



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

**PREDICCIÓN DE LA DEMANDA ELÉCTRICA
EN LA REGIÓN ORIENTE DE MÉXICO
UTILIZANDO REDES NEURONALES CON
DATOS DEL CENACE**

Tesis presentada al

Colegio de Física

como requisito parcial para la obtención del grado de

LICENCIADO EN FÍSICA APLICADA

por

Eira Xanat Toral Mora

Asesorada por

Dr. Enrique Varela Carlos
Dr. Humberto Antonio Salazar Ibargüen

Puebla Pue.
04 de Diciembre de 2025



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

**PREDICCIÓN DE LA DEMANDA ELÉCTRICA
EN LA REGIÓN ORIENTE DE MÉXICO
UTILIZANDO REDES NEURONALES CON
DATOS DEL CENACE**

Tesis presentada al

Colegio de Física

como requisito parcial para la obtención del grado de

LICENCIADO EN FÍSICA APLICADA

por

Eira Xanat Toral Mora

Asesorada por

Dr. Enrique Varela Carlos

Dr. Humberto Antonio Salazar Ibargüen

Puebla Pue.

04 de Diciembre de 2025

**Título: PREDICCIÓN DE LA DEMANDA ELÉCTRICA EN
LA REGIÓN ORIENTE DE MÉXICO UTILIZANDO
REDES NEURONALES CON DATOS DEL CENACE**

Estudiante: EIRA XANAT TORAL MORA

COMITÉ

Dr. Iván Fuentecilla Cárcamo
Presidente

Dr. José Juan Castro Alva
Secretario

Dr. Sebastián Rosado Navarro
Vocal

Dr. Luis Manuel Villaseñor Cendejas
Vocal

Dr. Enrique Varela Carlos
Dr. Humberto Antonio Salazar Ibargüen
Asesor

Agradecimientos

Quiero expresar mi más profundo agradecimiento al doctor Enrique Varela por su acompañamiento y apoyo en este camino de aprendizaje dentro del mundo de la Ciencia de Datos. Su dedicación y entrega hacia la enseñanza son una inspiración constante para quienes tenemos el privilegio de ser sus alumnos. Al doctor Humberto Salazar, por haberme brindado la oportunidad de desarrollar este trabajo en el CIIEC, y por su apoyo invaluable que me permitió concluirlo de manera satisfactoria.

A mi papá Tláloc, por su constante acompañamiento, cariño y —por qué no decirlo— sus regaños, que me impulsaron a continuar una carrera que en sus inicios parecía difícil de terminar, porque siempre me ha enseñado la importancia de salir adelante sin importar las circunstancias.

A mi mamá Paty, quien me heredó el amor por las matemáticas y quien repeditamente me decía "hija, se ve que te apasiona la física, no lo dudes". Aunque no pudo acompañarme físicamente para verme concluir este logro, sé que ha estado conmigo en cada paso que doy, desde su cielo.

A mi abuelita Beatriz, por su apoyo incondicional y por cada llamada que me recordaba que, sin importar la distancia, me esperaba con orgullo y con el deseo de verme superarme día con día.

A mi hermano Yael, por sus palabras de ánimo —tan particulares como efectivas— cada vez que me veía dudar o cansarme, y porque gracias a él tuve el primer acercamiento real a la física.

Y finalmente, a mi mejor amigo Emiliano, por su paciencia infinita al explicarme los conceptos más complejos de física cuando más lo necesitaba, y por no soltar mi mano en los momentos más difíciles de esta etapa siendo foránea.

Índice general

Resumen	XIII
1. Importancia del pronóstico de la demanda de la energía eléctrica	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivos Específicos y Metas	2
1.3. Hipótesis	2
1.4. Justificación	2
2. Modelos de Series Temporales Tradicionales	5
2.1. Características de las series de tiempo	5
2.2. Métodos de análisis de series de tiempo	6
2.3. Forecasting	6
2.4. Métodos de evaluación	7
2.4.1. Cálculo del coeficiente de determinación R^2	7
3. Redes Neuronales y Aprendizaje Automático	9
3.1. Redes Neuronales Artificiales	9
3.2. LSTM (Long Short-Term Memory)	10
3.3. Boosted Decision Trees (BDTs)	11
4. Fundamentos de la predicción de la demanda eléctrica en la región Oriente	13
4.1. Situación del Sistema Eléctrico en Nacional y el papel del CENACE	13
4.2. El Papel del CENACE	13
4.3. Avances Internacionales en la Predicción de la Demanda Eléctrica	14
4.4. Limitaciones de los métodos tradicionales de predicción de demanda eléctrica en México	15
4.5. Selección de los Modelos Predictivos	16
5. Recolección de datos	17
5.1. Descripción de la base de datos del CENACE	17
5.1.1. Relaciones entre las columnas	18
5.2. Procesamiento	19
5.3. Tratamiento de datos	20
5.3.1. Descarga de datos	20
5.3.2. Normalización y estandarización de la base de datos	21
5.3.3. Valores faltantes y justificación de la imputación	22

6. Modelo LSTM	25
6.1. Arquitectura del modelo LSTM para la predicción de demanda eléctrica	25
6.1.1. Preparación de los datos	25
6.1.2. Creación de secuencias temporales	26
6.1.3. Definición del modelo LSTM	26
6.1.4. Predicción para el año 2024	27
6.1.5. Visualización de los resultados	27
6.1.6. Resultados obtenidos	28
7. Modelo XGBoost	31
7.1. Arquitectura del modelo XGBoost para la predicción de demanda eléctrica	31
7.1.1. Preparación de los datos	31
7.1.2. Creación de características temporales avanzadas	32
7.1.3. Configuración y entrenamiento del modelo XGBoost	33
7.1.4. Evaluación del modelo	34
7.1.5. Visualización de resultados	34
7.1.6. Resultados obtenidos	36
8. Discusión	37
8.1. Comparación y coherencia entre los modelos LSTM y XGBoost	37
8.1.1. Metodología y coherencia en el tratamiento de los datos	37
8.1.2. Comparación del desempeño	37
8.1.3. Rendimiento computacional de los modelos	37
8.1.4. Conclusiones comparativas	39
8.2. Ventajas	39
8.3. Limitaciones	40
9. Conclusiones	41
A. Código de descarga y preparación de datos	43

Índice de figuras

5.1. Página CENACE de la estimación de la demanda	19
5.2. Página CENACE del pronóstico de la demanda.	20
5.3. Fechas para la descarga	20
5.4. Lista de archivos csv	21
5.5. Visualización de archivo csv	21
6.1. Comparación entre las predicciones del modelo LSTM y los datos reales de demanda en la región ORI (enero–octubre 2024).	28
7.1. Comparación entre la demanda real y la predicha por el modelo XGBoost en la región ORI (enero–octubre 2024).	35

Índice de tablas

4.1. Comparación de enfoques de predicción de demanda eléctrica	16
6.1. Coeficiente de determinación (R^2) mensual del modelo LSTM para el año 2024 en la región Oriente.	28
7.1. Coeficiente de determinación (R^2) mensual del modelo XGBoost para el año 2024 en la región Oriente.	36
8.1. Comparación del rendimiento computacional entre los modelos LSTM y XGBoost.	38

Resumen

El presente trabajo tiene como objetivo desarrollar un modelo de predicción de la demanda eléctrica en la región Oriente de México, utilizando redes neuronales artificiales con datos históricos proporcionados por el Centro Nacional de Control de Energía (CENACE). Este estudio surge de la necesidad de mejorar la precisión de los pronósticos energéticos ante los desafíos actuales del sistema eléctrico nacional, tales como el crecimiento sostenido del consumo, la incorporación de fuentes renovables y la búsqueda de un suministro confiable y eficiente.

Para abordar el problema, se construyeron y compararon dos modelos predictivos: una red neuronal recurrente del tipo Long Short-Term Memory (LSTM) y un modelo basado en árboles potenciados (XGBoost). Ambos fueron entrenados con datos horarios de demanda, generación e intercambio neto de energía del periodo 2020–2023, y posteriormente aplicados para estimar la demanda de 2024. El modelo LSTM se enfocó en capturar dependencias temporales de largo plazo mediante secuencias de entrada, mientras que XGBoost utilizó características estadísticas y retardos temporales para representar las variaciones de la serie.

Los resultados muestran que ambos modelos ofrecen un desempeño robusto, con valores del coeficiente de determinación R^2 superiores al 0.98 en la mayoría de los meses analizados. Sin embargo, el modelo LSTM presentó una mayor capacidad para representar los patrones horarios y estacionales de la demanda, mientras que XGBoost destacó por su eficiencia computacional y menor sensibilidad a valores atípicos. La comparación entre ambos enfoques permite concluir que las redes neuronales recurrentes constituyen una herramienta de gran potencial para la planeación eléctrica en México, al ofrecer predicciones más precisas que los métodos tradicionales y facilitar la toma de decisiones en la operación del sistema eléctrico nacional.

Capítulo 1

Importancia del pronóstico de la demanda de la energía eléctrica

1.1. Planteamiento del problema

El uso de la energía eléctrica es un elemento esencial para el funcionamiento de la sociedad moderna (**cenace2025**). Desde la movilidad urbana, la industria y los servicios, hasta las actividades cotidianas en los hogares, la electricidad sostiene prácticamente todos los procesos que permiten el desarrollo económico y tecnológico de un país. La capacidad de garantizar un suministro continuo, estable y eficiente depende directamente de una adecuada planificación, la cual sólo es posible si se dispone de pronósticos confiables de la demanda eléctrica.

A nivel internacional, la importancia del pronóstico eléctrico ha motivado el desarrollo de numerosos estudios y metodologías avanzadas. En la revisión de Dong (**dong2025**), que analiza más de 650 trabajos publicados entre 2014 y 2023, muestra que los sistemas eléctricos de países como China, Estados Unidos, Australia y diversas naciones europeas dependen cada vez más de modelos predictivos modernos debido a la creciente complejidad de las series de demanda. Estos estudios evidencian que la predicción eléctrica no solo es una herramienta operativa, sino un componente estratégico para el funcionamiento de los sistemas energéticos contemporáneos.

México enfrenta desafíos cada vez más complejos derivados del crecimiento sostenido de la demanda energética, la transición hacia fuentes renovables y la necesidad de garantizar un suministro eficiente y confiable. La planificación y operación dependen en gran medida de la capacidad de predecir con precisión la demanda eléctrica, lo cual permite optimizar la generación, distribución y uso de los recursos energéticos. Sin embargo, los métodos tradicionales de predicción presentan limitaciones significativas al trabajar con datos que exhiben patrones no lineales, estacionales y con dependencias complejas.

Estas deficiencias se agravan en un contexto donde factores externos, como cambios climáticos, horarios laborales y eventos socioeconómicos, ejercen una influencia significativa sobre la demanda eléctrica, aumentando la dificultad de realizar predicciones precisas. La falta de herramientas robustas para abordar esta complejidad puede derivar en ineficiencias operativas, sobrecostos en la generación y riesgos de interrupciones en el suministro.

En este contexto, el uso de tecnologías modernas, como las redes neuronales, se presenta como una solución prometedora. Estas herramientas tienen la capacidad de manejar grandes volúmenes de datos, identificar patrones complejos y generar predicciones más precisas que los métodos

tradicionales. Sin embargo, en México, el uso de redes neuronales para la predicción de la demanda eléctrica está todavía en etapas iniciales, y su potencial no ha sido completamente explorado ni implementado en la práctica.

