EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

"DESARROLLO DE UN MODELO INTELIGENTE PARA PREDECIR EL PRECIO MARGINAL LOCAL (PML) EN UN NODO DE LA RED ELÉCTRICA NACIONAL"

QUE PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA INDUSTRIAL

PRESENTA
I.I. Marcos Fidel Guzmán Escobar

DIRECTOR DE TESIS
Dr. Alberto Alfonso Aguilar Lasserre

CODIRECTOR DE TESIS
M.I.I. Marco Julio Del Moral Argumedo

CODIRECTOR DE TESIS
M.I.I. Nicasio Hernández Flores

ORIZABA, VERACRUZ, MÉXICO **ABRIL 2024**

Orizaba, Veracruz, 20/marzo/2024
Dependencia: **División de Estudios de Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: **I**

C. GUZMÁN ESCOBAR MARCOS FIDEL
Candidato(a) a Grado de Maestro(a) en:
INGENIERÍA INDUSTRIAL
PRESENTE

De acuerdo con el reglamento de Titulación vigente de los Centros e Institutos Tecnológicos Federales del Tecnológico Nacional de México, de la Secretaría de Educación Pública, y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

"DESARROLLO DE UN MODELO INTELIGENTE PARA PREDECIR EL PRECIO MARGINAL LOCAL (PML) EN UN NODO DE LA RED ELÉCTRICA NACIONAL"

Comunico a usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
CIENCIA – TÉCNICA - CULTURA®

OFELIA LANDETA ESCAMILLA
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN

EDUCACIÓN SECRETARÍA DE EDUCACIÓN PÚBLICA | TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLOGICO DE ORIZABA
DIV. DE EST. DE POSGRADO E INVEST.

CSR/magh

OG-13-F06

Orizaba Veracruz, 22/febrero/2024
Asunto: **Revisión de trabajo escrito**

C. OFELIA LANDETA ESCAMILLA
ENCARGADA DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
P R E S E N T E.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.:

GUZMÁN ESCOBAR MARCOS FIDEL

La cual lleva el título de:

DESARROLLO DE UN MODELO INTELIGENTE PARA PREDECIR EL PRECIO MARGINAL LOCAL (PML) EN UN NODO DE LA RED ELÉCTRICA NACIONAL

y concluyen que se acepta.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
CIENCIA – TÉCNICA - CULTURA®

PRESIDENTE:	**DR. ALBERTO AGUILAR LASSERRE**	FIRMA
SECRETARIO:	**MII JULIO DEL MORAL ARGUMEDO**	FIRMA
VOCAL:	**MII NICASIO HERNÁNDEZ FLORES**	FIRMA
VOCAL SUP.:	**MII CONSTANTINO GERARDO MORAS SÁNCHEZ**	FIRMA

TA-09-F18

Declaración de originalidad y cesión de derechos

Orizaba, Veracruz, el día 18 del mes de abril del año 2024.

El que suscribe:

C. **Marcos Fidel Guzmán Escobar**

Declaro que esta tesis, que tiene una extensión de 205 hojas, cuartillas, ha sido escrita por mí y constituye el registro escrito del trabajo de la tesis titulada

"DESARROLLO DE UN MODELO INTELIGENTE PARA PREDECIR EL PRECIO MARGINAL LOCAL (PML) EN UN NODO DE LA RED ELÉCTRICA NACIONAL"

del programa: Maestría en Ingeniería Industrial, bajo la asesoría y dirección del Dr. Alberto Alfonso Aguilar Lasserre como director de tesis, el codirector Dr. Marco Julio Del Moral Argumedo y el codirector externo el Mtro. Nicasio Hernández Flores, y no ha sido sometida en ninguna otra institución previamente.

Todos los datos y las referencias a materiales ya publicados están debidamente identificados con su respectivo crédito e incluidos en las notas bibliográficas y en las citas que se destacan como tal y, en los casos que así lo requieran, cuento con las debidas autorizaciones de quienes poseen los derechos patrimoniales. Por lo tanto, me hago responsable de cualquier litigio o reclamación relacionada con derechos de propiedad intelectual, exonerando de toda responsabilidad al Tecnológico Nacional de México campus Orizaba.

También declaro que, al presentar esta tesis, cedo los derechos del trabajo al Tecnológico Nacional de México campus Orizaba para su difusión, con fines académicos y de investigación, bajo las regulaciones propias de la institución y que si existe algún acuerdo de confidencialidad de la información lo haré saber en forma escrita para que se omitan las secciones correspondientes.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: depi_orizaba@tecnm.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente de este.

Marcos Fidel Guzmán Escobar

Nombre y firma

AGRADECIMIENTOS

Esta sección es de suma importancia ya que me brinda la oportunidad de expresar mi profundo reconocimiento a todas aquellas personas que han contribuido de manera significativa durante esta etapa revolucionaria de mi vida, tanto en el desarrollo del proyecto como en la consecución de mi grado de maestría en ingeniería industrial. Si bien es cierto que cada individuo logra alcanzar aquello que se propone por su propia voluntad, motivado por necesidades, objetivos y deseos, también es importante reconocer el papel fundamental que juegan las personas significativas en este proceso. Aunque hay muchas personas más que podrían ser mencionadas en esta sección, he decidido destacar a aquellas que han estado más cercanas a mí y han dejado una huella profunda en mi camino. Sin embargo, quiero dejar constancia de que cada contribución, por más pequeña que parezca, ha sido valorada y apreciada en su justa medida.

En primer lugar, esta Dios, por ser quien hace posible todo en mi vida y sin Él no podría concebirla siquiera, pues es Él quien me lleva por caminos y me regresa con bien sumando significativamente en todos los aspectos de mi ser, permitiéndome desarrollarme y contribuir en este mundo.

Merecidamente, agradezco por mucho y por siempre a mi hermosa esposa Nidia Rodríguez Mazahua; gracias por tus consejos, tu apoyo en todos los sentidos, por tu compañía y lo mucho que representas, te conviertes en el más grande de mis intereses personales para proteger y apoyar en la vida. ¡Gracias, amor por todo cuanto sacrificas por mí y lo importante que soy para ti! La vida nos ha bendecido con una hermosa historia juntos y vamos por buen camino porque así lo decidimos, porque existe amor, respeto y responsabilidad para construir juntos; recuerdo bien que la idea de hacer la maestría la considere realmente por ti, y si alguien sabe lo que cuesta y lo que pase para lograrlo... eres tú, no podría haber sido más afortunado y te agradeceré siempre cada gesto, critica y compañía recibidas por ti, por lo que este logro también lo compartimos y quedará de testimonio de cómo se forman los buenos matrimonios.

A mi mamá por sus sacrificios en el momento más vulnerable de mi vida que fue cuando era niño y que con su ejemplo pude ser la persona que soy integra y trabajadora que no busca el mal sino el bien. Su buen ejemplo y trabajo dejan huella imborrable en mi consciencia, logrando

en mí una razón más para ser feliz y sentirme muy afortunado.

A mis hermanos (Guadalupe, Jesús y Alan) muchas gracias porque uno sabe lo mucho que nos queremos ver bien, sanos y felices, esforzándonos por lograr cosas significativas en nuestras vidas, y porque sé lo importante que son para mí, como yo para ustedes.

Gracias a mi cuñada y suegra porque mucho de este proceso se consiguió con el apoyo de cosas materiales que se necesitan y ayudan para las cosas intangibles como la estabilidad y la seguridad; que pueden hacer que uno se concentre más en los asuntos académicos como de las necesidades del hogar de manera más eficientemente.

Deseo agradecer a mi director de tesis, el Dr. Alberto Alfonso Aguilar Lasserre por su orientación experta, paciencia y consideración. Sobre todo, agradezco que aceptara ser mi director como también el permitirme trabajar en mis ideas y supervisarlas correctamente.

Asimismo, quiero agradecer a mis profesores y profesoras por compartir sus conocimientos y experiencias, así como por su constante motivación por enseñar externándome de cierta manera su pensar y consideración. Particularmente a mis codirectores de tesis el Dr. Marco Julio y el Mtro. Nicasio Hernández ya que sus consejos y retroalimentación han sido invaluables para el desarrollo de este proyecto. Al Mtro. Magno Ángel porque hace mucho trabajo en la coordinación de la maestría y que reconozco que es una ayuda invaluable también.

Agradezco al Instituto Nacional de Electricidad y Energías Limpias (INEEL) por su colaboración con el TecNM campus Orizaba en la autorización del proyecto, así como al Dr. Gustavo Arroyo por su colaboración con nosotros.

Por último, quiero reconocer el apoyo financiero proporcionado por CONACYT, que ha hecho posible la realización de este proyecto de investigación; el apoyo económico resulta ser tan fuerte en este mundo que no se puede prescindir del financiamiento para una maestría tan importante y difícil de cumplir.

A todas estas personas, mi más profundo agradecimiento.

DEDICATORIAS

El presente proyecto lo dedico a los jóvenes estudiantes y autodidactas que quieren conocer el enfoque del Machine Learning, series de tiempo y pronósticos realizados de manera robusta e integral, pues es una obra aprobada por expertos de la materia (mencionados en los agradecimientos) y que me llevó más de un año conseguir su versión final con un enfoque local y práctico que puede facilitar la asimilación de los conceptos dentro de la inteligencia artificial mediante código Python y particularmente en pronósticos respaldados por la metódica dirección del aprendizaje automático y tecnología disponibles en México al año 2024.

El proyecto ayudará como ejemplar de buenas prácticas de machine Learning para problemas de regresión en aprendizaje supervisado y cada sección está enfocada a cubrir de manera actualizada los temas de revisión de la literatura, marco teórico, metodología y resultados con el mejor de las explicaciones que a mi criterio se puede encontrar, por lo que recomiendo usarlos como ejemplo de enseñanzas de nivel superior principalmente y como referente para futuros trabajos también.

Dedicaré también el trabajo realizado a la ciencia, porque se abordó un problema complejo que aún estaba sin solucionar satisfactoriamente, pues el enfoque de la estadística y algunos softwares tienen sus limitaciones cuando no cuentan con la bondad de la inteligencia artificial y fue mediante el enfoque de redes neuronales recurrentes que se desarrollan en este proyecto que se soluciona el problema de pronóstico de una serie de tiempo muy volátil y cambiante por su diferente localidad en el país. El aporte a la literatura con el método propuesto sienta las bases de un modelo eficiente, integral y acreditado que puede implementarse con cobertura nacional y lograr resultados que pueden revolucionar el enfoque contemporáneo que se tiene y dirigirlo hacia los siguientes niveles de eficiencia.

