(5), 1979–1996. https://doi.org/https://doi.org/10.1111/ffe.13977

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ichwani, R., Price, S., Oyewole, O. K., Neamtu, R., & Soboyejo, W. O. (2023). Using machine learning for prediction of spray coated perovskite solar cells efficiency: From experimental to theoretical models. *Materials and Design*, *233*.

https://doi.org/10.1016/j.matdes.2023.112161

Jacobsson, T. J., Hultqvist, A., García-Fernández, A., Anand, A., Al-Ashouri, A., Hagfeldt, A., Crovetto, A., Abate, A., Ricciardulli, A. G., Vijayan, A., Kulkarni, A., Anderson, A. Y., Darwich, B. P., Yang, B., Coles, B. L., Perini, C. A. R., Rehermann, C., Ramirez, D., Fairen-Jimenez, D., … Unger, E. (2022). An open-access database and analysis tool for perovskite solar cells based on the FAIR data principles. *Nature Energy*, *7*(1), 107–115. https://doi.org/10.1038/s41560-021-00941-3

Jiang, Y., Li, X., Luo, H., Yin, S., & Kaynak, O. (2022). Quo vadis artificial intelligence? *Discover Artificial Intelligence*, *2*(1), 4. https://doi.org/10.1007/s44163-022-00022-8

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Kojima, A., Teshima, K., Shirai, Y., & Miyasaka, T. (2009). Organometal Halide Perovskites as Visible-Light Sensitizers for Photovoltaic Cells. *Journal of the American Chemical Society*, *131*(17), 6050–6051. https://doi.org/10.1021/ja809598r

Liu, Y., & Wu, H. (2017). Prediction of Road Traffic Congestion Based on Random Forest. *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, *2*, 361–364. https://doi.org/10.1109/ISCID.2017.216

Mellit, A., & Kalogirou, S. A. (2018). *Chapter II-1-D - A Survey on the Application of Artificial Intelligence Techniques for Photovoltaic Systems* (S. A. B. T.-M. H. of P. (Third E. Kalogirou (ed.); pp. 735–761). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-809921-6.00019-7

Morgan, D., & Jacobs, R. (2020). Opportunities and Challenges for Machine Learning in Materials

Science. *Annual Review of Materials Research*, *50*(1), 71–103. https://doi.org/10.1146/annurev-matsci-070218-010015

Ning, S., Zhang, S., Sun, J., Li, C., Zheng, J., Khalifa, Y. M., Zhou, S., Cao, J., & Wu, Y. (2020). Ambient Pressure X-ray Photoelectron Spectroscopy Investigation of Thermally Stable Halide Perovskite Solar Cells via Post-Treatment. *ACS Applied Materials and Interfaces*, *12*(39), 43705–43713. https://doi.org/10.1021/acsami.0c12044

NREL. (2024). *Best Research-Cell Efficiency Chart.* National Renewable Energy Laboratory. https://www.nrel.gov/pv/cell-efficiency.html

Odabaşı, Ç., & Yıldırım, R. (2020a). Assessment of Reproducibility, Hysteresis, and Stability Relations in Perovskite Solar Cells Using Machine Learning. *Energy Technology*, *8*(12). https://doi.org/10.1002/ente.201901449

Odabaşı, Ç., & Yıldırım, R. (2020b). Machine learning analysis on stability of perovskite solar cells. *Solar Energy Materials and Solar Cells*, *205*. https://doi.org/10.1016/j.solmat.2019.110284

Shin, K.-S., Lee, T. S., & Kim, H. (2005). An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, *28*(1), 127–135. https://doi.org/https://doi.org/10.1016/j.eswa.2004.08.009

Stoumpos, C. C., Malliakas, C. D., & Kanatzidis, M. G. (2013). Semiconducting tin and lead iodide perovskites with organic cations: phase transitions, high mobilities, and near-infrared photoluminescent properties. *Inorganic Chemistry*, *52*(15), 9019–9038. https://doi.org/10.1021/ic401215x

Trinchero, R., & Canavero, F. (2021). Machine Learning Regression Techniques for the Modeling

of Complex Systems: An Overview. *IEEE Electromagnetic Compatibility Magazine*, *10*(4), 71–79. https://doi.org/10.1109/MEMC.2021.9705310

Wang, W., Men, C., & Lu, W. (2008). Online prediction model based on support vector machine. *Neurocomputing*, *71*(4), 550–558. https://doi.org/https://doi.org/10.1016/j.neucom.2007.07.020

Wasserstein, R. L., & Lazar, N. A. (2016). The ASA Statement on p-Values: Context, Process, and Purpose. *The American Statistician*, *70*(2), 129–133. https://doi.org/10.1080/00031305.2016.1154108

Willmott, C. J. (1981). ON THE VALIDATION OF MODELS. *Physical Geography*, *2*(2), 184–194. https://doi.org/10.1080/02723646.1981.10642213

Wu, X., Liu, Z., Yin, L., Zheng, W., Song, L., Tian, J., Yang, B., & Liu, S. (2021). A Haze Prediction Model in Chengdu Based on LSTM. *Atmosphere*, *12*(11). https://doi.org/10.3390/atmos12111479

Yang, W. S., Noh, J. H., Jeon, N. J., Kim, Y. C., Ryu, S., Seo, J., & Seok, S. Il. (2015). High-performance photovoltaic perovskite layers fabricated through intramolecular exchange. *Science*, *348*(6240), 1234–1237. https://doi.org/10.1126/science.aaa9272

Yoo, J. J., Seo, G., Chua, M. R., Park, T. G., Lu, Y., Rotermund, F., Kim, Y.-K., Moon, C. S., Jeon, N. J., Correa-Baena, J.-P., Bulović, V., Shin, S. S., Bawendi, M. G., & Seo, J. (2021). Efficient perovskite solar cells via improved carrier management. *Nature*, *590*(7847), 587–593. https://doi.org/10.1038/s41586-021-03285-w

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*, 308–324.

https://doi.org/https://doi.org/10.1016/j.trc.2015.02.019

Zhang, Z., Lu, J., Zhou, G., & Liao, X. (2018). Research on Tool Wear Prediction Based on LSTM and ARIMA. *Proceedings of the 2018 International Conference on Big Data Engineering and Technology*, 73–77. https://doi.org/10.1145/3297730.3297732