Por lo tanto, surge la necesidad de desarrollar e implementar modelos que puedan superar las limitaciones de los enfoques tradicionales y contribuir a la modernización tecnológica del sistema eléctrico mexicano. Este trabajo busca abordar este problema, demostrando cómo estas tecnologías avanzadas pueden integrarse en el contexto actual para optimizar la gestión del sistema eléctrico y satisfacer las demandas del futuro.

1.2. Objetivos

Desarrollar un modelo de predicción de la demanda eléctrica en la región Oriente en México utilizando redes neuronales con datos históricos proporcionados por el CENACE.

1.2.1. Objetivos Específicos y Metas

- **Analizar e identificar el valor agregado de los modelos predictivos** de redes neuronales artificiales en la era de evolución tecnológica actual.
- **Validar el rendimiento de los modelos** propuestos y evaluarlo haciendo la comparación con datos reales propuestos por el CENACE.
- **Crear un análisis comparativo** que demuestre el valor agregado de utilizar modelos de redes neuronales frente a metodologías tradicionales.

1.3. Hipótesis

El uso de redes neuronales recurrentes optimizadas, como los modelos LSTM y XGBoost, permite predecir la demanda eléctrica con mayor precisión que los modelos tradicionales debido a su capacidad para capturar patrones no lineales y dependencias temporales en los datos históricos proporcionados por el CENACE. Esto resulta en una herramienta más eficaz para la planificación y operación del sistema eléctrico en México.

1.4. Justificación

El pronóstico de la demanda eléctrica es una tarea esencial para la planificación del sistema eléctrico nacional, ya que permite anticipar el comportamiento del consumo y contribuir a una operación más eficiente del sistema. En México, los métodos tradicionales utilizados para estimar la demanda suelen enfrentarse a importantes limitaciones, especialmente cuando los datos presentan variaciones no lineales y dependencias complejas. Los métodos tradicionales no logran capturar adecuadamente los patrones no lineales ni las dependencias temporales que caracterizan a las series de demanda eléctrica. Ante estos retos, es necesario explorar enfoques modernos que permitan una mayor capacidad de adaptación a los cambios que ocurren en la demanda eléctrica.

En la actualidad, diversos países han incorporado modelos basados en redes neuronales y aprendizaje automático para abordar problemas similares. De acuerdo con la revisión realizada por Dong et al. (2025), más de 650 artículos publicados entre 2014 y 2023 muestran que modelos como LSTM, GRU, CNN y variantes híbridas se han convertido en herramientas ampliamente utilizadas

para el pronóstico de corto plazo en sistemas eléctricos de China, Estados Unidos, Australia, India y distintos países europeos. Este tipo de modelos ha demostrado su efectividad para capturar comportamientos complejos, aprender patrones con alta variabilidad y generar predicciones que superan a métodos tradicionales cuando existen dependencias temporales de largo plazo.

Tomando como referencia estos avances internacionales, la presente tesis propone utilizar dos modelos modernos: una red neuronal del tipo LSTM y un modelo basado en árboles (XGBoost). Ambos modelos han mostrado un desempeño adecuado en estudios internacionales donde es necesario representar variaciones rápidas en la demanda o aprender patrones no lineales en los datos de consumo. En este sentido, el trabajo desarrollado busca adaptar estas técnicas al caso mexicano, utilizando datos reales de la región Oriente proporcionados por el CENACE con el propósito de generar un pronóstico para el año 2024 y contribuir a la modernización de los métodos empleados en la planificación energética nacional.

Capítulo 2

Modelos de Series Temporales Tradicionales

Las **series de tiempo** son un tipo especial de datos en los que las observaciones se registran en intervalos regulares o irregulares a lo largo del tiempo. Esta estructura temporal hace que las series de tiempo tengan características únicas, como la *dependencia temporal* entre las observaciones, que es uno de los factores más importantes a considerar en su análisis.

Es habitual observar series temporales en sets de datos, como la cantidad de ventas en un determinado tiempo, el cambio de temperatura por temporadas, la variación del valor de la moneda, entre otras. Es común encontrarlas en muchos campos de conocimiento, como en economía, ciencias naturales, negocios y, ahora, para nuestro tema de interés, el consumo de energía eléctrica (box2015.)

■ Definición de una Serie Temporal

Una **serie temporal** es una secuencia de observaciones ordenadas en el tiempo, denotada por:

$$Y_t = \{y_1, y_2, \dots, y_t, \dots\}, \quad t \in \mathbb{Z}^+$$

donde y_t representa el valor de la serie en el tiempo t .

2.1. Características de las series de tiempo

- **Dependencia temporal:** A diferencia de otros conjuntos de datos donde las observaciones pueden ser independientes, en las series de tiempo existe una relación entre los valores anteriores y futuros.
- **Estacionariedad:** Una serie de tiempo es estacionaria cuando sus propiedades estadísticas (media, varianza y autocorrelación) no dependen del tiempo. Es por ello que los métodos tradicionales presentan dificultades para hacer predicciones con precisión (box2015).
- **Tendencia:** Es el patrón a largo plazo que indica si los valores de la serie tienden a aumentar, disminuir o permanecer constantes.
- **Estacionalidad:** Son patrones cíclicos que se repiten en intervalos regulares.
- **Ruido:** Son las fluctuaciones aleatorias que no se ajustan a ningún patrón predecible.

- **Ciclos:** Patrones que se repiten, pero no necesariamente en intervalos regulares, como los ciclos económicos.

2.2. Métodos de análisis de series de tiempo

Existen varios métodos matemáticos y algoritmos para analizar y predecir series de tiempo (hamilton1994).

1. Modelos estadísticos clásicos:

- *ARIMA (AutoRegressive Integrated Moving Average)*: Modelo para series estacionarias y no estacionarias.
- *SARIMA*: Extensión del ARIMA que considera la estacionalidad.
- *Suavización exponencial (ETS)*: Técnica que da mayor peso a observaciones recientes.

2. Métodos basados en redes neuronales:

- *Redes neuronales recurrentes (RNN)* y *LSTM (Long Short-Term Memory)*: Ideales para manejar dependencias temporales largas.
- *Árboles de decisión*: Modelos predictivos que dividen los datos en ramas jerárquicas basadas en reglas simples de decisión. Permiten identificar relaciones no lineales.

2.3. Forecasting

El forecasting o pronóstico es una disciplina que busca predecir el comportamiento futuro de una variable basándose en datos históricos y patrones identificables (hyndman2018).

Su funcionamiento se basa en la identificación de tendencias, patrones estacionales y fluctuaciones en los datos para realizar predicciones precisas. Se aplica en diversos campos, como la economía, la meteorología, el marketing y, especialmente, en la predicción de la demanda eléctrica, donde su precisión es crucial para estimar el consumo de la demanda de energía para la toma de decisiones.

Según Hyndman y Athanasopoulos (2018) (hyndman2018), el forecasting es “el proceso de estimación de valores futuros de una variable basada en información pasada y presente, utilizando modelos matemáticos y estadísticos.”

Matemáticamente, este problema se puede representar de la siguiente manera:

Sea $X = \{x_1, x_2, \dots, x_T\}$ una serie temporal con T observaciones y queremos predecir los siguientes H valores futuros:

$$\hat{x}_{T+1}, \hat{x}_{T+2}, \dots, \hat{x}_{T+H} \quad (2.1)$$

donde H es el **horizonte de predicción**.

El objetivo del forecasting es encontrar una función f que relacione los valores pasados con las predicciones futuras:

$$\hat{x}_{T+h} = f(x_T, x_{T-1}, \dots, x_{T-p}) \quad (2.2)$$

donde p es el número de rezagos (lags) que se utilizan en la predicción.

2.4. Métodos de evaluación

Para conocer la precisión de un modelo existen diferentes métodos de evaluación.

2.4.1. Cálculo del coeficiente de determinación R^2

El coeficiente de determinación R^2 , que indica la proporción de la varianza de los datos que el modelo es capaz de explicar.

$$R^2 = 1 - \frac{\sum (y_{\text{real}} - y_{\text{pred}})^2}{\sum (y_{\text{real}} - \bar{y})^2} \quad (2.3)$$

Un valor de $R^2 = 1$ implica un ajuste perfecto, mientras que valores negativos indican que el modelo es menos preciso que una predicción basada en la media (**pedregosa2011scikit**).

Capítulo 3

Redes Neuronales y Aprendizaje Automático

3.1. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) son modelos computacionales basados en la composición neuronal del cerebro humano. Según Goodfellow, Bengio y Courville (**goodfellow2016**), "las redes neuronales artificiales son un conjunto de algoritmos que están diseñados para reconocer patrones mediante el aprendizaje a partir de datos de entrada."

Las unidades de procesamiento de las RNA son las neuronas, que se conforman por tres tipos de capas:

- **Capa de entrada:** Recibe los datos de entrada y los transmite a la siguiente capa.
- **Capas ocultas:** Procesan la información aplicando transformaciones no lineales mediante funciones de activación
- **Capa de salida:** Produce la predicción final basada en la activación de las neuronas en las capas ocultas.

Con ellas se pueden modelar relaciones complejas en los datos. Se representan matemáticamente como:

$$\hat{x}_{T+1} = \sigma(W \cdot X + b) \quad (3.1)$$

donde:

- W es la matriz de pesos de la red neuronal, que son los parámetros que la red ajusta durante el entrenamiento. Son los valores que determinan cuánto influye cada entrada en la salida de la neurona.
- X es el vector de entrada con valores pasados $(x_T, x_{T-1}, \dots, x_{T-p})$.
- b es el sesgo (bias).
- σ es la función de activación, como *ReLU* o *sigmoide*.

Las RNA aprenden ajustando los pesos mediante algoritmos de optimización basados en el *descenso de gradiente*. El procedimiento se realiza en los siguientes pasos:

- 1. Cálculo de la salida de la red.
- 2. Evaluación del error con una función de pérdida como:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.2)$$

- 3. Ajuste de los pesos utilizando **retropropagación del error**:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial L}{\partial w_{ij}} \quad (3.3)$$

donde η es la tasa de aprendizaje.

Un arreglo de una dimensión es un vector y de 2 o más dimensiones es una matriz.

3.2. LSTM (Long Short-Term Memory)

Las **Redes LSTM** son un tipo de red neuronal recurrente diseñada para aprender dependencias a largo plazo en datos secuenciales.

Introducidas por Hochreiter y Schmidhuber (**brownlee2018; hochreiter1997**), resuelven el problema del desvanecimiento del gradiente en RNN convencionales mediante una arquitectura que incluye una memoria que se actualiza mediante compuertas de entrada, olvido y salida, lo que las hace ideales para la predicción de series temporales.

Sus tres compuertas fundamentales:

- **Compuerta de olvido:** Decide qué información de la memoria anterior debe descartarse, según la ecuación:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (3.4)$$

- **Compuerta de entrada:** Controla qué información nueva se agrega a la memoria:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (3.5)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \quad (3.6)$$

- **Compuerta de salida:** Determina qué parte de la memoria se usa como salida en el siguiente estado:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (3.7)$$

$$h_t = o_t \cdot \tanh(c_t). \quad (3.8)$$

La memoria de la celda c_t se actualiza combinando la información retenida y la nueva información aprendida:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{C}_t. \quad (3.9)$$

Las redes neuronales LSTM son altamente usadas para los datos secuenciales porque aprenden patrones temporales en los datos, recordando relaciones de corto y largo plazo. También mitigan el problema del gradiente desvanecido, permitiendo el aprendizaje de patrones en secuencias largas.