Finalmente dedicaré también el proyecto a mis familiares y seres queridos.

RESUMEN

El Precio Marginal Local (PML) representa un componente crucial en la gestión y planificación del sistema eléctrico de cualquier país, en el caso de México, con sus tres sistemas de interconexión y una diversidad geográfica y climática significativa, el pronóstico del PML no solo es un desafío sino también una necesidad imperante para el mercado eléctrico mayorista. Este indicador es esencial para comprender la dinámica del mercado energético y para impulsar proyectos de infraestructura que satisfagan las necesidades energéticas tanto actuales como futuras del país.

Esta investigación se centró en analizar 28 nodos representativos de los tres sistemas de interconexión del país y establecer una base sólida para el desarrollo de pronósticos precisos; para abordar la tarea de pronosticar el PML, se emplearon cinco algoritmos de vanguardia: Árboles de Decisión para Regresión, Regresión Polinomial de Segundo Grado, SARIMA, LSTM y una combinación de Prophet con LSTM Secuencial. Los resultados obtenidos fueron prometedores, logrando errores porcentuales medios absolutos (MAPE) en un rango de 42% a 23%. También se incluyen métricas como el Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE), para medir la eficacia de los algoritmos utilizados en el pronóstico del PML.

La metodología desarrollada, combina el análisis detallado, la integración de datos climáticos y el uso de algoritmos avanzados, para erigir un modelo replicable y robusto. Esta investigación proporciona un referente valioso para futuras investigaciones y aplicaciones prácticas en el ámbito del pronóstico energético en México.

Finalmente, se desarrolló una interfaz gráfica de usuario utilizando Python y Tkinter denominada “PML Explorer”, que permite a los usuarios visualizar los datos históricos de los nodos, ofreciendo una gráfica dinámica para seleccionar rangos de fechas específicos y guardarlos. Posteriormente, el usuario puede ejecutar el código de Python para obtener los pronósticos a 48 horas hacia el futuro generados por los cinco algoritmos.

ASTRACT

The Local Marginal Price (PML) represents a crucial component in the management and planning of the electrical system in any country. In the case of Mexico, with its three interconnection systems and significant geographical and climatic diversity, forecasting the PML is not only a challenge but also an imperative need for the wholesale electricity market. This indicator is essential for understanding the dynamics of the energy market and for driving infrastructure projects that meet the current and future energy needs of the country.

This research focused on analyzing 28 representative nodes of the country's three interconnection systems and establishing a solid foundation for the development of accurate forecasts. To address the task of forecasting the PML, five cutting-edge algorithms were employed: Decision Trees for Regression, Second Degree Polynomial Regression, SARIMA, LSTM, and a combination of Prophet with Sequential LSTM. The results obtained were promising, achieving Mean Absolute Percentage Errors (MAPE) in a range of 42% to 23%. Other metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) were also included to address the effectiveness of the algorithms used in forecasting the PML.

The developed methodology, which combines detailed analysis, the integration of climatic data, and the use of advanced algorithms, establishes a replicable and robust model. This research provides a valuable reference for future research and practical applications in the field of energy forecasting in Mexico.

Finally, a user interface called “PML Explorer” was developed using Python and Tkinter, which allows users to visualize the historical data of the nodes. It offers a dynamic graph to select specific date ranges and save them. Subsequently, users can run the Python code to obtain forecasts 48 hours into the future generated by the five algorithms.

ÍNDICE

AGRADECIMIENTOS i
DEDICATORIAS vi
RESUMEN vii
ASTRACT viii
INTRODUCCIÓN 1
CAPÍTULO I. GENERALIDADES 4
1.1 Introducción 4
1.2 Planteamiento del problema 4
1.3 Objetivos: 6
1.3.1 Objetivo general 6
1.3.2 Objetivos específicos 6
1.4 Justificación 7
1.5 Hipótesis 7
1.6 Descripción de la empresa en colaboración 7
1.6.1 Misión 8
1.6.2 Visión 8
1.6.3 Política 8
1.6.4 Objetivos de la institución 8
1.7 Estado del arte 9
1.7.1 Machine Learning (ML) y optimización 9
1.7.2 Machine Learning (ML) e incertidumbre 10
1.7.3 Machine Learning (ML) y otras técnicas 11
1.7.4 Machine Learning (ML) en el área energético 12
1.7.5 Deep Learning (DL) y optimización 13
1.7.6 Deep Learning (DL) e incertidumbre 13
1.7.7 Deep Learning (DL) y otras técnicas 14
1.7.8 Deep Learning (DL) en el área energético 15
1.8 Conclusión 19
CAPÍTULO II. MARCO TEÓRICO 20
2.1 Introducción: 20
2.2 Sector energético de México. 20

2.2.1 Precio Marginal Local 21
2.2.2 Volatilidad 22
2.3 Inteligencia artificial 25
2.3.1 Redes neuronales artificiales 25
2.3.2 Fundamento biológico: 25
2.3.3 Topologías de las redes neuronales: 26
2.3.4 Ventajas y desventajas 28
2.3.5 Aprendizaje automático 28
2.3.6 Métodos de aprendizaje de los algoritmos de Machine Learning 30
2.3.7 Fases del Machine Learning 32
2.4 Métricas de los errores en pronostico 34
2.5 Árboles de Decisión para Regresión 35
2.6 Regresión Polinomial de Segundo Grado 35
2.7 SARIMA (Seasonal AutoRegressive Integrated Moving Average) 35
2.8 LSTM (Long Short-Term Memory) 36
2.9 LSTM Combinado con PROPHET 37
2.10 Conclusión 37
CAPÍTULO III. METODOLOGÍA 38
3.1 Introducción 38
3.2 Pasos de la Metodología 38
3.2.1 Definición del problema: 38
3.2.2 Recolección de las bases de datos: 39
3.2.3 Preprocesamiento 39
3.2.4 Construcción del modelo 39
3.2.5 Métricas de desempeño 40
3.2.6 Comparación del modelo 40
3.2.7 Afinación (Tuning) 41
3.2.8 Implementación 41
3.2.9 Exploración y mejoras 42
3.3 Definición del problema (panorama inicial) 42
3.4 Recolección de las bases datos 43
3.5 Preprocesamiento 45

3.5.1 Discriminación de variables. 47
3.5.2 Análisis estadístico. 51
3.5.3 Imputación de datos. 67
3.6 Construcción del modelo: 69
3.6.1 Árbol de decisión para regresión (Decision Tree Regressor): 69
3.6.2 Regresión polinomial de segundo grado 71
3.6.3 SARIMA: 75
3.6.4 LSTM: 80
3.6.5 LSTM Secuencial + Prophet: 81
3.7 Métricas de desempeño 84
3.7.1 Validación del modelo de Árbol de decisión para regresión 84
3.7.2 Métricas de los errores del modelo de árbol de decisión para regresión 85
3.7.3 Interpretación general del modelo de árbol de decisión para regresión 87
3.7.4 Validación del modelo de Regresión polinominal de segundo grado 87
3.7.5 Métricas de los errores del modelo de Regresión polinominal de segundo grado 88
3.7.6 Interpretación general del modelo regresión polinominal 90
3.7.7 Validación del modelo de SARIMA: 90
3.7.8 Métricas de los errores del modelo de SARIMA: 91
3.7.9 Interpretación general del modelo SARIMA: 92
3.7.10 Validación del modelo de LSTM: 93
3.7.11 Métricas de los errores del modelo LSTM: 94
3.7.12 Interpretación general del modelo LSTM: 95
3.7.13 Validación del modelo LSTM + Prophet: 95
3.7.14 Métricas de los errores del modelo LSTM + Prophet: 96
3.7.15 Interpretación general del modelo LSTM + Prophet: 97
3.8 Comparación de los algoritmos de pronósticos: 98
3.9 Afinación (Tuning): 99
3.9.1 Ingeniería de características: 99
3.9.2 Validación con Walk-forward. 100
3.9.3 Análisis de sensibilidad: 103
3.9.4 Búsqueda en cuadricula: 114
3.10 Conclusión 116

CAPÍTULO IV. RESULTADOS 117
4.1 Introducción 117
4.2 Implementación: 117
4.2.1 Los árboles de decisión para regresión. 118
4.2.2 Validación durante la implementación: 151
4.3 Exploración y mejoras: 163
4.3.1 Interfaz Gráfica de Usuario (GUI): 163
4.3.2 Entrega de resultados: 169
4.4 Tabla comparativa de las métricas de los modelos 176
4.5 Conclusion 178
CONCLUSIÓN: 179
RECOMENDACIONES 181
BIBLIOGRAFÍA 182

ÍNDICE DE FIGURAS

Figura 0.1 Precios marginales de los sistemas de conexión nacional 2
Figura 2.1. Neurona natural típica. 26
Figura 2.2. RNA de propagación hacia adelante 27
Figura 2.3 RNR (Red Neuronal Recurrente) donde la propagación forma un ciclo. 27
Figura 2.4. Deep Learning dentro de las técnicas de la I.A. 29
Figura 2.5. Comparativa de algoritmos de las redes neuronales tradicionales vs redes neuronales de Machine Learning 30
Figura 2.6. Tipos de algoritmos del Machine Learning 30
Figura 2.7. Entrenamiento de una neurona artificial. 32
Figura 2.8. Fase de prueba de un modelo de redes neuronales. 33
Figura 2.9. Métricas de desempeño para validar los modelos de Machine Learning. 34
Figura 3.1. Proceso de construcción de un modelo de Machine Learning. 38
Figura 3.2 Datos originales del Precio Marginal Local 43
Figura 3.3 Datos originales climatológicos 44
Figura 3.4 Bibliotecas de lectura, análisis y procesamiento de datos. 46
Figura 3.5. Preprocesamiento de datos del PML: 46
Figura 3.6. Variables disponibles del CENACE y CONAGUA 47
Figura 3.7. Data set de los modelos de regresión 49
Figura 3.8 Selección de los 720 registros más actuales de la base de datos. 51
Figura 3.9 Enfoque clásico y STL para el análisis y descomposición de las series temporales. . 52
Figura 3.10 Análisis clásico de la estacionalidad mediante descomposición de series de tiempo. 53
Figura 3.11 Análisis STL de la estacionalidad mediante descomposición de series de tiempo... 55
Figura 3.12. Comandos de creación de la serie y sus componentes temporales del PML. 57
Figura 3.13 Serie de tiempo del PML 57
Figura 3.14 Componentes de la serie de tiempo; Tendencia, estacionalidad y residuales del nodo 07LPZ-115 58
Figura 3.15 Comandos de obtención de estadísticos descriptivos del PML en un "*Boxplot*" 60
Figura 3.16 Gráfica de caja de estadísticos descriptivos de las variables del PML 61
Figura 3.17 Estadísticos descriptivos del nodo 07LPZ-115 61