3.3. Boosted Decision Trees (BDTs)

El algoritmo XGBoost (Extreme Gradient Boosting) es una técnica de aprendizaje supervisado basada en el método de boosting, el cual es una técnica que combina múltiples modelos débiles, comúnmente árboles de decisión, en una secuencia de etapas para formar un modelo fuerte (hastie2009; chen2016).

Matemáticamente, la predicción final se construye como una suma de árboles:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}, \quad (3.10)$$

donde $\hat{y}_i$ es la predicción para la muestra i , f_k es el árbol k -ésimo y $\mathcal{F}$ es el espacio de funciones que define todos los árboles posibles.

El objetivo es minimizar la siguiente función de pérdida regularizada:

$$\mathcal{L} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (3.11)$$

donde l es la función de pérdida (por ejemplo, error cuadrático medio), y $\Omega(f_k)$ es un término de regularización que penaliza la complejidad del modelo, definido como:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2, \quad (3.12)$$

donde T es el número de hojas del árbol, w son los valores predichos en cada hoja, γ controla el número de nodos y λ controla la magnitud de los pesos para evitar sobreajuste.

Aplicación en Series Temporales

XGBoost no es intrínsecamente secuencial como las redes neuronales recurrentes, por lo que no puede modelar directamente dependencias temporales. Sin embargo, puede aplicarse eficazmente en series temporales al reformular el problema como un conjunto de datos supervisado. Esto se logra mediante la construcción de variables explicativas a partir de rezagos temporales (lags), que permiten capturar la dinámica a corto plazo en la serie.

Una vez que los datos se estructuran adecuadamente, XGBoost puede entrenarse como un modelo de regresión para predecir valores futuros de la serie. Entre sus ventajas se encuentra su robustez frente a datos ruidosos, la capacidad de manejar relaciones no lineales y la posibilidad de interpretar la importancia relativa de cada variable predictora.

Ventajas en Contextos de Predicción Temporal

- Alta precisión en tareas de regresión sobre datos estructurados.
- Control del sobreajuste mediante regularización L1 y L2.
- Capacidad de manejar valores faltantes y trabajar eficientemente con grandes volúmenes de datos.
- Interpretabilidad mediante la evaluación de la importancia de características.
- Posibilidad de combinarse con redes neuronales para formar modelos híbridos.

Capítulo 4

Fundamentos de la predicción de la demanda eléctrica en la región Oriente

4.1. Situación del Sistema Eléctrico en Nacional y el papel del CENACE .

El Sistema Eléctrico Nacional (SEN) de México es una infraestructura compleja y vital que garantiza el suministro de energía eléctrica en todo el país. Su operación y control están a cargo del Centro Nacional de Control de Energía (CENACE), un organismo público descentralizado con personalidad jurídica y patrimonio propio.(cenace2025).

El SEN está compuesto por diversas centrales eléctricas que emplean diferentes tecnologías de generación, incluyendo fuentes renovables y convencionales. Estas centrales están interconectadas a través de la Red Nacional de Transmisión y las Redes Generales de Distribución, lo que permite una distribución eficiente de la energía eléctrica a lo largo del territorio nacional.

4.2. El Papel del CENACE

El CENACE tiene la responsabilidad de:

1. Control Operativo del SEN: Gestiona en tiempo real la operación del sistema eléctrico para asegurar un suministro continuo y confiable de energía eléctrica.
2. Operación del Mercado Eléctrico Mayorista (MEM): Administra el MEM, promoviendo la competencia y la eficiencia en la generación y comercialización de electricidad. Esto incluye la asignación y despacho óptimos de las centrales eléctricas para satisfacer la demanda energética (cenace2025).
3. Garantizar el Acceso Imparcial a la Red: Asegura que todos los participantes del mercado tengan acceso equitativo a la Red Nacional de Transmisión y a las Redes Generales de Distribución, fomentando un entorno competitivo y transparente (cenace2025).
4. Planeación de la Expansión y Modernización de la Red: Es responsable de formular programas para la ampliación y modernización de la infraestructura de transmisión y distribución

eléctrica, los cuales, una vez autorizados por la Secretaría de Energía (SENER), se incorporan al Programa de Desarrollo del Sistema Eléctrico Nacional PRODESEN (**cenace2025**).

La planificación del desarrollo del SEN se fundamenta en el estudio de la demanda y el consumo eléctrico proyectados a mediano y largo plazo, considerando las estimaciones de la demanda máxima integrada, tanto en términos brutos como netos, y el consumo total de energía. Este enfoque permite estructurar de forma eficiente el crecimiento y la ampliación de las capacidades en generación, transmisión, distribución y comercialización de electricidad, además de garantizar el suministro adecuado de los insumos esenciales (**cenace_funciones2025**).

Los siguientes factores influyen en la demanda y el consumo eléctrico neto:

- **Crecimiento Económico** Hace referencia a el crecimiento del Producto Interno Bruto (PIB). A mayor crecimiento de la economía de una región habrá mayor demanda de electricidad (**prodesen2023**).
- **Crecimiento Poblacional** Es el crecimiento poblacional de una región que está relacionado con los servicios públicos y comerciales, desarrollo industrial y crecimiento de vivienda que tiene como consecuencia más demanda eléctrica (**prodesen2023**).
- **Estacionalidad** Los factores climáticos son la primera consecuencia de la estacionalidad, tales como los fenómenos El Niño, La Niña, huracanes, entre otros, estos provocan diversas variaciones en la demanda y consumo de energía eléctrica (**prodesen2023**).
- **Precio de combustibles** El precio de los combustibles repercute al precio de las tarifas eléctricas, por lo tanto, en el consumo y demanda de la energía eléctrica (**prodesen2023**).
- **Pérdidas de energía eléctrica** El efecto encargado en las pérdidas de energía eléctrica por cuestiones técnicas es el efecto Joule, que es el responsable del calentamiento de los conductores eléctricos (**prodesen2023**).

4.3. Avances Internacionales en la Predicción de la Demanda Eléctrica

Durante la última década, el uso de modelos basados en aprendizaje profundo se ha convertido en una de las principales líneas de investigación para la predicción de la demanda eléctrica a nivel mundial. Este enfoque responde a la creciente complejidad de las series de carga, las cuales presentan variaciones no lineales y comportamientos influenciados por factores climáticos, estacionales y socioeconómicos, dificultando su representación mediante métodos tradicionales.

En el análisis realizado por Dong (**dong2025**), que revisa más de 650 estudios sobre predicción eléctrica, se destaca que las arquitecturas LSTM y GRU son las más utilizadas en países como China, Australia, Estados Unidos y diversas regiones de Europa. Estos modelos han sido aplicados exitosamente en redes urbanas, residenciales e industriales, debido a su capacidad para aprender dependencias temporales prolongadas y capturar variaciones complejas en datos horarios.

La revisión también reporta el uso de modelos basados en CNN y TCN, empleados para extraer patrones locales y variaciones de corto plazo mediante operaciones convolucionales. Asimismo, en regiones como España, Alemania e Irlanda se han explorado modelos híbridos que combinan LSTM con CNN o técnicas de boosting, con el fin de representar simultáneamente tendencias, estacionalidades y rezagos. Por su parte, en Estados Unidos, operadores como ISO-NE y PJM han incorporado modelos modernos en esquemas de pronóstico horario apoyados en bases de datos de

alta resolución.

En conjunto, estos avances evidencian una tendencia internacional hacia el uso de redes neuronales especializadas y algoritmos de aprendizaje automático para mejorar la precisión del pronóstico de demanda. Dichas metodologías han mostrado una notable adaptabilidad ante cambios estructurales en el consumo, razón por la cual su adopción en este trabajo se alinea con las estrategias internacionales más recientes y contribuye a la modernización del análisis aplicado al contexto mexicano.

4.4. Limitaciones de los métodos tradicionales de predicción de demanda eléctrica en México

En el contexto del sistema eléctrico mexicano, la predicción de la demanda energética es una actividad crítica para la planeación de la generación, transmisión y distribución de energía. De acuerdo con el Programa de Desarrollo del Sistema Eléctrico Nacional PRODESEN 2024-2038 (**prodesen2024**), el pronóstico de la demanda y consumo se realiza a partir de un proceso que integra balances de energía, escenarios macroeconómicos, eficiencia energética y proyecciones regionales. Para la elaboración de los pronósticos, se emplean modelos tradicionales como regresión simple, modelos exponenciales, adaptativos de primer orden, Holt-Winters, ARIMA, regresión lineal múltiple y técnicas de optimización horaria.

Estos métodos, aunque robustos en aplicaciones históricas, presentan limitaciones al enfrentar series temporales que exhiben patrones altamente no lineales, estacionalidades complejas y múltiples fuentes de variabilidad exógena, como cambios climáticos, alteraciones en los patrones de consumo socioeconómicos y variaciones en los precios de la energía. Además, los modelos tradicionales requieren una selección manual de variables explicativas y suponen relaciones estructurales fijas, lo que restringe su capacidad de adaptación ante cambios dinámicos en el sistema eléctrico.

La creciente participación de energías renovables también ha incrementado la variabilidad del sistema eléctrico. Este contexto contribuye a la necesidad de métodos de predicción más flexibles ya que la metodología tradicional modela la demanda a partir de datos agregados, frecuentemente a escala anual o mensual, con escasa resolución horaria, y su precisión depende de la calidad de los supuestos macroeconómicos de largo plazo. Esto limita su capacidad para capturar fenómenos súbitos o eventos no previstos, como contingencias climáticas extremas o transformaciones rápidas en la estructura de consumo.

Ante estos desafíos, se vuelve necesario explorar metodologías que permitan trabajar directamente con datos horarios y que ofrezcan una mayor capacidad de adaptación. Se consideran dos alternativas modernas: redes neuronales tipo LSTM y el algoritmo XGBoost, cuya incorporación busca superar las limitaciones de los métodos convencionales y mejorar la precisión del pronóstico de la demanda eléctrica en México.

Métodos Tradicionales PRODESEN	Modelo LSTM	Modelo XGBoost
Capturan la información mediante relaciones lineales y parámetros fijos (tendencia, estacionalidad, rezagos)	Aprende dependencias temporales largas y relaciones no lineales a partir de secuencias	Representa la dinámica temporal mediante características creadas explícitamente (lags, ventanas, estadísticas)
Predicción agregada (anual, regional)	Predicción horaria (secuencial)	Predicción horaria (en tabla estructurada)

Tabla 4.1: Comparación de enfoques de predicción de demanda eléctrica

En la Tabla 4.1 se comparan los principales enfoques utilizados para la predicción de la demanda eléctrica en el contexto de esta tesis: el método tradicional descrito en el PRODESEN 2024-2038, basado en modelos estadísticos convencionales; las redes neuronales profundas tipo LSTM; y el algoritmo XGBoost. Esta comparación fundamenta la necesidad de explorar tecnologías emergentes que superen limitaciones y respondan mejor a la creciente complejidad del sistema eléctrico nacional.

4.5. Selección de los Modelos Predictivos

Se seleccionaron dos enfoques complementarios para abordar la predicción de la demanda eléctrica con base en los datos proporcionados por el CENACE: las redes neuronales LSTM y el algoritmo de árboles potenciados XGBoost (**lara2021**). Ambos modelos han demostrado una alta eficacia en tareas de modelado de series temporales, cada uno con fortalezas particulares.