Figura 3.18 Diagrama de correlación múltiple de las variables del nodo 07LPZ-115..... 63
Figura 3.19 Sesgo y curtosis del nodo 07LPZ-115 65
Figura 3.20 Imputación de datos faltantes en la base de datos del nodo 07LPZ-115 67
Figura 3.21 Dataset obtenida después del preprocesamiento de datos del nodo 07LPZ-115 68
Figura 3.22 Comandos para identificar la variables y subconjuntos en el data set. 69
Figura 3.23 Cardinalidad de los subconjuntos; entrenamiento, validación y prueba. 70
Figura 3.24 Estructura del modelo de árbol de decisión para regresión. 70
Figura 3.25 Comandos de identificación de las variables y subconjuntos del modelo de regresión polinominal..... 72
Figura 3.26 Estructura del modelo de Regresión Polinominal. 73
Figura 3.27 Comando para graficar los pronósticos del modelo de Regresión Polinominal 74
Figura 3.28 Comandos para la ejecución de Auto-ARIMA 75
Figura 3.29 Resultados del mejor modelo determinado por Auto-ARIMA..... 77
Figura 3.30 Estructura del modelo de SARIMA 79
Figura 3.31 Estructura del modelo LSTM..... 80
Figura 3.32 Estructura del modelo LSTM con Prophet 82
Figura 3.33 Comandos de ensamble de los modelos LSTM con Prophet..... 83
Figura 3.34 Validación del modelo de árbol de decisión para regresión con "*timeseriesplit*"..... 85
Figura 3.35 Métricas de desempeño del modelo de árboles de decisión para regresión. 86
Figura 3.36 Grafica de pronósticos del PML con el modelo de árbol de decisión para regresión. 87
Figura 3.37 Validación del modelo de Regresión polinominal mediante "*timeseriesplit*" 88
Figura 3.38 Métricas de desempeño del modelo de Regresión Polinominal 88
Figura 3.39 Gráfica de pronósticos del modelo de Regresión Polinominal. 89
Figura 3.40 Validación del modelo SARIMA con "*timeseriesplit*" 90
Figura 3.41 Métricas de desempeño del modelo SARIMA. 91
Figura 3.42 Valores reales vs Pronosticados por el modelo SARIMA. 91
Figura 3.43 Gráfica del pronóstico del PML con el modelo SARIMA..... 92
Figura 3.44 Validación del modelo LSTM con "*timeseriesplit*". 93
Figura 3.45 Métricas de desempeño del modelo LSTM 94
Figura 3.46 Grafica del pronóstico del PML con el modelo LSTM 95
Figura 3.47 Validación del modelo LSTM + Prophet con "timeseriesplit". 96
Figura 3.48 Métricas de desempeño del modelo LSTM + Prophet..... 96

Figura 3.49 Grafica de los pronósticos del modelo LSTM + Prophet..... 97
Figura 3.50 Comparación de las métricas de los modelos. 98
Figura 3.51 Comandos de aplicación de ingeniería de características al dataset "PML"..... 100
Figura 3.52 Comandos de validación al modelo con "*walk-forward*" 101
Figura 3.53 Validación del modelo LSTM + Prophet con "*Walk - forward*". 102
Figura 3.54 Arquitecturas de las 2 redes LSTM a someter el análisis de sensibilidad. 104
Figura 3.55 Descripción de las arquitecturas y escenarios del análisis de sensibilidad. 105
Figura 3.56 Gráfica de todos los escenarios del Análisis de Sensibilidad. 110
Figura 3.57 Gráfica del Análisis de sensibilidad de la red 1+1. 110
Figura 3.58 Gráfica del Análisis de Sensibilidad de la red 2+1. 111
Figura 3.59 Métricas obtenidas durante el análisis de sensibilidad. 112
Figura 3.60 Búsqueda cuadricula de hiperparámetros "alta demanda" "festivos" y "estacionales" en LSTM + Prophet. 114
Figura 4.1 Localización geográfica de los nodos para la implementación del modelo. 117
Figura 4.2 Lista de los 28 nodos para la implementación del modelo. 118
Figura 4.3 Diagrama de árbol del nodo 1 de la implementación del modelo. 119
Figura 4.4 Menú principal de la Interfaz Gráfica de Usuario "*PML Explorer".* 163
Figura 4.5 Ventana para visualizar las fechas de los datos del PML 164
Figura 4.6 Ventana de selección del rango de tiempo para visualizar en la GUI. 164
Figura 4.7 Ventana del visualizador de datos del PML de la Interfaz Gráfica de Usuario (GUI). 165
Figura 4.8 Ventana emergente del visualizador al usar la herramienta zoom de la Interfaz de Usuario..... 166
Figura 4.9 Interfaz de Usuario y botón de redireccionamiento al código de Análisis y pronósticos del PML. 167
Figura 4.10 Lectura de archivos para visualizar datos al inicio del código. 167
Figura 4.11 Widgets de dialogo para aceptar los archivos y proceder a compilarlos en el código. 168
Figura 4.12 Lectura de fechas de los archivos a fusionar para iniciar el código. 168
Figura 4.13 Arquitectura de la red LSTM (1 capa de 128 neuronas y una neurona de salida).. 169
Figura 4.14 Validación del modelo en el primer Split 170
Figura 4.15 Validación del modelo en el segundo Split. 170
Figura 4.16 Validación del modelo en el tercer Split 170

Figura 4.17 Validación del modelo en el conjunto de prueba. 171
Figura 4.18 Validación del modelo LSTM..... 172
Figura 4.19 Lista de los pronósticos obtenidos del modelo LSTM..... 172
Figura 4.21 Métricas del modelo LSTM + Prophet 173
Figura 4.20 Gráfica de los pronósticos a futuro del modelo LSTM..... 173
Figura 4.22 Búsqueda cuadricula para el modelo LSTM + Prophet. 174
Figura 4.23 Lista de predicciones del PML para el nodo 07LPZ-115 La Paz, Baja California Sur, BCS..... 174
Figura 4.24 Gráfica de predicciones del PML del nodo 07LPZ-115 La Paz, Baja California Sur, BCS..... 175
Figura 4.25 Gráfica interactiva en "*Plotly*" de las predicciones del PML del nodo 07LPZ-115. 175

ÍNDICE DE TABLAS

Tabla 1.1 Estado del arte basados en Machine Learning..... 17
Tabla 1.2 Estado del Arte basados en Deep Learning..... 18
Tabla 3.1. Tabla de los escenarios del 1 al 5 en el análisis de sensibilidad..... 106
Tabla 3.2. Tabla de los escenarios del 6 al 10 en el análisis de sensibilidad..... 107
Tabla 3.3. Tabla de los escenarios del 11 al 15 en el análisis de sensibilidad..... 108
Tabla 3.4. Tabla de los escenarios del 16 al 20 en el análisis de sensibilidad..... 109
Tabla 4.1. Análisis de variables del nodo 1 (02MMT-400), con árbol de decisión y Matriz de correlación..... 121
Tabla 4.2. Análisis de variables del nodo 2 (06MON-115), con árbol de decisión y Matriz de correlación..... 122
Tabla 4.3. Análisis de variables del nodo 3 (01TTH-230), con árbol de decisión y Matriz de correlación..... 123
Tabla 4.4. Análisis de variables del nodo 4 (08SLC-230), con árbol de decisión y Matriz de correlación..... 124
Tabla 4.5. Análisis de variables del nodo 5 (03AGM-400), con árbol de decisión y Matriz de correlación..... 125
Tabla 4.6. Análisis de variables del nodo 6 (08PTE-115), con árbol de decisión y Matriz de correlación..... 126
Tabla 4.7. Análisis de variables del nodo 7 (07SAF-115), con árbol de decisión y Matriz de correlación..... 127
Tabla 4.8. Análisis de variables del nodo 8 (07CPD-230), con árbol de decisión y Matriz de correlación..... 128
Tabla 4.9. Análisis de variables del nodo 9 (07TCT-69), con árbol de decisión y Matriz de correlación..... 129
Tabla 4.10. Análisis de variables del nodo 10 (07VPA-230), con árbol de decisión y Matriz de correlación..... 130
Tabla 4.11. Análisis de variables del nodo 11 (07LRO-115) con árbol de decisión y Matriz de correlación..... 131
Tabla 4.12. Análisis de variables del nodo 12 (07CAD-115), con árbol de decisión y Matriz de correlación..... 132

Tabla 4.13. Análisis de variables del nodo 13 (07GAO-115), con árbol de decisión y Matriz de correlación. 133
Tabla 4.14. Análisis de variables del nodo 14 (07LPZ-115), con árbol de decisión y Matriz de correlación. 134
Tabla 4.15. Análisis de variables del nodo 15 (07SNT-115), con árbol de decisión y Matriz de correlación. 135
Tabla 4.16. Análisis de variables del nodo 16 (08TUM-115), con árbol de decisión y Matriz de correlación. 136
Tabla 4.17. Análisis de variables del nodo 17 (06CBD-400), con árbol de decisión y Matriz de correlación. 137
Tabla 4.18. Análisis de variables del nodo 18 (07rum-69), con árbol de decisión y Matriz de correlación. 138
Tabla 4.19. Análisis de variables del nodo 19 (03LVT-69), con árbol de decisión y Matriz de correlación. 139
Tabla 4.20. Análisis de variables del nodo 20 (05MVI-400), con árbol de decisión y Matriz de correlación. 140
Tabla 4.21. Análisis de variables del nodo 21 (05PAR-115), con árbol de decisión y Matriz de correlación. 141
Tabla 4.22. Análisis de variables del nodo 22 (04GSV-115), con árbol de decisión y Matriz de correlación. 142
Tabla 4.23. Análisis de variables del nodo 23 (01AFM-230), con árbol de decisión y Matriz de correlación. 143
Tabla 4.24. Análisis de variables del nodo 24 (01ANL-85), con árbol de decisión y Matriz de correlación. 144
Tabla 4.25. Análisis de variables del nodo 25 (04PLD-230), con árbol de decisión y Matriz de correlación. 145
Tabla 4.26. Análisis de variables del nodo 26 (06CEN-138), con árbol de decisión y Matriz de correlación. 146
Tabla 4.27. Análisis de variables del nodo 27 (03AMX-69), con árbol de decisión y Matriz de correlación. 147
Tabla 4.28. Análisis de variables del nodo 28 (02CBD-400) con árbol de decisión y Matriz de correlación. 148
Tabla 4.29. Resultado de los nodos Interconectados y ordenados de acuerdo con el nivel de

correlación. 150
Tabla 4.30. Validación *walk-forward* a los modelos entrenados para los nodos 1 al 4..... 151
Tabla 4.31. Validación *walk-forward* a los modelos entrenados para los nodos 5 al 9..... 153
Tabla 4.32. Validación *walk-forward* a los modelos entrenados para los nodos 10 al 14..... 155
Tabla 4.33. Validación *walk-forward* a los modelos entrenados para los nodos 15 al 19..... 157
Tabla 4.34. Validación *walk-forward* a los modelos entrenados para los nodos 20 al 24..... 159
Tabla 4.35. Validación *walk-forward* a los modelos entrenados para los nodos 25 al 28..... 161
Tabla 4.36. Métricas de errores registrados en la implementación a 28 nodos del país..... 176

ABREVIATURAS

PML	Precio Marginal Local
CENACE	Centro Nacional de Control de Energía
ML	Machine Learning
RNA	Red Neuronal Artificial
IA	Inteligencia Artificial
INEEL	Instituto Nacional de Electricidad y Energías Limpias
CFE	Comisión Federal de Electricidad
MEM	Mercado Eléctrico Mayorista
K-NN	K-Nearest Neighbors
MAPE	(Mean Absolute Percentage Error): Error Porcentual Absoluto Medio
MAE	(Mean Absolute Error): Error Absoluto Medio
MSE	(Mean Squared Error): Error Cuadrático Medio
RMSE	(Root Mean Squared Error): Raíz del Error Cuadrático Medio
RNN	(Recurrent Neural Network): Red Neuronal Recurrente
LSTM	(Long Short-Term Memory): Memoria a Largo y Corto Plazo
ADR	Árbol de Decisión para Regresión
SARIMA	(Seasonal Auto Regressive Integrated Moving Average): Media móvil integrado autorregresivo estacional
RPSG	Regresión Polinominal de Segundo grado
STL	(Seasonal and Trend decomposition using Loess): Descomposición Estacional y de Tendencia usando Loess

INTRODUCCIÓN

La ley de la Industria Eléctrica (LIE) (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006) establece que el Centro Nacional de Control de Energía (CENACE) tendrá a su cargo el control operativo del SEN (Servicio Eléctrico Nacional) y la operación del Mercado Eléctrico Mayorista (MEM). La implementación del MEM incluye un Mercado de Energía de Corto Plazo (MECP) que comprende al Mercado del Día en Adelanto (MDA) y al Mercado de Tiempo Real (MTR), el CENACE realizará el despacho económico de las Unidades de Central Eléctrica para cada uno de los mercados. El resultado de dicho despacho económico son los Precios Marginales de las Reservas en cada Zona de Reserva y los Precios Marginales Locales (PML´s) de la energía en cada Nodo del SEN, integrados por un componte de energía, congestión y pérdidas (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006).

Desde el arranque del MECP para cada Sistema Interconectado, diariamente se reciben ofertas de Compra de Energía y Ofertas de Venta de Energía y Servicios Conexos (SC) antes de las 10:00 horas, con las ofertas y otros insumos calculados por el CENACE, se ejecuta el modelo de optimización de Asignación del Unidades del Mercado del Día en Adelanto (AU-MDA), el cual asigna y despacha Unidades de Central Eléctrica (UCE) en el MDA; con esta herramienta se decide el programa horario de arranques, paros, cambios de configuración, potencias de generación y asignación de los Servicios Conexos, determinando los PML´s de la energía y los precios de los SC por Zonas de Reserva. (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006)

El Precio Marginal Local (PML) refleja el valor de la energía en un tiempo y lugar determinado. Este se calcula por medio de un modelo de optimización que minimiza los costos de generación mientras se satisface la demanda y se respetan las restricciones físicas de la red eléctrica. (Alonso & Gabriel, 2017) Por lo tanto, la importancia de un precio justo mejorará los indicadores de disponibilidad, confiabilidad y eficiencia que se maneja en todos los procesos de potencia que se manejan en CFE, que, a su vez, afectarán de manera positiva a los clientes consumidores de electricidad.

Estadísticamente podemos ver un patrón de volatilidad de los PML (véase Figura 0.1).

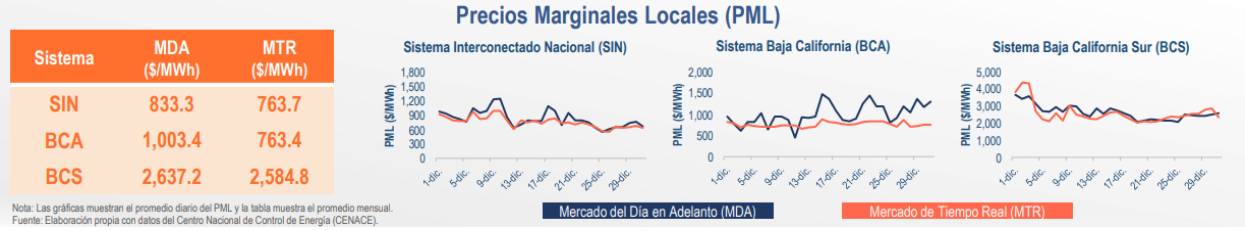


Figura 0.1 Precios marginales de los sistemas de conexión nacional

Durante diciembre de 2021, el Precio Marginal Local (PML) se ubicó en $833.3/MWh en promedio y registró una disminución porcentual de 15.0% e incrementó 49.0% en comparación mensual e interanual, respectivamente. La disminución del PML con respecto al mes previo se debió, preliminarmente, a una disminución de la demanda. (Nexus.Energia.mx, 2021)

La gestión y análisis de bases de datos que albergan grandes volúmenes de información requieren métodos más eficaces que los enfoques tradicionales. En el contexto del Mercado Eléctrico de Corto Plazo (MECP), se observa una constante fluctuación en los precios de la energía. Esta dinámica genera una considerable cantidad de datos cada hora, este fenómeno resulta en la acumulación de extensas bases de datos, cuyo manejo eficiente es crucial para comprender y predecir las tendencias del mercado.

Por esa razón, el presente documento presenta un modelo inteligente de pronóstico con redes neuronales para el Precio Marginal Local (PML) considerando el clima como factor a integrar y que influye en el precio de la energía de un nodo en específico. En la realización del modelo se llevó a cabo con el Instituto Nacional de Electricidad y Energías Limpias (INEEL), el cual es un centro de investigación dedicado a dar solución a las necesidades del sector eléctrico y energético, teniendo como usuario principal la Comisión Federal de Electricidad (CFE).

Este trabajo comienza con una introducción a la problemática de la volatilidad en los precios marginales, destacando su relevancia para el Mercado Eléctrico Mayorista (MEM) y la Comisión Federal de Electricidad (CFE) en garantizar la oferta de energía. Se enfatiza la importancia de predecir estos precios de manera oportuna para el Centro Nacional de Control de Energía (CENACE), asegurando así un despacho eficiente y a tiempo.

Después, se detallan los objetivos y la hipótesis de la investigación. Seguidamente, se realiza una revisión de la literatura reciente para identificar estudios previos relacionados con la predicción de precios de energía y enfoques similares.

Dentro del marco teórico, se exponen los fundamentos técnicos y teóricos que sustentan este estudio. Se exploran aspectos clave en el funcionamiento de los modelos de pronóstico, así como las técnicas de inteligencia artificial y los algoritmos de aprendizaje automático empleados.

En el capítulo dedicado a la metodología, se describen las etapas seguidas para desarrollar el modelo inteligente. Este modelo se optimiza para mejorar su rendimiento y se valida mediante métricas que cuantifican los errores en los pronósticos.

En el capítulo de resultados, se obtienen los resultados de los 5 modelos de pronósticos donde el mejor de todos es el modelo LSTM secuencial, logrando métricas de los errores para comprender la eficiencia del modelo de machine Learning al implementar el modelo a 28 nodos de interconexión.

El proyecto culmina con el desarrollo de una Interfaz de Usuario Gráfica (GUI) llamada “PML Explorer”. Esta interfaz interactiva permite a los usuarios visualizar de manera intuitiva los datos del Precio Marginal Local (PML). Además, incorpora una funcionalidad clave: la capacidad de ejecutar un código desarrollado en Python que integra cinco algoritmos avanzados de aprendizaje automático. Esta herramienta está diseñada para predecir el PML hasta 48 horas en el futuro, mejorando significativamente la precisión de los pronósticos y facilitando la toma de decisiones basada en datos concretos y análisis predictivo.

CAPÍTULO I. GENERALIDADES

1.1 Introducción

Este capítulo aborda la problemática asociada al precio marginal de los nodos de energía del país, destacando la importancia del pronóstico oportuno de los precios en el MEM. También, se exponen los objetivos, la hipótesis, y se revisa la literatura reciente en investigaciones previas que han abordado el pronóstico de los precios de la energía.