La base de datos del CENACE consiste en registros horarios de demanda eléctrica, los cuales presentan variaciones de corto y largo plazo y patrones estacionales pronunciados. En este tipo de series, es necesario emplear modelos capaces de aprovechar la estructura temporal o de representarla mediante ingeniería de características (**taylor2007**).

Las redes LSTM son adecuadas para esta tarea debido a su capacidad para aprender dependencias a largo plazo y patrones temporales complejos, lo cual es esencial en el caso de la demanda eléctrica, que exhibe ciclos diarios, semanales y estacionales. Estas redes pueden conservar información relevante durante múltiples pasos de tiempo, permitiendo una representación contextualizada de la serie.

Por otro lado, XGBoost destaca por su eficiencia computacional, interpretabilidad y precisión en tareas de regresión, especialmente cuando se trabaja con conjuntos de datos estructurados. Aunque no modela secuencias de forma nativa, puede capturar relaciones locales mediante la incorporación de características rezagadas como variables predictoras. Su capacidad para manejar grandes volúmenes de datos y su rapidez de entrenamiento lo convierten en un competidor sólido para la predicción horaria.

En conjunto, ambos modelos ofrecen perspectivas complementarias: LSTM aporta una representación secuencial profunda de la serie, mientras que XGBoost ofrece una aproximación estructurada basada en características explícitas. Su evaluación comparativa proporciona una visión integral sobre diferentes estrategias modernas aplicables al pronóstico de la demanda eléctrica.

Capítulo 5

Recolección de datos

5.1. Descripción de la base de datos del CENACE

La base de datos comprende las siguientes columnas (`cenace_demanda_real2025`).

■ **Sistema:**

- *Tipo de datos:* Cadena de texto (`str`).
- *Descripción:* Representa el sistema eléctrico regional, por ejemplo:
 - **BCA:** Baja California.
 - **BCS:** Baja California Sur.
 - **SIN:** Sistema Interconectado Nacional.

■ **Área:**

- *Tipo de datos:* Cadena de texto (`str`).
- *Descripción:* Subdivisión geográficas, comprende las regiones de:
 - **BCA:** Baja California.
 - **BCS:** Baja California Sur.
 - **CEN:** Centro.
 - **NES:** Noreste.
 - **NOR:** Norte.
 - **NTE:** Noroeste.
 - **OCC:** Occidente.
 - **ORI:** Oriente.
 - **PEN:** Peninsular.

Para fines de estudio, se trabajará específicamente con el área ORI en la cual se encuentra la ciudad de Puebla.

• **Generación (MWh):**

- *Tipo de datos:* Número decimal (`float`).
- *Descripción:* Cantidad de energía generada en la región, medida en megavatios-hora (MWh).

• **Importación Total (MWh):**

- *Tipo de datos:* Número decimal (`float`).

- *Descripción:* Energía eléctrica importada desde otras regiones o sistemas, medida en MWh.
- **Exportación Total (MWh):**
 - *Tipo de datos:* Número decimal (`float`).
 - *Descripción:* Energía eléctrica exportada hacia otras regiones o sistemas, medida en MWh.
- **Intercambio neto entre Gerencias (MWh):**
 - *Tipo de datos:* Cadena de texto (`str`) o número decimal (`float`).
 - *Descripción:* Balance neto del intercambio de energía entre diferentes gerencias. Valores como `-` indican datos no disponibles o no aplicables.
- **Estimación de Demanda por Balance (MWh):**
 - *Tipo de datos:* Número decimal (`float`).
 - *Descripción:* Estimación del consumo eléctrico total en la región, basada en cálculos de balance.
- **Pronóstico (MWh):**
 - *Tipo de datos:* Número decimal (`float`).
 - *Descripción:* Pronóstico de la demanda eléctrica, obtenido a partir de modelos predictivos.
- **Timestamp:**
 - *Tipo de datos:* Fecha y hora (`datetime`).
 - *Descripción:* Marca de tiempo que indica el momento específico de registro de los datos.

5.1.1. Relaciones entre las columnas

- **Balance de generación e importación/exportación:**

$$\text{Demanda estimada (MWh)} \approx \text{Generación (MWh)} + \text{Importación Total (MWh)} - \text{Exportación Total (MWh)}. \quad (5.1)$$

- **Pronóstico vs. estimación de demanda:**
 - El *Pronóstico* es una predicción futura, mientras que la *Estimación de Demanda* puede ser más cercana al consumo real registrado.
- **Intercambio neto entre gerencias:**
 - Un valor positivo indica que una gerencia entrega más energía de la que consume.
 - Un valor negativo indica que recibe más energía de otras gerencias.

5.2. Procesamiento

En la página oficial del CENACE dedicada a la estimación de la demanda eléctrica (**cenace_demanda_real2025**), se muestran los valores horarios de demanda. La Figura 5.5 presenta la interfaz desde la cual se descargaron los datos históricos empleados en este estudio.

Estimación de la Demanda Real

Por Balance

Fecha de Operación del Mercado
Visualizar: 16/01/2025

Si desea descargar los reportes de un periodo específico, proporcione:

Fecha Inicial: 01/01/2025

Fecha Final: 16/01/2025

Descargar en archivo .zip

Fecha de Operación del Mercado	No. de Liquidación Asociada	Csv	Pdf	Html	Fecha de Publicación
16/01/2025 12:00:00 a. m.	0	CSV	PDF	HTML	30/01/2025 12:25:01 p. m.

Por Retiros

Fecha de Operación del Mercado
Visualizar: 17/01/2025

Si desea descargar los reportes de un periodo específico, proporcione:

Fecha de Operación del Mercado	No. de Liquidación Asociada	Csv	Pdf	Html	Fecha de Publicación
17/01/2025 12:00:00 a. m.	0	CSV	PDF	HTML	31/01/2025 01:25:01 a. m.

Figura 5.1: Página CENACE de la estimación de la demanda

La Demanda Real del Sistema representa la cantidad de energía utilizada en el Sistema Eléctrico Nacional (SEN) durante un Día de Operación, correspondiendo al consumo neto de energía. Su estimación es una tarea fundamental del CENACE, que emplea dos metodologías principales para su cálculo: por balance y por retiros.

La estimación por balance se basa en la generación neta inyectada al sistema en cada hora, descontando la energía exportada. Este método incluye tanto pérdidas técnicas como no técnicas dentro del sistema eléctrico. Por otro lado, la estimación por retiros considera el total de compras de energía realizadas por las Entidades Responsables de Carga, incluidas las exportaciones, pero excluyendo las pérdidas de la red que forman parte del Mercado Eléctrico Mayorista.

Ambas estimaciones son esenciales para la liquidación de transacciones en el MEM, ya que permiten calcular de manera precisa el consumo de energía y ajustar los costos dentro del mercado.

La Figura 5.2 muestra la plataforma del CENACE donde se publica el pronóstico horario de la demanda eléctrica (**cenace_pronostico_demanda**).

Para cada Día de Operación, el CENACE estima la demanda esperada en el Mercado del Día en Adelanto, tomando como referencia las ofertas de compra enviadas por los Participantes del Mercado. Este pronóstico es fundamental para la planificación y asignación eficiente de recursos dentro del Mercado Eléctrico Mayorista.

Asimismo, en el proceso de Asignación Suplementaria de Unidades de Central Eléctrica para Confiabilidad, el CENACE realiza una estimación de la demanda en cada Zona de Carga para

Día de Operación	Sistema	Csv	Pdf	Html	Fecha de Publicación
31/01/2025	SEN				30/01/2025 11:00:01 p. m.

Figura 5.2: Página CENACE del pronóstico de la demanda.

los siguientes siete Días de Operación. Este procedimiento permite garantizar el abastecimiento eléctrico y la estabilidad operativa del Sistema Eléctrico Nacional (SEN), asegurando que las unidades de generación necesarias estén disponibles para responder a la demanda.

En total, para la región Oriente (ORI) se recopilieron 42,285 registros horarios correspondientes al periodo 2020–2024. Este volumen de información proporciona una base sólida para el entrenamiento de modelos de predicción basados en series temporales, permitiendo capturar tanto tendencias anuales como variaciones estacionales y cambios abruptos en la demanda eléctrica.

5.3. Tratamiento de datos

5.3.1. Descarga de datos

Fecha Inicial:
01/01/2020

Fecha Final:
31/12/2024

Descargar en archivo .zip

Figura 5.3: Fechas para la descarga

En el lado lateral izquierdo se encuentra un apartado para descargar los archivos por fechas. En este caso, para la base de datos se usó desde enero de 2020 a diciembre de 2024.

La descarga da un listado de archivos por día, cada uno especifica la fecha en el título del archivo. Contienen la siguiente información:

Nombre	Tipo
Demanda Real Balance_0_v3 Dia Operacion 2024-11-01 v2024 11 15_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-02 v2024 11 16_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-03 v2024 11 17_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-04 v2024 11 18_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-05 v2024 11 19_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-06 v2024 11 20_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-07 v2024 11 21_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-08 v2024 11 22_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-09 v2024 11 23_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-10 v2024 11 24_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-11 v2024 11 25_12 25 01	Microsoft Excel Comma S...
Demanda Real Balance_0_v3 Dia Operacion 2024-11-12 v2024 11 26_12 25 01	Microsoft Excel Comma S...

Figura 5.4: Lista de archivos csv

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10	BCA	BCA	1	1097.62109	31.18324	29.5867	---	1099.21764		
11	BCA	BCA	2	1057.97564	27.8844	25.60927	---	1060.23078		
12	BCA	BCA	3	1036.9812	14.29629	16.21068	---	1035.03681		
13	BCA	BCA	4	1030.27948	13.32465	16.8484	---	1026.75373		
14	BCA	BCA	5	1047.21282	6.95402	9.41331	---	1044.75353		
15	BCA	BCA	6	1087.0894	19.08236	11.29412	---	1094.87764		
16	BCA	BCA	7	1136.31729	117.17231	59.79563	---	1193.69397		
17	BCA	BCA	8	1180.95132	163.96969	65.13567	---	1279.78533		
18	BCA	BCA	9	1250.65568	120	21.89054	---	1348.75714		
19	BCA	BCA	10	1320.69571	94.67108	24.11906	---	1391.24773		
20	BCA	BCA	11	1389.17931	80.23863	38.66898	---	1430.74897		
21	BCA	BCA	12	1350.97925	201.04863	90.51995	---	1461.50794		
22	BCA	BCA	13	1467.38204	61.53716	29.95119	---	1498.96802		
23	BCA	BCA	14	1478.52981	61.18687	29.68341	---	1510.03326		
24	BCA	BCA	15	1473.31679	118.30396	80.75404	---	1510.86671		
25	BCA	BCA	16	1337.36561	341.25454	154.49862	---	1524.15153		
26	BCA	BCA	17	1282.82696	368.33957	117.78193	---	1533.3846		
27	BCA	BCA	18	1335.65583	390.06376	119.32691	---	1606.39269		

Figura 5.5: Visualización de archivo csv

Antes de crear la base de datos se tienen que tener en cuenta todas las características de los datos en crudo:

1. Se descarga un archivo csv por día
2. La fecha se encuentra el nombre del archivo
3. Cada CSV contiene la hora en formato 24 horas
4. Los primeros 8 renglones del CSV contienen información innecesaria

Aquí se pueden detectar diversas peculiaridades de los archivos que se tienen que cambiar para poder tener una base de datos que sea útil para el análisis.