1.2 Planteamiento del problema

El Mercado Eléctrico Mayorista (MEM) se creó como una estipulación de la Reforma Eléctrica en México (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006), Con el MEM se asegura contar con todos los insumos para la operación óptima del Sistema Eléctrico Nacional (SEN) el cual se encuentra dividido por tres sistemas:

- Sistema Interconectado Nacional (SIN)
- Sistema Baja California (BCA)
- Sistema Baja California Sur (BCS)

Todo este sistema se encuentra a cargo del Centro Nacional de Control de Energía, cuya misión es administrar los recursos energéticos en busca de un crecimiento económico. El Precio Marginal Local, se define como el precio de la energía en un nodo determinado en el SEN para un periodo definido, calculado por el CENACE para el MECP (Mercado Eléctrico de Corto Plazo). El PML es el resultado de tres componentes:

1. **Componente de Congestión Marginal**, que representa el costo marginal de congestionamiento en cada Nodo;
2. **Componente de Energía Marginal**, que representa el costo marginal de energía en el nodo de referencia del Sistema Interconectado correspondiente y;

3. **Componente de Pérdidas Marginales**, que representa el costo marginal de pérdidas en cada Nodo.

El PML está integrado por la suma de tres componentes: PML = Componente Energía + Componente Congestión + Componente Pérdida, Cada uno de estos componentes puede asumir valores positivos, negativos o incluso ser nulos. En el desarrollo de un modelo de machine Learning para pronosticar el PML, resulta proactivo la integración de variables climáticas. Estas variables son particularmente relevantes en el contexto de la demanda y generación de energía en México, ya que proporcionan una perspectiva adicional y crítica para mejorar la precisión y relevancia del modelo de pronóstico.

El desafío principal en el Mercado Eléctrico Mayorista (MEM) radica en la constante generación de grandes volúmenes de datos. Esta acumulación de información representa un reto significativo para los responsables de la toma de decisiones operativas y estratégicas (como el CENACE), quienes deben gestionar y analizar eficientemente estos datos. De aquí surge la necesidad de aplicar métodos analíticos avanzados para el monitoreo, diagnóstico, predicción, planeación y acción. El objetivo de esta investigación es mejorar el pronóstico del Precio Marginal Local (PML), detectando tendencias mediante la integración de factores previamente no considerados, como las condiciones climáticas, que pueden tener un impacto significativo en el PML. Este enfoque no solo proporciona una visión más completa del PML, sino que también facilita una toma de decisiones más informada y estratégica.

Con base en lo mencionado anteriormente acerca del MEM y el PML, se obtiene un modelo de predicción eficiente que sirva como una herramienta en la estimación del PML real. Este modelo, apoyado en algoritmos de Machine Learning implementados en Python, será complementado con una Interfaz Gráfica de Usuario (GUI), facilitando así su uso en un servicio web interactivo (ambiente Jupyter). Esto permitirá a los usuarios obtener resultados rápidos y confiables de forma interactiva, mejorando la precisión y utilidad en la estimación del PML.

1.3 Objetivos:

A continuación, se presenta el objetivo general seguido de los objetivos específicos a alcanzar en el proyecto.

1.3.1 Objetivo general

Desarrollar un modelo inteligente de predicción para estimar el Precio Marginal Local (PML) en cualquier nodo de interconexión de la Red Eléctrica Nacional con un horizonte de predicción de 48 horas con el objetivo de reducir significativamente la incertidumbre asociada con la oferta y demanda de energía.

1.3.2 Objetivos específicos

- Gestionar y preprocesar una base de datos de los PML´s considerando el sistema con sus variables correspondientes de energía y meteorológicos.
- Diseñar y desarrollar un modelo inteligente de predicción basados en algoritmos de Machine Learning.
- Validar el desempeño del modelo inteligente mediante métricas de precisión, la matriz de correlación y árboles de decisión.
- Demostrar la capacidad de diagnóstico del modelo inteligente en la implementación a 28 nodos, mediante métricas de errores de pronóstico.
- Crear un modelo de pronóstico en Python mediante el entorno de *jupyter* notebook que sea interactivo con el usuario al contar con una interfaz de visualización de datos del PML.

1.4 Justificación

En los mercados a corto plazo de energía, los precios pueden ser muy volátiles debido a las condiciones del sistema, por lo que se propone la aplicación del Machine Learning al procesamiento de los datos, mediante la técnica de aprendizaje automático para aportar un pronóstico más eficiente al encontrar patrones de comportamiento y determinar la estructura del sistema al considerar factores que no han sido analizados. Estos factores meteorológicos son: la temperatura del aire, las precipitaciones, la radiación solar y la rapidez del viento, que se consideran que pueden ser las variables que justifiquen la volatilidad del PML.

El modelo desarrollado está orientado a optimizar la determinación de los Precios Marginales Locales (PML); este trabajo representa un logro significativo en el desarrollo de una metodología para predecir los PML mediante modelos de Machine Learning, lo cual puede ser de gran utilidad como referencia, porque en un mercado donde el precio de la electricidad es un indicador crítico, proporcionar un pronóstico preciso y confiable de los PML se vuelve fundamental para todos los participantes del mercado energético mexicano.

1.5 Hipótesis

El desarrollo de un modelo inteligente de predicción basado en redes neuronales y aprendizaje profundo, específicamente utilizando técnicas como LSTM (Long Short-Term Memory) y Prophet, permitirá realizar pronósticos más precisos y confiables del Precio Marginal Local (PML) para un nodo específico a 48 horas a futuro.

1.6 Descripción de la empresa en colaboración

Este proyecto se desarrolla en colaboración con el Instituto Nacional de Electricidad y Energías Limpias (INEEL), un prestigioso centro de investigación mexicano con sede en Cuernavaca. Fundado en 1975 originalmente como Instituto de Investigaciones Eléctricas, el INEEL se ha establecido como un líder en la promoción de soluciones innovadoras para el ahorro energético y económico, así como en el impulso del desarrollo sustentable dentro de la industria energética. Con más de cuatro décadas de experiencia, INEEL no solo asesora a la Comisión Federal de Electricidad (CFE) y otras entidades clave del sector, sino que también participa

activamente en proyectos pioneros de innovación y desarrollo tecnológico. Cuenta con una extensa experiencia en el desarrollo de soluciones tecnológicas avanzadas para entidades como CFE, PEMEX, CENACE, entre otros. Es reconocida con profundo conocimiento en los procesos de negocio y en la transformación digital del sector energético. De acuerdo con el (Instituto Nacional de Electricidad y Energías Limpias, 2023) tanto los objetivos como la misión, visión y política de la empresa son:

1.6.1 Misión

Tiene como misión realizar investigación aplicada y desarrollo tecnológico en apoyo a la soberanía energética y al desarrollo sostenible en materia de electricidad y energías limpias. E impulsar el desarrollo sustentable de la industria energética mediante la innovación.

1.6.2 Visión

Tiene como visión ser reconocido como líder en investigación aplicada y desarrollo tecnológico en materia de electricidad y energías limpias. Además, que se le acreditan aportaciones únicas a la competitividad de la industria por la formación de ecosistemas de innovación y creación de valor (gobierno, industria y academia) que propician e impulsan la sustentabilidad de la industria energética.

1.6.3 Política

Es compromiso del Instituto Nacional de Electricidad y Energías Limpias satisfacer las necesidades de sus clientes y superar sus expectativas mediante la innovación, la eficacia y la mejora continua de sus procesos, dentro del marco normativo, legal y regulatorio aplicable en materia de gestión de la calidad, desarrollo sustentable, igualdad laboral, gestión del medio ambiente, seguridad y salud en el trabajo.

1.6.4 Objetivos de la institución

Los objetivos que pretende lograr dicho instituto son los siguientes:

1. Realizar investigación aplicada, desarrollo tecnológico e innovación para contribuir al fortalecimiento operativo y tecnológico de las Empresas Productivas del Estado (EPE),

los organismos públicos, las empresas mixtas y empresas de capital nacional que participan en su cadena de proveeduría.

2. Fortalecer la vinculación del Instituto con instituciones, dependencias y entidades relacionadas con el sector energético, en apoyo al desarrollo de tecnologías críticas.
3. Contribuir a la formación de especialistas para atender las necesidades de la transición energética y el fortalecimiento del sector.
4. Contar con la viabilidad financiera, así como el desarrollo sostenido de la Institución para apoyar a la Transición Energética Soberana (TES) en materia científica y tecnológica.

1.7 Estado del arte

En la revisión de la literatura acerca del uso de técnicas de inteligencia artificial (IA) se ha expandido en los últimos años, debido a los grandes y satisfactorios resultados que han logrado al aplicarlas; por ejemplo, en la modelación de demanda incierta, sistemas de control (temperatura, velocidad, tráfico, etc.), tecnología informática, procesamiento y reconocimiento de imágenes, pronósticos con series de tiempo y últimamente también en la optimización de sistemas.