5.3.2. Normalización y estandarización de la base de datos

Para garantizar la integridad y usabilidad de la información, se realizaron las siguientes transformaciones:

1. Eliminación de Filas Innecesarias

Los archivos CSV contenían las primeras 8 filas con información no relevante para el análisis. Se implementó un script en Bash para eliminar estas filas y conservar únicamente los datos estructurados desde la fila 9 en adelante.

2. Extracción de la Fecha desde el Nombre del Archivo

Dado que la fecha de cada reporte estaba contenida en el nombre del archivo, se desarrolló un script en Bash que extrae esta fecha y la incorpora como una nueva columna en cada archivo CSV. La fecha fue estandarizada en el formato YYYY-MM-DD para su correcta manipulación en el análisis posterior.

3. Unificación de Archivos

Cada archivo CSV representaba un día de operación, por lo que fue necesario consolidar toda la información en un solo archivo maestro. Se implementó un script que extrae los encabezados de un solo archivo y concatena los datos de todos los archivos eliminando encabezados duplicados generando un único archivo denominado mergeddata.csv con la totalidad de los registros.

4. Creación de la Columna TIMESTAMP

- Para facilitar el análisis temporal, se creó una columna TIMESTAMP, combinando la fecha y la hora de cada registro en el formato estándar YYYY-MM-DD HH:MM:SS.
- Se hizo un ajuste para la columna Hora, asegurando que los valores estuvieran en un rango válido (0-23 horas).
- Se concatenó con la fecha correspondiente y se convirtió a un formato de fecha-hora legible.

5. Verificación y Exploración de Datos

- Una vez consolidada la base de datos final, se realizaron verificaciones para garantizar la calidad de los datos:
- Se contó y verificó el número total de registros y columnas en la base de datos final el rango de fechas para asegurar la correcta integración de los datos históricos.
- Se revisaron las primeras y últimas filas para corroborar la estructura y consistencia de los registros. El proceso permitió la construcción de un archivo final finaldata.csv, que contiene la información de demanda eléctrica estructurada para su análisis. Este archivo servirá como base para modelos predictivos y estudios sobre la evolución de la demanda energética en el sistema eléctrico nacional.

5.3.3. Valores faltantes y justificación de la imputación

Antes de aplicar cualquier transformación se evaluó la presencia de datos faltantes en las series horarias correspondientes a la región Oriente (ORI). El conjunto contiene 42,285 registros horarios y cinco variables, lo que equivale a un total de 211,425 celdas. Se identificaron únicamente 48 valores faltantes, concentrados exclusivamente en la variable *Intercambio Neto entre Gerencias*, mientras que las demás columnas no presentaron valores nulos. Esto representa tan solo el 0.0227 % del total de datos.

Dado que la proporción de valores faltantes es extremadamente baja y no muestra un patrón sistemático, se aplicó imputación mediante la media aritmética de la variable afectada. Este método es adecuado cuando los valores faltantes pueden considerarse *Missing Completely At Random* (MCAR) y es ampliamente utilizado en series temporales energéticas (hyndman2018). Su implementación mediante SimpleImputer de la biblioteca scikit-learn (pedregosa2011) permite conservar la estructura temporal sin eliminar observaciones completas, lo cual es fundamental para la creación de ventanas deslizantes y para garantizar la continuidad requerida por los modelos predictivos empleados.

Este proceso asegura que el conjunto depurado esté completamente libre de valores nulos y preparado para la normalización y construcción de características temporales utilizadas en las etapas posteriores.

Capítulo 6

Modelo LSTM

6.1. Arquitectura del modelo LSTM para la predicción de demanda eléctrica

6.1.1. Preparación de los datos

Para el entrenamiento del modelo se utilizaron datos horarios de la región Oriente (ORI) del Sistema Eléctrico Nacional proporcionados por el CENACE, abarcando el periodo de 2020 a octubre de 2024. El conjunto incluye las siguientes variables: estimación de demanda por balance, generación total y el intercambio neto de energía entre gerencias.

$$\mathbf{Datos} = \begin{bmatrix} \text{Estimación de Demanda por Balance (MWh)} \\ \text{Generación (MWh)} \\ \text{Intercambio neto entre Gerencias (MWh)} \end{bmatrix} \quad (6.1)$$

El archivo de datos fue cargado en Python y se estableció el índice temporal utilizando la columna `Timestamp`, lo que permite manipular los registros como una serie de tiempo. Posteriormente, se filtraron los registros del área ORI y se excluyeron los datos posteriores a diciembre de 2023, dejando el periodo 2020–2023 para entrenamiento y reservando 2024 como conjunto de prueba.

```
1 train_data = pd.read_csv('database_2020_2021_2022_2023_2024_oct.csv',
2                           index_col='Timestamp', parse_dates=True)
3 train_data_or = train_data[train_data['Area'] == 'ORI']
4     ['Estimacion_de_Demanda_por_Balance_(MWh)',
5      'Generacion_(MWh)',
6      'Intercambio_netto_entre_Gerencias_(MWh)']
7 train_data_or = train_data_or[train_data_or.index.year <= 2023]
```

La calidad de los datos es un aspecto crítico en el modelado de series temporales. Por ello, se sustituyeron los valores ausentes o erróneos ('--', 'NaN', espacios en blanco) por valores nulos y se imputaron mediante la media aritmética de cada variable, técnica adecuada cuando las observaciones perdidas no presentan un patrón sistemático.

Esta imputación se realizó con la clase `SimpleImputer` de la biblioteca `scikit-learn` (pedregosa2011). Luego, se aplicó la normalización `MinMaxScaler`, que transforma cada variable al rango $(-1, 1)$. Esta normalización es esencial para redes neuronales recurrentes, ya que evita saturación en las funciones de activación y mejora la estabilidad numérica durante el entrenamiento (lara2021).

6.1.2. Creación de secuencias temporales

Con el fin de que el modelo LSTM aprenda patrones temporales, se transformó la serie original en un conjunto de entrenamiento supervisado utilizando una ventana deslizante de 60 pasos (horas). Cada ventana contiene 60 observaciones consecutivas de las tres variables de entrada, y la etiqueta corresponde al valor de demanda una hora después. Este procedimiento convierte la serie temporal en un tensor tridimensional con forma $(n, 60, 3)$, adecuado para el entrenamiento de redes LSTM.

```

1 def create_dataset(dataset, look_back=60):
2     X, Y = [], []
3     for i in range(len(dataset) - look_back):
4         a = dataset[i:(i + look_back), :]
5         X.append(a)
6         Y.append(dataset[i + look_back, 0])
7     return np.array(X), np.array(Y)
8
9 X_train, y_train = create_dataset(train_scaled, 60)

```

6.1.3. Definición del modelo LSTM

El modelo propuesto es una red Sequential compuesta por tres capas LSTM apiladas, con 100, 100 y 50 unidades respectivamente. Las dos primeras capas devuelven secuencias completas (`return_sequences=True`), mientras que la última solo entrega la salida final (`return_sequences=False`). Después de las dos primeras capas se aplica un Dropout del 20% para reducir el sobreajuste. Finalmente, se añade una capa densa (`Dense`) de salida con una neurona, que representa el valor de demanda estimado.

```

1 model = Sequential([
2     LSTM(100, input_shape=(X_train.shape[1], X_train.shape[2]),
3         return_sequences=True),
4     Dropout(0.2),
5     LSTM(100, return_sequences=True),
6     Dropout(0.2),
7     LSTM(50, return_sequences=False),
8     Dense(1)
9 ])

```

El modelo se compiló con la función de pérdida `mean_squared_error` y el optimizador Adam con tasa de aprendizaje 0.001. Según (Iara2021), esta combinación de hiperparámetros (learning rate de 10^{-3} y tamaño de lote de 32) ofrece resultados estables en problemas de predicción de series temporales con redes recurrentes.

```

1 model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
2 model.fit(X_train, y_train, epochs=80, batch_size=32, verbose=1)

```

6.1.4. Predicción para el año 2024

Para evaluar la capacidad de generalización del modelo, se utilizaron los datos reales de 2024 (de enero a octubre). Estos datos fueron preprocesados de manera idéntica al conjunto de entrenamiento, reutilizando los mismos objetos de imputación y escalado ajustados previamente para evitar fuga de información (hyndman2021).

```

1 real_data_2024 = train_data[
2     (train_data.index.year == 2024) & (train_data.index.month <= 10)
3 ]
4 real_data_2024 = real_data_2024[real_data_2024['Area'] == 'ORI'][[
5     'Estimacion_de_Demanda_por_Balance_(MWh)',
6     'Generacion_(MWh)',
7     'Intercambio_neto_entre_Gerencias_(MWh)'
8 ]]

```

Se formaron las secuencias necesarias para realizar las predicciones, concatenando las últimas 60 horas de 2023 con las primeras de 2024, garantizando la continuidad temporal del modelo.

6.1.5. Visualización de los resultados

Para comparar las predicciones con los valores reales de la demanda, se graficaron ambas series en el mismo eje temporal utilizando Matplotlib. La Figura 6.1 muestra la concordancia entre la demanda observada y la demanda predicha para el periodo enero–octubre 2024.

```

1 plt.figure(figsize=(20, 12))
2 plt.plot(real_data_2024.index, real_demand,
3         label='Datos_Reales_2024', color='blue', linewidth=1.8)
4 plt.plot(real_data_2024.index[:len(predictions)], predictions,
5         label='Predicciones_2024', color='red', linewidth=1)
6 plt.xlim(real_data_2024.index[0],
7         real_data_2024.index[len(predictions) - 1])
8 plt.margins(x=0)
9 plt.title('Predicciones_vs_Datos_Reales_ORI_Enero-Octubre_2024', fontsize=22)
10 plt.xlabel('Fecha', fontsize=18)
11 plt.ylabel('Estimacion_de_Demanda_por_Balance_(MWh)', fontsize=18)
12 plt.tick_params(axis='both', which='major', labelsize=14)
13 plt.xticks(rotation=30)
14 plt.legend(fontsize=16)
15 plt.grid(True)
16 plt.tight_layout()
17 plt.show()

```

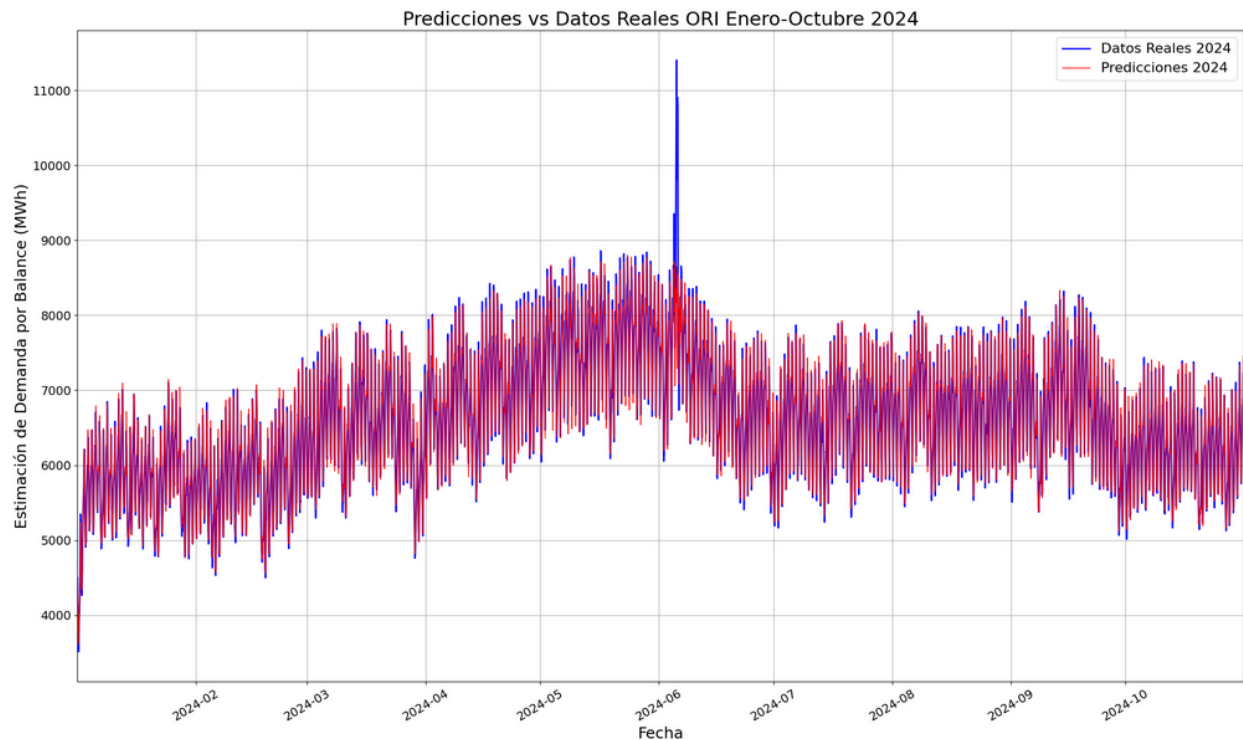


Figura 6.1: Comparación entre las predicciones del modelo LSTM y los datos reales de demanda en la región ORI (enero–octubre 2024).

6.1.6. Resultados obtenidos

Tabla 6.1: Coeficiente de determinación (R^2) mensual del modelo LSTM para el año 2024 en la región Oriente.

Mes	R^2
Enero	0.9883
Febrero	0.9878
Marzo	0.9900
Abril	0.9891
Mayo	0.9750
Junio	0.8545
Julio	0.9891
Agosto	0.9880
Septiembre	0.9895
Octubre	0.9860

El modelo LSTM mostró un desempeño satisfactorio, como se observa en la Tabla 6.1, obtuvo valores del coeficiente de determinación (R^2) superiores al 0.98 en la mayoría de los meses, indicando una alta capacidad de ajuste a los datos reales de demanda eléctrica. El desempeño disminuyó durante el mes de junio ($R^2 = 0.8545$), asociado a una mayor a condiciones atípicas de la red eléctrica durante dicho periodo. Este comportamiento coincide con lo documentado en **taylor2007** y **smyl2020**, donde modelos híbridos y recurrentes presentan un mejor ajuste en datos energéticos. Los resultados son consistentes con la capacidad de las redes LSTM para capturar dependencias

temporales de largo plazo en series eléctricas multivariantes.

Capítulo 7

Modelo XGBoost

7.1. Arquitectura del modelo XGBoost para la predicción de demanda eléctrica

7.1.1. Preparación de los datos

Para la construcción del modelo XGBoost se utilizaron los mismos datos horarios de la región Oriente del Sistema Eléctrico Nacional proporcionados por el CENACE, abarcando el periodo de 2020 a octubre de 2024. El conjunto contiene las siguientes variables: estimación de demanda por balance (variable objetivo), generación total, intercambio neto entre gerencias, importación total y exportación total.

$$\mathbf{Datos} = \begin{bmatrix} \text{Estimación de Demanda por Balance (MWh)} \\ \text{Generación (MWh)} \\ \text{Intercambio neto entre Gerencias (MWh)} \end{bmatrix} \quad (7.1)$$

El proceso inicia con la carga del archivo CSV mediante la biblioteca pandas, estableciendo la columna `Timestamp` como índice temporal. Posteriormente se filtra la región ORI y se homogenizan los valores erróneos o vacíos ('--', 'NaN', 'SIN').

Los valores faltantes se imputan con la media aritmética utilizando la clase `SimpleImputer` de `scikit-learn` (pedregosa2011), garantizando un conjunto completo de datos sin pérdida de información. Este preprocesamiento coincide con el descrito en la sección del modelo LSTM y asegura coherencia entre ambos enfoques.

```
1 def load_and_prepare_data(filepath):
2     data = pd.read_csv(filepath, index_col='Timestamp', parse_dates=True)
3     data = data[data['Area'] == 'ORI'][[
4         'Estimacion_de_Demanda_por_Balance_(MWh)',
5         'Generacion_(MWh)',
6         'Intercambio_netos_entre_Gerencias_(MWh)',
7         'Importacion_Total_(MWh)',
8         'Exportacion_Total_(MWh)'
9     ]]
10    data.replace(['--', 'NaN', '', ' ', '.....', 'SIN'], np.nan,
11                inplace=True)
12    imputer = SimpleImputer(strategy='mean')
```

```

12 data = pd.DataFrame(imputer.fit_transform(data), columns=data.columns,
13                      index=data.index)
13 return data

```

7.1.2. Creación de características temporales avanzadas

Una diferencia clave entre los modelos LSTM y XGBoost radica en la forma de representar la secuencia temporal. Mientras que las redes LSTM aprenden dependencias a partir de tensores tridimensionales, los modelos basados en árboles requieren un conjunto de variables tabulares. Por ello, se construyó un conjunto de características derivadas a partir de una ventana deslizante de 120 horas, equivalente a 5 días.

Cada ventana genera un vector de entrada que incluye:

- Los valores recientes de las 120 horas previas de todas las variables.
- Estadísticos de la ventana: media, desviación estándar, mínimos, máximos y cuantiles (10 %, 25 %, 75 % y 90 %).
- Una tendencia simple calculada como la diferencia entre el último y el primer valor de la ventana.
- Medias móviles de 24 y 168 horas, que reflejan estacionalidades diarias y semanales en la demanda eléctrica.
- Rezagos exactos de 24 y 168 horas para capturar repeticiones periódicas.

Este procedimiento, comúnmente denominado *sliding window*, transforma la serie temporal en un formato supervisado, donde cada registro resume información histórica previa sin incluir valores futuros, evitando así fuga de información (brownlee2018; hyndman2018).

```

1 def create_advanced_features(data, look_back=120):
2     features = []
3     target = []
4     n_features = data.shape[1]
5
6     for i in range(look_back, len(data)):
7         window = data.iloc[i-look_back:i]
8         current_row = data.iloc[i, 0]
9         recent_values = window.values.flatten()
10
11         stats = [
12             window.mean().values,
13             window.std().values,
14             window.min().values,
15             window.max().values,
16             window.quantile(0.1).values,
17             window.quantile(0.25).values,
18             window.quantile(0.75).values,
19             window.quantile(0.9).values,
20             (window.values[-1] - window.values[0]).flatten()
21         ]
22
23         moving_avg_24h = data.iloc[max(0, i-24):i].mean().values
24         moving_avg_168h = data.iloc[max(0, i-168):i].mean().values
25         lag_24h = data.iloc[i-24].to_numpy() if i >= 24 else np.zeros(
            n_features)

```



```

26     lag_168h = data.iloc[i-168].to_numpy() if i >= 168 else np.zeros(
27         n_features)
28
29     all_features = np.concatenate([
30         recent_values,
31         *[s.flatten() for s in stats],
32         moving_avg_24h.flatten(),
33         moving_avg_168h.flatten(),
34         lag_24h.flatten(),
35         lag_168h.flatten()
36     ])
37
38     features.append(all_features)
39     target.append(current_row)
40
41     return np.array(features), np.array(target)

```

7.1.3. Configuración y entrenamiento del modelo XGBoost

El modelo utilizado fue un XGBRegressor implementado mediante la biblioteca xgboost (chen2016). Este algoritmo pertenece a la familia de métodos de *gradient boosting*, en el cual múltiples árboles de decisión se construyen de manera secuencial, y cada árbol busca corregir los errores del anterior.

En este trabajo se establecieron los siguientes parámetros:

- `n_estimators = 600` y `learning_rate = 0.01`, lo que implementa una estrategia de *shrinkage* o aprendizaje progresivo.
- `max_depth = 10`, `subsample = 0.8` y `colsample_bytree = 0.8`, que controlan la complejidad del modelo y previenen el sobreajuste.
- `gamma = 0.2`, `reg_alpha = 0.1`, `reg_lambda = 1.0` y `min_child_weight = 3`, que añaden regularización a los pesos y divisiones de los árboles.
- `eval_metric = 'mae'`, que evalúa la desviación media absoluta y resulta más robusta ante valores atípicos (hastie2009).

```

1 def build_xgb_model():
2     return XGBRegressor(
3         n_estimators=600,
4         learning_rate=0.01,
5         max_depth=10,
6         subsample=0.8,
7         colsample_bytree=0.8,
8         gamma=0.2,
9         reg_alpha=0.1,
10        reg_lambda=1.0,
11        min_child_weight=3,
12        objective='reg:squarederror',
13        eval_metric='mae',
14        tree_method='hist',
15        random_state=42,
16        early_stopping_rounds=20
17    )

```

Se dividieron los datos en un 80 % para entrenamiento y un 20 % para prueba, respetando el orden cronológico de la serie. Del conjunto de entrenamiento se reservó un 10 % final para validación, empleado en el criterio de *early stopping*, que detiene el aprendizaje cuando la métrica de validación deja de mejorar, evitando así sobreajuste. Además, se aplicó un `RobustScaler` de `scikit-learn` para reducir la influencia de valores atípicos en las características generadas.

```

1 def train_and_evaluate(filepath):
2     data = load_and_prepare_data(filepath)
3     X, y = create_advanced_features(data, look_back=120)
4
5     split_idx = int(len(X) * 0.8)
6     X_train, X_test = X[:split_idx], X[split_idx:]
7     y_train, y_test = y[:split_idx], y[split_idx:]
8
9     val_split = int(len(X_train) * 0.9)
10    X_val, y_val = X_train[val_split:], y_train[val_split:]
11    X_train, y_train = X_train[:val_split], y_train[:val_split]
12
13    scaler = RobustScaler()
14    X_train = scaler.fit_transform(X_train)
15    X_val = scaler.transform(X_val)
16    X_test = scaler.transform(X_test)
17
18    model = build_xgb_model()
19    model.fit(X_train, y_train, eval_set=[(X_val, y_val)], verbose=50)

```

7.1.4. Evaluación del modelo

Tras el entrenamiento, se generaron predicciones sobre el conjunto de prueba (correspondiente al año 2024). El desempeño del modelo se evaluó mediante el coeficiente de determinación R^2 , calculado de forma mensual para analizar la estabilidad del modelo frente a variaciones estacionales.

```

1 y_pred = model.predict(X_test)
2 test_dates = data.index[120+split_idx:]
3 months = test_dates.month
4
5 r2_results = []
6 for month in range(1, 11):
7     mask = (months == month)
8     if np.sum(mask) > 0:
9         r2 = r2_score(y_test[mask], y_pred[mask])
10        r2_results.append({'Mes': month, 'R-squared': r2})
11        print(f"R_{mes_{month}}: {r2:.4f}")