La predicción de series temporales se puede realizar utilizando redes neuronales recurrentes (RNN) como la capacidad de tener una "memoria" que las hace buenas para tareas de predicción de secuencias largas. La RNN puede recordar información contextual a través de activaciones de capas ocultas que se pasan de un paso en el tiempo a otro. Diferentes variantes de RNN han sido útiles cuando la dependencia temporal de los datos es importante. (SatvikKhuntia, AtharHanif, QadeerAhmed, JohnLahti, & MaartenMeijer, 2022). A continuación, se presenta la revisión de la literatura de la aplicación de distintas técnicas de la Inteligencia Artificial como son el Machine Learning y Deep Learning en el pronóstico de series de tiempo:

1.7.1 Machine Learning (ML) y optimización

El algoritmo propuesto (véase el artículo 6 de la Tabla 1.1) utiliza dos conjuntos de características de entrada independientes: indicadores técnicos internos y precios de mercado externos. Este esquema de pronóstico bilateral pasa por un proceso de selección de características de dos etapas compuesto por la expansión del conjunto de características, regresiones híbridas de

algoritmos genéticos y aprendizaje automático para seleccionar características importantes y filtrado de puntuación de importancia para seleccionar características óptimas. Finalmente, se selecciona el mejor subconjunto de características para el pronóstico y la interpretación. Para mejorar la interpretabilidad local en su estudio, recurre al suavizado de Savitzky-Golay como parte del ajuste de curva óptimo por partes para examinar cada agrupación potencial de características externas. En comparación con otras técnicas de interpretabilidad de la importancia de las características que solo se basan en un solo punto de datos o en un período de datos completo, la técnica de interpretabilidad propuesta supera sus limitaciones y refleja las características principales de los datos de series temporales. (Kyung Keun Yun, Sang Won Yoon, & Won, 2022)

1.7.2 Machine Learning (ML) e incertidumbre

Por otra parte la aplicación de técnicas de machine Learning han hecho frente a la incertidumbre por ejemplo el documento (véase el artículo 1 de la Tabla 1.1) se centra en el desarrollo de un nuevo método de predicción de olas para la generación de energía pero con un enfoque probabilístico; En particular, el enfoque de aprendizaje automático bayesiano (BML) para la predicción de ondas resueltas en fase, que puede aprovechar la capacidad de ML para abordar problemas no lineales mientras toma diferentes tipos de incertidumbres en cuenta a través del marco bayesiano. Como la inferencia de la verdadera distribución posterior en el modelo de red neuronal bayesiana (BNN) es generalmente inviable debido a su extrema complejidad computacional, se han utilizado varios métodos de aproximación, como deserción Monte Carlo (MC), el desarrollo de hardware ML moderno (por ejemplo, GPU). El método propuesto está diseñado para tener en cuenta tanto la incertidumbre aleatoria (es decir, la incertidumbre de la información de onda resuelta en fase) como la incertidumbre epistémica (la incertidumbre debida a la capacidad del modelo). Según el conocimiento de los autores, esta es la primera vez que se logra la predicción de ondas no lineales en tiempo real con incertidumbre cuantificada que incluye incertidumbres epistémicas y aleatorias. (Jincheng Zhang, Xiaowei Zhao, Siya Jin, & Greaves, 2022)

Los avances tecnológicos facilitan el uso de métodos de aprendizaje automático en los campos médicos (véase artículo 2 de la Tabla 1.1). En la combinación de herramientas de

aprendizaje automático y datos de expresión génica para detectar pacientes con cáncer. Este artículo analizó el impacto que tienen diferentes técnicas de reducción de dimensionalidad en los modelos de aprendizaje automático utilizados para la predicción del cáncer. Se utilizaron técnicas de reducción de la dimensionalidad como el análisis de componentes principales (PCA), PCA con núcleo y codificador automático para reducir la dimensionalidad de los datos de secuenciación de ARN. Se entrenaron y probaron dos clasificadores de aprendizaje automático, a saber, la red neuronal y la máquina de vectores de soporte, utilizando los datos originales, dimensionalmente reducidos y relevantes para el cáncer. Los resultados mostraron que la reducción de la dimensionalidad afecta positivamente el desempeño de los clasificadores. Además, el codificador automático funcionó mejor que PCA y PCA con kernel. Estos hallazgos indican el potencial de la reducción de la dimensionalidad para mejorar los resultados analíticos de los modelos de clasificación de aprendizaje automático en datos de alta dimensión. (Md Faisal kabir, Tianjie Chen, & Ludwig, 2022)

1.7.3 Machine Learning (ML) y otras técnicas

Considerando las múltiples técnicas del Machine Learning encontramos la técnica de algoritmos de agrupamiento; la literatura (véase artículo 3 de la Tabla 1.1) ofrece la propuesta del modelo híbrido basado en la descomposición de modo variacional (VMD), análisis de agrupamiento, red LSTM, aprendizaje de conjuntos de apilamiento y complementación de errores para el pronóstico de la velocidad del viento en el cual todos los componentes se realizan en la plataforma Flink para garantizar la eficiencia del pronóstico. Más específicamente, el módulo VMD se emplea para desintegrar la serie de velocidad del viento en una tendencia primaria y varias subseries fluctuantes, después se llevan a cabo agrupaciones de k-means y redes LSTM para deducir las características latentes de la tendencia primaria y se aplica el aprendizaje de conjuntos de apilamiento que consta de dos etapas para inferir las abstracciones fluctuantes de las otras subseries; Además, se incorpora el complemento de error para evaluar la secuencia de error creada por los resultados preliminares. Finalmente, los resultados experimentales han demostrado que el modelo propuesto supera a los modelos contrastivos en precisión y eficiencia de pronóstico. (Zexian Sun, Mingyu Zhao, & Zhao, 2022)

Otra técnica se aplicó en el contexto de la calificación crediticia, los métodos de conjunto

basados en árboles de decisión, como el método de bosque aleatorio (véase artículo 5 de la Tabla 1.1). Se propone un método de calificación crediticia interpretable y de alto rendimiento llamado regresión de árbol logístico penalizado (PLTR), que utiliza información de árboles de decisión para mejorar el rendimiento de la regresión logística. Formalmente, las reglas extraídas de varios árboles de decisión de poca profundidad construidos con variables predictivas originales se utilizan como predictores en un modelo de regresión logística penalizado. PLTR nos permite capturar los efectos no lineales que pueden surgir en los datos de calificación crediticia mientras se preserva la interpretabilidad intrínseca del modelo de regresión logística. Las simulaciones de Monte Carlo y las aplicaciones empíricas que utilizan cuatro conjuntos de datos reales de incumplimiento crediticio muestran que PLTR predice el riesgo crediticio con mucha más precisión que la regresión logística y se compara competitivamente con el método de bosque aleatorio. (Elena Dumitrescu, Sullivan Hué, Christophe Hurlin, & Tokpavi, 2022)

1.7.4 Machine Learning (ML) en el área energético

El uso de redes neuronales es ampliamente practicado en análisis y optimización, por ejemplo; con el fin de reducir el impacto de la producción de energía eólica y la incertidumbre del precio de la electricidad en los ingresos de la energía eólica que participa en el mercado eléctrico, el documento (véase artículo 4 de la Tabla 1.1) propone como primera etapa modelar la producción eólica y del precio de la electricidad, buscando la máxima renta de los parques eólicos en el mercado diario, se obtiene el poder de oferta óptimo de cada parque eólico en el mercado diario mediante el uso de “Quantum algoritmo genético”; mediante la computación cuántica hace que se procesen sus algoritmos a una mayor velocidad. En la segunda etapa, en función del volumen de la oferta ganadora del mercado diario y la producción real, mediante la coordinación de cada parque eólico para utilizar los servicios de carga y descarga de la central eléctrica de almacenamiento de energía compartida, el ingreso neto de múltiples parques eólicos en el día - por delante y el mercado en tiempo real se maximiza. (Xiyun Yang, Liwei Fan, Xiangjun Li, & Meng, 2022).

Principalmente podemos ver el alcance de la IA en el PML de acuerdo con el enfoque del articulo 7 (véase Tabla 1.1). El estudio consideró una base de datos compuesta por 11 variables independientes (integradas por factores estacionales, ambientales, operativos y económicos) para

pronosticar el PML. Los datos fueron sometidos a un preprocesamiento consistente en filtros de limpieza, suavizado y normalización. Posteriormente, se entrenó un modelo basado en redes neuronales artificiales (RNA) y otro de lógica difusa. Al final, se realizó un análisis de sensibilidad global (GSA). El día del año fue la variable con mayor impacto en los componentes de pérdida y congestión. Lo anterior está asociado al déficit de capacidad de transmisión eléctrica. Por otro lado, los precios de los combustibles impactan significativamente en el componente energético, que está relacionado con el perfil de mezcla de generación de la región peninsular. (A.Livas-García, Tzuc, May, RasikhTariq, & Torres, 2022)

1.7.5 Deep Learning (DL) y optimización

Otro trabajo propuesto (véase artículo 7 de la Tabla 1.2) establece una red neuronal "gris" de predicción aunado con una propuesta de algoritmos genéticos donde predicen y miden el rendimiento del modelo para aplicarlo iterativamente y optimizando su capacidad de predicción del proceso, el modelo de predicción "gris" aquí propuesto requiere menos información comparado con los métodos tradicionales siendo muy preciso mediante las mediciones de errores entre capas detentando las neuronas con valores erráticos para recalcularlos o reconfigurando sus pesos, volviendo a pasar el proceso de entrenamiento y medición, arrojando pronósticos satisfactorios. (WEI Wei & Chuan, 2022)

Otro ejemplo de optimización se lee en el artículo 1 (véase Tabla 1.2) qué se basa en la densidad del núcleo gaussiano y la simulación Monte Carlo. El objetivo principal del enfoque propuesto es predecir, para cada hora del día, el comportamiento de volatilidad del par de divisas seleccionado. En particular la red LSTM tiene la función de largo plazo debido a su estructura cíclica y adecuada para las series temporales. Incluye un preprocesamiento basado en la "transformada de Wavelet empírica" (EWT) y el post procesamiento basado en el "modelo de Machine Learning extremo robusto" (ORELM) y optimizado con el algoritmo de "enjambre de partículas" (PSO). (HuiLiu∗ & ZhihaoLong, 2020)

1.7.6 Deep Learning (DL) e incertidumbre

La investigación del articulo 2 (véase Tabla 1.2) nos da un modelo de red neuronal

profunda que utiliza los registros de transacciones insensibilizados y la información pública del mercado para predecir la tendencia del precio de las acciones. Teniendo en cuenta la correlación entre las acciones, que aplicando una capa neuronal (BiLSTM) aprovecha el potencial de las capas LSTM de retro propagación pero bidireccionalmente para el vector de salida será evaluado por los vectores de información del mercado y precios; se utiliza un algoritmo negativo parecido a la función de perdida que comúnmente se utiliza en las métricas de los pesos pero este algoritmo llamado "Adam" minimizara los valores de perdida en todas las instancias durante el entrenamiento. (Jiawei Long, Zhaopeng Chen, Weibing He, Taiyu Wu, & Ren, 2020)

Otro trabajo (véase artículo 4 de la Tabla 1.2) utiliza una "red integrada de fluctuación de datos" (DFN) donde aplicando varios algoritmos de "Inteligencia Artificial" (IA) crean el algoritmo denominado modelo DFN-AI. En el modelo DFN-AI propuesto, se tiene un análisis complejo de series temporales y lo hace con tres estructuras de las redes; "redes neuronales de retro propagación" (BPNN) que aquí funciona como una técnica de análisis de regresión múltiple y sabiendo sus limitaciones ahora es analizada con otra "red neuronal de función radial" (RBFNN) que crea un subespacio llamado "hiper esfera" que mejora el algoritmo de cálculo de los pesos entre neuronas vecinas propiciando la última capa neuronal llamada "extreme Learning Machine" para aprender la red neuronal final de las dos anteriores. (Wang, Zhao, Duc, Wang, & Chen, 2018)