```

7.1.5. Visualización de resultados

Se generó una gráfica comparativa entre los valores reales de demanda eléctrica y las predicciones del modelo XGBoost para el periodo de prueba correspondiente a enero–octubre de 2024. Para ello, se empleó la biblioteca `matplotlib` y se aplicó una configuración visual optimizada que permite una apreciación más clara de las series.

```

1 plt.figure(figsize=(20, 12))
2 plt.plot(test_dates, y_test, label='Datos Reales 2024', color='blue', linewidth=1.8)

```

```

3 plt.plot(test_dates[:len(y_pred)], y_pred, label='Predicciones_XGBoost_2024',
4          color='red', linewidth=1)
5 plt.xlim(test_dates[0], test_dates[len(y_pred) - 1])
6 plt.margins(x=0)
7 plt.title('Predicciones_XGBoost_vs_Datos_Reales_ORI_Enero-Octubre_2024',
8           fontsize=22)
9 plt.xlabel('Fecha', fontsize=18)
10 plt.ylabel('Estimación de Demanda por Balance (MWh)', fontsize=18)
11 plt.tick_params(axis='both', which='major', labelsize=14)
12 plt.xticks(rotation=30)
13 plt.legend(fontsize=16)
14 plt.grid(True)
15 plt.tight_layout()
16 plt.show()

```

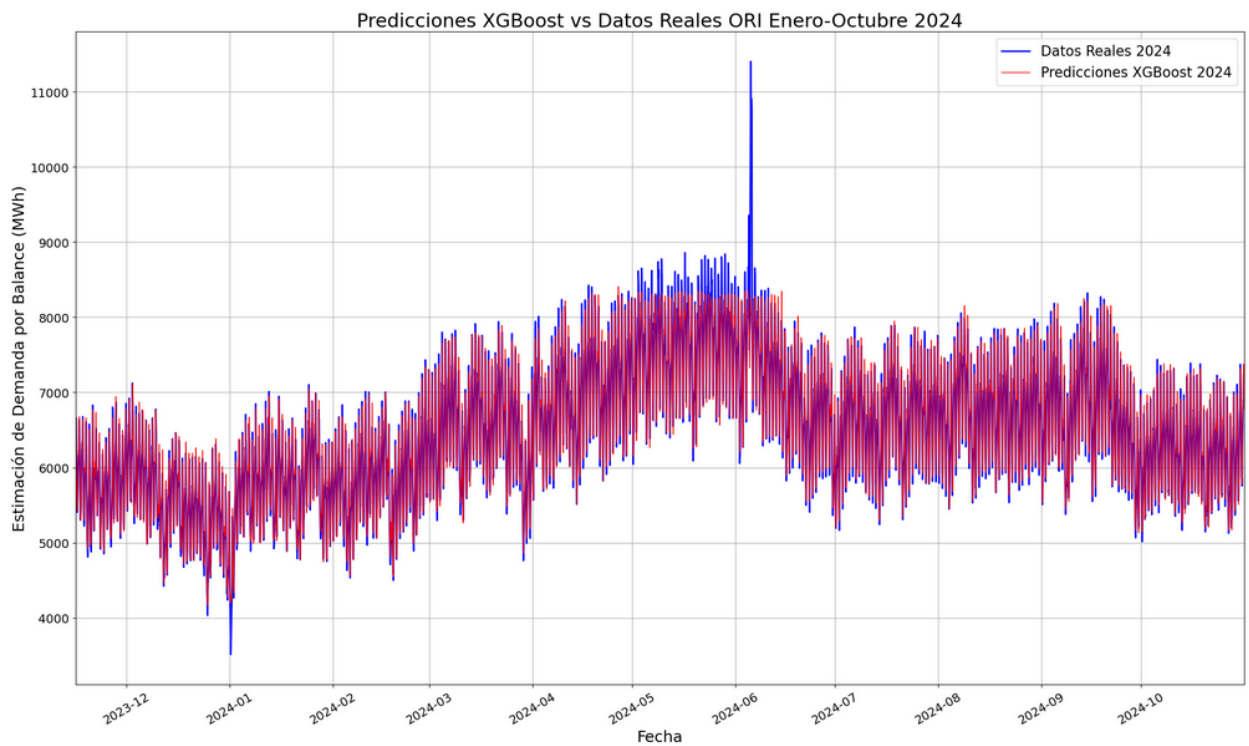


Figura 7.1: Comparación entre la demanda real y la predicha por el modelo XGBoost en la región ORI (enero–octubre 2024).

7.1.6. Resultados obtenidos

Tabla 7.1: Coeficiente de determinación (R^2) mensual del modelo XGBoost para el año 2024 en la región Oriente.

Mes	R^2
Enero	0.9740
Febrero	0.9846
Marzo	0.9851
Abril	0.9749
Mayo	0.9209
Junio	0.8111
Julio	0.9834
Agosto	0.9834
Septiembre	0.9827
Octubre	0.9837

El modelo XGBoost logró un desempeño sobresaliente, con valores del coeficiente de determinación superiores a 0.97 en la mayoría de los meses analizados. Los meses de mayo y junio presentan una ligera disminución del ajuste, posiblemente asociada a variaciones abruptas en la demanda eléctrica o eventos atípicos en la red. Estos resultados evidencian la capacidad del modelo para capturar patrones no lineales y estacionales a partir de un conjunto de características ingenierizadas, ofreciendo un desempeño comparable al obtenido con la red LSTM.

La metodología implementada, basada en un preprocesamiento consistente, la generación de variables temporales avanzadas y una validación temporal adecuada, garantiza la fiabilidad y replicabilidad de los resultados (**chen2016**; **hyndman2018**; **brownlee2018**).

Capítulo 8

Discusión

8.1. Comparación y coherencia entre los modelos LSTM y XGBoost

8.1.1. Metodología y coherencia en el tratamiento de los datos

El flujo de preprocesamiento y validación fue diseñado para mantener consistencia entre ambos modelos:

- Se aplicó el mismo proceso de limpieza, imputación de valores faltantes y normalización de los datos, garantizando la homogeneidad del conjunto de entrada.
- En ambos casos se utilizó un esquema de validación temporal, evitando la fuga de información entre conjuntos de entrenamiento y prueba.
- Los modelos fueron entrenados y evaluados sobre las mismas variables (demanda, generación, intercambio neto, importación y exportación), lo que permite una comparación directa de resultados.

8.1.2. Comparación del desempeño

Ambos modelos demostraron un alto nivel de ajuste respecto a la demanda real, con valores promedio de R^2 superiores a 0.97 durante el periodo de prueba 2024. El modelo LSTM obtuvo un R^2 promedio de 0.987, mientras que el modelo XGBoost alcanzó un valor promedio de 0.964, mostrando un desempeño muy competitivo pese a su naturaleza no secuencial.

8.1.3. Rendimiento computacional de los modelos

Ambos modelos fueron entrenados bajo las mismas condiciones de hardware para asegurar una comparación consistente. En particular, se empleó una GPU *NVIDIA Tesla P100-PCIE* de 16 GB, donde el modelo LSTM ocupó aproximadamente 15.5 GB de memoria durante el entrenamiento, mientras que XGBoost utilizó menos de 0.4 GB. Esta diferencia refleja la naturaleza de cada algoritmo y los recursos que requieren para su proceso de optimización.

Tiempo de entrenamiento

El modelo XGBoost mostró un tiempo de entrenamiento significativamente menor. El proceso completo de ajuste del modelo tomó:

- **136.11 segundos**, equivalentes a **2.27 minutos**.

Por otro lado, el modelo LSTM requirió 80 épocas. Cada una tuvo duración aproximada de 67 segundos, lo que resulta en un tiempo total cercano a:

- $80 \times 67 \text{ s} \approx 5,360 \text{ segundos}$, es decir, **89.3 minutos**.

Uso de memoria

Durante el entrenamiento, el modelo LSTM alcanzó un uso aproximado de 15.5 GB de memoria GPU, lo cual se debe al manejo de secuencias largas y al almacenamiento de estados ocultos. En contraste, XGBoost utilizó menos de 0.5 GB, dado que su entrenamiento no requiere almacenar estados temporales.

Estos resultados son coherentes con lo expuesto por Hastie, Tibshirani y Friedman (**hastie2009**), quienes señalan que los métodos de *boosting* son altamente eficientes en memoria y escalan adecuadamente para conjuntos de datos de tamaño moderado.

Comparación de la energía total estimada

A pesar de las diferencias computacionales, ambos modelos presentaron predicciones coherentes al comparar la energía total del periodo de prueba. La Tabla 8.1 resume los resultados más relevantes. Es importante mencionar que la diferencia en la energía real total utilizada por cada modelo se debe a que emplean longitudes de ventana distintas (60 horas para LSTM y 120 horas para XGBoost), lo cual modifica ligeramente la cantidad de observaciones disponibles en la etapa de prueba.

Tabla 8.1: Comparación del rendimiento computacional entre los modelos LSTM y XGBoost.

Aspecto	LSTM	XGBoost
Tiempo de entrenamiento	89.3 minutos	2.27 minutos
Uso de memoria GPU	~15.5 GB	< 0.5 GB
Energía real (MWh)	48,366,874.33	54,612,937.34
Energía predicha (MWh)	48,298,021.40	54,496,376.00
Error relativo	-0.14 %	-0.21 %
Hardware utilizado	GPU NVIDIA Tesla P100-PCIE 16 GB	

La cercanía entre la energía real y la energía estimada indica que ambos modelos capturan adecuadamente la tendencia global de la demanda eléctrica, aunque presenten diferencias en la calidad predictiva a corto plazo.

Consideraciones prácticas

Desde una perspectiva operativa, la elección entre ambos modelos depende tanto de la precisión deseada como de los recursos computacionales disponibles. El modelo LSTM presenta una mayor capacidad para aprender dependencias temporales complejas; sin embargo, requiere:

- mayor tiempo de entrenamiento,
- mayor capacidad de memoria,
- y hardware especializado, como unidades GPU.

Por otro lado, XGBoost es considerablemente más rápido y ligero, pudiendo ejecutarse eficientemente incluso sin aceleración por GPU. Esto lo convierte en una alternativa adecuada para ambientes donde se requiere un reentrenamiento frecuente o una integración continua en sistemas industriales.

8.1.4. Conclusiones comparativas

La comparación entre los modelos LSTM y XGBoost permite destacar las siguientes observaciones:

1. **Ambos modelos presentan coherencia metodológica.** La coincidencia en los resultados confirma que el preprocesamiento, la imputación de datos y el esquema de validación fueron correctamente diseñados, asegurando comparabilidad y solidez experimental.
2. **Ambos enfoques son complementarios.** El modelo LSTM resulta más adecuado para capturar dinámicas temporales prolongadas, mientras que XGBoost ofrece interpretabilidad, eficiencia computacional y un excelente rendimiento en horizontes de corto plazo. Su desempeño conjunto muestra que ambos modelos ofrecen perspectivas útiles dentro del pronóstico energético.

En conclusión, los resultados obtenidos demuestran que tanto las redes neuronales recurrentes (LSTM) como los modelos de árboles potenciados (XGBoost) son herramientas eficaces para el pronóstico de la demanda eléctrica, siempre que se acompañen de un adecuado procesamiento de datos, una validación temporal correcta y una representación coherente de las relaciones temporales y exógenas del sistema eléctrico.