1.7.7 Deep Learning (DL) y otras técnicas

La metodología propuesta de selección de características de conjunto (EFS) (véase artículo 3 de la Tabla 1.2) que comprende los algoritmos "Boruta y Regularized Random Forest" (RRF) se utiliza para seleccionar las características explicativas mediante arboles de decision que clasifican los datos de entrada al ser un aprendizaje automático supervisado; Se utilizan dos técnicas avanzadas de inteligencia artificial, "Regularized Greedy Forest" (RGF) que es una técnica de árbol de decision que simula en un ambiente Python y "Deep Neural Network" (DNN) que usa la función "ReLu" (rectified linear unit) simplificando mucho los pesos de las funciones de activación entre neuronas, y junto con "Kernel Principal Component Analysis" (KPCA) y "Autoencoder" (AE) para la previsión; todas estas funciones establecen pesos a las

neuronas entre las capas ocultas que solo pueden ser cero o un numero “z” lo que simplifica mucho el tiempo de ejecución de las funciones y cálculos matriciales que al complementarse con las ultimas capas ajustan los pesos. (Indranil Ghosh, Chaudhuri, Alfaro-Cortés, & Gámez, 2022)

El documento (véase artículo 5 de la Tabla 1.2) concluye con una discusión de las ventajas y las limitaciones potenciales de las técnicas de IA revisadas en “sistemas multiagentes” (útil para teorías de juegos y las negociaciones automatizadas), la categoría de “Nature Inspired Intelligence” que sus algoritmos basan sus iteraciones en encontrar máximos y mínimos y ejemplos de estas técnicas son los “Algoritmos Genéticos”. Las dos últimas técnicas son Machine Learning que para efectos de análisis de datos en grandes volúmenes son ideales y que pueden ser de aprendizaje supervisado o no supervisado obteniendo capas con información más abstractas que han aprendido de sus antecesoras en su estructura. Cabe mencionar que recomienda el uso de la técnica de Machine Learning en las predicciones de mercados energéticos y tratamientos de flexibilidad. (Antonopoulos, Robu, Benoit Couraud, Desen Kirli, & Norbu, 2020)

1.7.8 Deep Learning (DL) en el área energético

En el artículo 9 (véase Tabla 1.2) se describe y caracteriza la evolución temporal de la serie de tiempo de los PML´s horarios – diarios resultantes del MECP en México, analizando simultáneamente su volatilidad a lo largo de los periodos para cada mercado de energía (MDA y MTR) y para cada sistema interconectado (SIN, BCA y BCS) analizando la serie de tiempo no es estacionaria debido a patrones asociados a la frecuencia semanal donde el análisis se hizo enfocándose a datos históricos analizados estadísticamente (Jeovani E. Santiago López & García, 2018).

El estudio del articulo 10 (véase Tabla 1.2) analiza la relación entre el precio marginal del mercado de electricidad y la demanda mediante el uso de conglomerados analizados matemáticamente, donde se definió que el precio de la electricidad depende fundamentalmente de la demanda y la potencia disponible que es cambiante a lo largo del tiempo (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006).

El artículo 6 de la Tabla 1.2, usa la historia y otros factores estimados en el futuro para "ajustar" y "extrapolar" los precios y las cantidades. Se desarrolla un método de red neuronal para pronosticar los precios de liquidación del mercado (MCP) para los mercados energéticos diarios. La estructura de la red neuronal es una “red de propagación inversa” (BP) de tres capas donde considera el precio de los combustibles y otros factores en una clásica red neuronal de retro propagación ya desde el año 2011. (Deepak Singhal & Swarup, 2011)

Pero un poco más actualizado, encontramos en la investigación en el artículo 8 (véase Tabla 1.2) que es un estudio sistemático y empírico de las funciones de pérdida que pueden optimizar la previsión de los precios al contado de la electricidad para el día siguiente. Primero se asignan pesos a la red neuronal que utiliza una regresión de los datos con la técnica de “K- vecino más cercano” (KNN) y una estructura reforzada con redes del tipo LSTM que tienen aprendizaje reforzado por retro propagación, obteniendo métricas estadísticas de errores estándar como MSE y MAE. (Ahmad Amine Loutfi, Mengtao Sun, Ijlal Loutfi, & Solibakke, 2022).

Los artículos presentados en la revisión de la literatura se concentran a continuación en la Tabla 1.1 y Tabla 1.2:

Tabla 1.1 Estado del arte basados en Machine Learning.

N	Referencia	Artículo	Enfoque	Indicadores
1	(Jincheng Zhang, Xiaowei Zhao, Siya Jin, & Greaves, 2022)	Predicción de olas oceánicas en tiempo real resuelta por fase con incertidumbre cuantificada basada en aprendizaje automático bayesiano variacional.	-Algoritmos Bayesianos -Regresión hibrida -Algoritmos Genéticos. -redes neuronales -Monte Carlo	DFT, DSWP, RMSE, MRMSE
2	(Md Faisal kabir, Tianjie Chen, & Ludwig, 2022)	Un análisis de rendimiento de algoritmos de reducción de dimensionalidad en modelos de aprendizaje automático para la predicción del cáncer	-Machine Learning -Reducción de dimensionalidad -RNA – seq -SMOTE	Hi-seq RNA PCA, RBF, SVM, ROC, medida F.
3	(Zexian Sun, Mingyu Zhao, & Zhao, 2022)	Modelo híbrido basado en descomposición de VMD, análisis de agrupamiento, red de memoria larga y corta, aprendizaje de conjuntos y complementación de errores para el pronóstico de la velocidad del viento a corto plazo asistido por la plataforma Flink	-Redes LSTM -Flink Platform -Algoritmos de agrupamiento.	Coeficiente GRU K-mean VMD
4	(Xiyun Yang, Liwei Fan, Xiangjun Li, & Meng, 2022)	Estrategia de programación y licitación del mercado diario y en tiempo real para la participación en la energía eólica basada en el almacenamiento compartido de energía	-Monte Carlo -QGA Algoritmo Genético Cuántico. -método Kantorovich -CPLEX	R, MSA
5	(Elena Dumitrescu, Sullivan Hué, Christophe Hurlin, & Tokpavi, 2022)	Aprendizaje automático para la calificación crediticia: mejora de la regresión logística con efectos de árbol de decisiones no lineales	-Arboles de decisión -predicción binaria -Monte Carlo -Bosques aleatorios	PLTR, SVM
6	(Kyung Keun Yun, Sang Won Yoon, & Won, 2022)	Modelo interpretable de pronóstico de precios de acciones usando regresiones de aprendizaje automático de algoritmos genéticos y la mejor selección de subconjuntos de características	-Algoritmos Genéticos -Ajuste óptimo de la curva por tramos.	RMSE, SG, POCF
7	(A.Livas-García, y otros, 2022)	Pronóstico de componentes locales del precio marginal con inteligencia artificial y análisis de sensibilidad: un estudio bajo clima tropical y energía renovable para el sureste mexicano	-Análisis hibrido sensibilizado global. -Redes neuronales -Algoritmos genéticos	AI-GSA

Tabla 1.2 Estado del Arte basados en Deep Learning

N	Referencia	Artículo	Enfoque	Indicadores
1	*(HuiLiu* & ZhihaoLong, 2020)*	Un modelo de aprendizaje profundo mejorado para predecir series temporales de precios del mercado de valores	-Series de tiempo y corrección del error -Deep Learning	-MAE, MAPE, RMSE, SDE.
2	(Jiawei Long, Zhaopeng Chen, Weibing He, Taiyu Wu, & Ren, 2020)	Un marco integrado de aprendizaje profundo y gráfico de conocimiento para la predicción de la tendencia del precio de las acciones: una aplicación en la bolsa de valores del mercado chino	- Predicción de tendencia bursátil -Red neuronal profunda -Gráfico de conocimiento	DSPNT, LSTM, RF-MT y AB-MT.
3	(Indranil Ghosh, Chaudhuri, Alfaro-Cortés, & Gámez, 2022)	Un enfoque híbrido para pronosticar los precios de futuros con consideración simultánea de la optimización en la selección de características del conjunto e inteligencia artificial avanzada	-bosques aleatorios -precios futuros y puntuales -I.A.	IA, TI, DA, NSE
4	(Minggang Wang, y otros, 2018)	Un método híbrido novedoso para pronosticar los precios del petróleo crudo utilizando ciencia de redes complejas y algoritmos de inteligencia artificial	-red neuronal compleja -precios del crudo -Algoritmos de inteligencia artificial	DFN, BPNN, MAPE, RMSE, DMS, DFN´s
5	(Antonopoulos, Robu, Benoit Couraud, Desen Kirli, & Norbu, 2020)	Enfoques de inteligencia artificial y aprendizaje automático para la respuesta del lado de la demanda de energía: una revisión sistemática	-Aprendizaje automático -Redes neuronales artificiales -Sistemas multi agente.	MCTS, GP, GA, ML, NSGA
6	(Deepak Singhal & Swarup, 2011)	Pronóstico del precio de la electricidad usando redes neuronales artificiales	-Redes neuronales artificiales -Pronóstico de precios	MAE, RMS, MCP
7	*(WEI Wei & Chuan, 2022)*	Un método de pronóstico combinado de red neuronal gris basado en algoritmo genético	-algoritmos genéticos -redes neuronales -modelo gris	Ar, Pr, RE
8	(Ahmad Amine Loutfi, Mengtao Sun, Ijlal Loutfi, & Solibakke, 2022)	Estudio empírico de la previsión del precio al contado de la electricidad para el día siguiente: información sobre una función de pérdida novedosa para entrenar redes neuronales	-Aprendizaje automático -Redes neuronales -Funciones de pérdida -Pronóstico diario	FFNN, CNN, LSTM, GRU, RNN
9	(Jeovani E. Santiago López & García, 2018).	Series de tiempo de los PML´s resultantes del MECP en México.	- Calculo de la variación histórica y dinámica. - Series de tiempo.	Promedios, Desviación estándar, varianza, asimetría y curtosis.
10	(Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006)	Análisis de la relación precio marginal y demanda de electricidad mediante conglomerados.	- distancia Inter conglomerado. - regresiones lineales y no lineales	-RMSE, GAP, Coeficientes de emparejamiento S_M y coeficiente de Jaccard S_J.