8.2. Ventajas

- **Modelado de relaciones no lineales:** Tanto LSTM como XGBoost permiten representar comportamientos no lineales de la demanda eléctrica, superando las limitaciones de los métodos estadísticos tradicionales.
- **LSTM — aprendizaje temporal profundo:** Su arquitectura basada en memoria permite capturar patrones horarios, diarios y estacionales, representando dependencias de corto y largo plazo sin necesidad de especificarlas manualmente.
- **XGBoost — eficiencia computacional:** Su capacidad de paralelización y optimización interna reduce significativamente los tiempos de entrenamiento, lo que lo hace adecuado para aplicaciones operativas con grandes volúmenes de datos.
- **Interpretabilidad del modelo:** XGBoost permite analizar la importancia de las variables y entender el efecto de cada característica en la predicción; LSTM, aunque menos interpretable, ofrece una representación más completa de la dinámica temporal.
- **Robustez ante ruido y datos faltantes:** XGBoost maneja valores nulos sin necesidad de imputaciones complejas, mientras que LSTM mantiene estabilidad mediante normalización y regularización.
- **Generalización y ajuste:** Ambos modelos incorporan mecanismos para prevenir el sobreajuste, como regularización L1/L2 en XGBoost y técnicas como *dropout* y *early stopping* en LSTM.
- **Integración práctica:** Los dos modelos pueden implementarse de forma sencilla utilizando librerías estándar de Python, facilitando su uso en flujos operativos o dashboards energéticos.

8.3. Limitaciones

- **LSTM — alta demanda computacional:** Su entrenamiento requiere recursos significativos, especialmente en arquitecturas profundas.
- **XGBoost — dependencia de la ingeniería de características:** Su desempeño depende de la calidad de las variables rezagadas y ventanas temporales diseñadas por el analista.
- **LSTM — sensibilidad a hiperparámetros:** Elecciones como el tamaño de la ventana temporal, número de neuronas o tasa de aprendizaje influyen directamente en el rendimiento.
- **XGBoost — representación limitada de dependencias secuenciales:** No captura la dinámica temporal de forma implícita, por lo que puede perder información entre transiciones si las características no están bien diseñadas.
- **Interpretabilidad reducida en LSTM:** Su naturaleza de “caja negra” dificulta explicar qué patrones concretos influyen en la predicción.
- **Dependencia de datos limpios:** Ambos modelos requieren series temporales completas y consistentes; errores sistemáticos pueden degradar el rendimiento.
- **Riesgo de sobreajuste:** Puede presentarse en LSTM sin regularización adecuada o en XGBoost cuando se usan demasiados árboles o gran profundidad.

Capítulo 9

Conclusiones

El desarrollo de modelos de predicción de la demanda eléctrica mediante técnicas de aprendizaje automático demostró que los métodos modernos son capaces de representar con mayor facilidad la complejidad de las series temporales energéticas. A diferencia de los enfoques estadísticos tradicionales, que requieren supuestos estrictos sobre la estructura de los datos, los modelos basados en aprendizaje automático permiten capturar relaciones no lineales, interacciones entre variables y patrones temporales que reflejan de manera más realista la dinámica del sistema eléctrico en la región Oriente de México.

Se implementaron dos aproximaciones complementarias: una red neuronal recurrente del tipo LSTM y un modelo basado en árboles potenciados (XGBoost). Ambos modelos mostraron un desempeño consistente durante el periodo de análisis, lo que confirma que los datos históricos del CENACE contienen información suficiente para anticipar patrones de comportamiento en la demanda eléctrica bajo esquemas predictivos adecuados.

Las diferencias metodológicas entre ambos enfoques reflejan fortalezas específicas. Las redes neuronales recurrentes LSTM destacan por su habilidad para conservar información relevante a lo largo de secuencias temporales extensas mediante mecanismos de memoria y puertas de control, lo cual resulta especialmente útil en sistemas eléctricos donde intervienen ciclos diarios, semanales y estacionales. Por otro lado, XGBoost demostró ser un modelo eficiente y robusto frente a variaciones locales y valores atípicos, aprovechando la ingeniería de características y la regularización inherente a los métodos de ensamble.

La comparación realizada sugiere que no existe un único modelo superior, sino que ambos enfoques aportan ventajas distintas y potencialmente complementarias. Mientras el LSTM ofrece una representación más directa de la dinámica secuencial, XGBoost destaca por su rapidez, facilidad de ajuste y bajo costo computacional. En concordancia con lo señalado por Brownlee (**brownlee2018**), estos métodos pueden integrarse en esquemas híbridos que combinen la capacidad de las redes neuronales para capturar dependencias temporales con la fortaleza de los modelos basados en árboles para explotar características derivadas.

Finalmente, los resultados evidenciaron que ambos modelos pueden coexistir de manera complementaria. Mientras el LSTM sobresale en la detección automática de dependencias secuenciales, XGBoost se mantiene como una alternativa eficiente para escenarios donde se requiere rapidez y facilidad de implementación. Su integración futura podría dar lugar a modelos híbridos que combinen las ventajas del aprendizaje profundo y los métodos de ensamble.

El uso de redes neuronales y técnicas de aprendizaje automático en la predicción de la demanda

eléctrica representa un avance sustancial hacia la modernización del sistema energético nacional. Su adopción permitiría al CENACE y a otras instituciones mejorar la planificación operativa, reducir pérdidas y fortalecer la estabilidad del sistema eléctrico mexicano en el marco de la transición energética.

Apéndice A

Código de descarga y preparación de datos

Eliminación de las primeras 8 filas de los archivos CSV

```
1 for file in "$directory"/*.csv; do
2     tail -n +9 "$file" > "${file%.csv}_clean.csv"
3 done
```

Listing A.1: Eliminación de filas iniciales en los archivos CSV

Extracción de la fecha desde el nombre del archivo

```
1 for file in "$directory"/*_clean.csv; do
2     if [ -f "$file" ]; then
3         chmod 644 "$file"
4
5         # Extraer fecha en formato YYYY-MM-DD
6         date=$(basename "$file" | grep -oP '\d{4}-\d{2}-\d{2}')
7
8         # Alternativa para archivos con formato distinto
9         if [ -z "$date" ]; then
10             date=$(basename "$file" | grep -oP '(?<=Dia_Operacion_)\d{4}-\d{2}-\d{2}')
11         fi
12
13         # Agregar la fecha como nueva columna
14         awk -v date="$date" 'BEGIN{FS=OFS=","}{
15             if(NR==1)_$0=$0",Fecha";
16             else_$0=$0",date;
17             }1' "$file" > "${file%.csv}_date.csv"
18     else
19         echo "Archivo_$file_no_encontrado_o_sin_permisos."
20     fi
21 done
22 echo "Fecha_extra da_y_agregada_a_cada_archivo."
```

Listing A.2: Extracción de la fecha e incorporación como columna

Unificación de archivos

```
1 output_file="$directory/merged_data.csv"
2
3 # Tomar encabezados del primer archivo
4 head -n 1 "$(ls "$directory"/*_date.csv | head -n 1)" > "$output_file"
5
6 # Concatenar contenido de todos los archivos
7 for file in "$directory"/*_date.csv; do
8     tail -n +2 "$file" >> "$output_file"
9 done
10
11 echo "Todos los archivos fueron unidos en $output_file."
```

Listing A.3: Unificación en un archivo maestro

Creación de la columna **Timestamp**

```
1 merged_df['Hora'] = pd.to_numeric(merged_df['Hora'], errors='coerce')
2
3 # Valores fuera de rango
4 merged_df.loc[merged_df['Hora'] > 23, 'Hora'] = 23
5 merged_df.loc[merged_df['Hora'] < 0, 'Hora'] = 0
6
7 # Formato de dos dígitos
8 merged_df['Hora'] = merged_df['Hora'].astype(int).astype(str).str.zfill(2)
9
10 # Timestamp completo
11 merged_df['Timestamp'] = pd.to_datetime(
12     merged_df['Fecha'] + '_' + merged_df['Hora'] + ':00:01'
13 )
```

Listing A.4: Construcción de la columna Timestamp en el archivo unificado

Bibliografía

- [1] Centro Nacional de Control de Energía (CENACE). *Gráfica de Demanda*. Disponible en <https://www.cenace.gob.mx/graficademanda.aspx>. [Consultado: 4 de abril de 2024].
- [2] Centro Nacional de Control de Energía (CENACE). *Estimación de Demanda Real*. Disponible en <https://www.cenace.gob.mx/Paginas/SIM/Reportes/EstimacionDemandaReal.aspx>. [Consultado: 4 de abril de 2024].
- [3] Centro Nacional de Control de Energía (CENACE). *Pronósticos de Demanda*. Disponible en <https://www.cenace.gob.mx/Paginas/SIM/Reportes/PronosticosDemanda.aspx>. [Consultado: 4 de abril de 2024].
- [4] Centro Nacional de Control de Energía (CENACE). *¿Quiénes somos? Funciones y atribuciones del CENACE*. Disponible en <https://www.cenace.gob.mx/>. [Consultado: 1 de abril de 2024].
- [5] Secretaría de Energía (SENER). (2023). *Prospectiva del Sector Eléctrico 2023*. Gobierno de México. Disponible en <https://www.gob.mx/sener>.
- [6] Secretaría de Energía (SENER). (2023). *Programa de Desarrollo del Sistema Eléctrico Nacional 2023–2037 (PRODESEN)*. Gobierno de México.
- [7] Secretaría de Energía (SENER). (2024). *Programa de Desarrollo del Sistema Eléctrico Nacional 2024–2038 (PRODESEN)*. Gobierno de México.
- [8] International Energy Agency (IEA). (2022). *Electricity Market Report 2022*. IEA, Paris.
- [9] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5a ed.). Wiley.
- [10] Chatfield, C. (2003). *The Analysis of Time Series: An Introduction* (6a ed.). Chapman & Hall/CRC.
- [11] Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.
- [12] Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples* (4a ed.). Springer.
- [13] Taylor, J. W., & McSharry, P. E. (2007). *Short-Term Load Forecasting Methods: An Evaluation Based on European Data*. *IEEE Transactions on Power Systems*, 22(4), 2213–2219.
- [14] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2a ed.). OTexts, Melbourne.
- [15] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [16] Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. *Neural Computation*, 9(8), 1735–1780.

- [17] Lara-Benítez, P., Carranza-García, M., & Riquelme, J. C. (2021). *An Experimental Review on Deep Learning Architectures for Time Series Forecasting*. arXiv preprint arXiv:2103.12057.
- [18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. *Journal of Machine Learning Research*, 12, 2825–2830.
- [19] Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. En *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.
- [20] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2a ed.). Springer.
- [21] Smyl, S. (2020). *A Hybrid Method of Exponential Smoothing and Recurrent Neural Networks for Time Series Forecasting*. *International Journal of Forecasting*, 36(1), 75–85.
- [22] Brownlee, J. (2018). *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, Australia.
- [23] Andrews, D. (2016). *Referencia citada en el contexto del pronóstico de demanda eléctrica*. [Detalles por complementar por el autor].
- [24] Dong, Q., Huang, R., Cui, C., Towey, D., Zhou, L., Tian, J., & Wang, J. (2025). *Short-Term Electricity-Load Forecasting by Deep Learning: A Comprehensive Survey*. *Journal of the ACM*. arXiv:2408.16202.