Las Tabla 1.1 y Tabla 1.2 presentan la literatura relacionada con el área de pronóstico de los precios de la electricidad y otros estudios: los modelos de inteligencia artificial (IA) basados en redes neuronales artificiales (ANN), series temporales estadísticas y enfoques adoptados por la comunidad investigadora para abordar el problema de la previsión de los precios de la electricidad. Con frecuencia se han obtenido resultados sobresalientes utilizando modelos estadísticos como ARIMA (modelo Autorregresivo integrado de media móvil). Sin embargo, las ANN tienen ventajas técnicas como la gestión de la complejidad de la no linealidad, búsqueda de patrones de comportamiento y aprendizaje de la base de datos que son muy útiles en ausencia de modelos para identificar la relación entre los datos y los parámetros que afectan a la serie.

Por otro lado, diversos estudios han realizado comparaciones entre distintas arquitecturas para determinar cuál ofrece la mejor precisión en el modelado y las técnicas de aprendizaje profundo. Sin embargo, es importante destacar que las variables comúnmente empleadas en el entrenamiento de estos modelos suelen dirigirse a los precios históricos de la electricidad y del petróleo, o a la demanda eléctrica y la oferta energética (el uso de otras variables además de la reportada en la literatura es limitado o reservado).

1.8 Conclusión

Hasta la fecha, la mayoría de los estudios se han centrado en predecir el valor total del Precio Marginal Local (PML), sin embargo, hay una notable ausencia de investigaciones que analicen de manera individualizada los componentes de energía, transmisión y congestión de la red, los cuales constituyen los elementos fundamentales del precio de la energía. Esta brecha en la investigación representa una oportunidad significativa, especialmente considerando las importantes implicaciones que los componentes de congestión, energía y pérdidas pueden tener en el análisis detallado del PML.

El presente proyecto, realizado en colaboración con el INEEL, pretende mejorar la comprensión y pronóstico del precio marginal local (PML) replicable a los nodos del país, abordando la problemática con un análisis completo de la base de datos y crear un modelo capaz de manejar los datos energéticos y climatológicos del PML en código Python.

CAPÍTULO II. MARCO TEÓRICO

2.1 Introducción:

Este capítulo 2 se sumerge en el marco teórico alineado con el Precio Marginal Local (PML). Inicia con una contextualización detallada del PML, explorando sus conceptos fundamentales, importancia y aplicaciones en el sector eléctrico.

Eventualmente se presenta la teoría que comprende la rama de la inteligencia artificial; sus tipos de aprendizaje, como también se observa los procesos de validación y la metodología empleada en el proyecto identificando los fundamentos de la teoría y las técnicas.

2.2 Sector energético de México.

La reestructuración legal del sector energético mexicano, impulsada por la reforma energética y la ley de transición energética, ha generado un creciente interés en la estimación de tarifas eléctricas. Estas reformas apuntan a reducir los precios de la electricidad, mejorar los costos de los combustibles y fomentar la generación de energía a partir de fuentes limpias. El mercado eléctrico mexicano, un mercado emergente, opera bajo un modelo de bolsa de energía supervisado por el Centro Nacional de Control de Energía (CENACE). Este mercado permite transacciones tanto en el Mercado del Día en Adelanto (MDA) como en el Mercado de Tiempo Real (MTR). Un componente clave de este mercado es el Precio Marginal Local (PML), que varía según el nodo del Sistema Interconectado Nacional (SIN) e incluye componentes de energía, pérdida y congestión. Los precios en este mercado competitivo son influenciados por varios factores, incluyendo los costos de combustible, la capacidad de transmisión y las operaciones de las plantas de energía. (Alonso & Gabriel, 2017)

Los precios de la electricidad del mercado eléctrico mexicano presentan propiedades como la inelasticidad de la demanda eléctrica, provocada principalmente por las actividades industriales y la falta de sistemas de almacenamiento de energía. Los precios de la electricidad presentan fluctuaciones que dependen de las altas demandas derivadas de las condiciones meteorológicas y las fluctuaciones en el suministro provocadas por fallas en el sistema o cortes de generación. (A.Livas-García, Tzuc, May, RasikhTariq, & Torres, 2022)

La Ley de la Industria Eléctrica (LIE) asigna al Centro Nacional de Control de Energía (CENACE) la responsabilidad del Control Operativo del Sistema Eléctrico Nacional (SEN), la gestión del Mercado Eléctrico Mayorista (MEM), y asegura un acceso equitativo y no discriminatorio a la Red Nacional de Transmisión y a las Redes Generales de Distribución. El CENACE, que antes de la Reforma Constitucional formaba parte de la CFE, se transforma a partir de su decreto de creación, el 28 de octubre de 2014, en un organismo público descentralizado (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, 2006).

El Mercado Eléctrico Mayorista (MEM) en México incluye un Mercado de Energía de Corto Plazo (MECP), que se compone del Mercado del Día en Adelanto (MDA) y el Mercado de Tiempo Real (MTR). En estos mercados, los participantes pueden enviar al Centro Nacional de Control de Energía (CENACE) sus ofertas de compra y venta de energía y servicios relacionados. CENACE, tras recibir estas ofertas, se encarga del despacho económico de las centrales eléctricas, determinando así los Precios Marginales Locales (PML) en cada Nodo P del Sistema Eléctrico Nacional (SEN), los cuales se basan en componentes de energía, congestión y pérdidas. El SEN se divide en tres sistemas principales: el Sistema Interconectado Nacional (SIN), el Sistema Baja California (BCA) y el Sistema Baja California Sur (BCS).

Un Nodo P corresponde a uno o varios nodos de conectividad de la red y cada nodo representa a una inyección o un retiro físico de energía eléctrica por lo que entre nodos P se identifican nodo(s) de origen y nodo(s) de destino. En cada Nodo P se determina el precio de la energía para las liquidaciones financieras en el Mercado Eléctrico Mayorista. El código que identifica el nodo es un acrónimo que integra la clave del Centro de Control Regional (primeros dos dígitos), la abreviatura del nombre de la instalación (siguientes tres o cuatro caracteres) y su nivel de tensión nominal en kV (KiloVolts). (Jeovani E. Santiago López & García, 2018)

2.2.1 Precio Marginal Local

El Precio Marginal Local, se define como el precio de la energía en un nodo determinado en el SEN para un periodo definido, a partir del precio marginal de energía en un Nodo P en el Modelo Comercial del Mercado, calculado por el CENACE para el MECP. El PML es el

resultado de tres componentes:

1. **Componente de Congestión Marginal**, que representa el costo marginal de congestionamiento en cada Nodo;
2. **Componente de Energía Marginal**, que representa el costo marginal de energía en el nodo de referencia del Sistema Interconectado correspondiente y;
3. **Componente de Pérdidas Marginales**, que representa el costo marginal de pérdidas en cada Nodo.

Desde el arranque del MECP para cada Sistema Interconectado, diariamente se reciben ofertas de Compra de Energía y Ofertas de Venta de Energía y Servicios Conexos (SC) antes de las 10:00 horas, con las ofertas y otros insumos calculados por el CENACE, se ejecuta el modelo de optimización de Asignación del Unidades del Mercado del Día en Adelanto (AU-MDA), el cual asigna y despacha Unidades de Central Eléctrica (UCE) en el MDA; con esta herramienta se decide el programa horario de arranques, paros, cambios de configuración, potencias de generación y asignación de los Servicios Conexos, determinando los PML´s de la energía y los precios de los SC por Zonas de Reserva. Para el caso del MTR, este se realiza con un cálculo Ex-post, empleando información real; es decir, se ejecuta el algoritmo para las condiciones operativas reales del día anterior. Los PML para el MTR Ex-post, se formulan como un problema de programación lineal que reflejan el comportamiento actual de los recursos de generación óptimo, considerando que el desempeño de los recursos de generación es también óptimo (Jeovani E. Santiago López & García, 2018).

2.2.2 Volatilidad

El precio de la electricidad esta siempre sometido a un fuerte comportamiento volátil, resultado directo de la naturaleza no almacenable a gran escala de la energía eléctrica; es decir, la demanda debe ser satisfecha en cada instante de tiempo. También, existen otros factores que contribuyen a generar variabilidad en los precios, entre otros; que la demanda eléctrica es incierta con una elasticidad-precio muy baja y, a que, la congestión en los sistemas de transmisión produce distorsiones de precio tales como el desacoplamiento de los sistemas. La volatilidad de los precios es una medida de inestabilidad en los precios futuros o incertidumbre en los movimientos de los precios futuros.

2.2.2.1 Factores de volatilidad

a) Incertidumbre de carga:

Para satisfacer la demanda de un sistema eléctrico, los requerimientos necesarios están estrechamente relacionados con las condiciones climáticas, las cuales, hasta cierto punto, resultan impredecibles. Un cambio inesperado en la temperatura climatológica conduce a que el consumo real difiera de la demanda pronosticada. Errores en los pronósticos es una de las razones principales en las oscilaciones en los PMLs y un mal funcionamiento del sistema. Asimismo, cuando el consumo real es considerablemente mayor que la demanda pronosticada, los recursos sincronizados en el sistema son insuficiente para satisfacer la demanda; por lo que, será necesario recurrir a las reservas para satisfacer las diferencias entre lo pronosticado y lo real; de esta forma, las reservas disminuirán, agravando el rendimiento general del sistema y motivando a que los PML´s aumenten de forma inesperadamente en lapsos cortos de tiempo.

b) Precios de combustible:

El precio de los combustibles empleado por las unidades generadoras es un producto con alta volatilidad que depende de las condiciones de mercado (oferta y demanda), transporte, almacenamiento, importación, etc. Los costos variables de los insumos empleados en la generación eléctrica se ven reflejados directamente en los precios de la electricidad. De esta forma, cuando alguna unidad marginal emplea cierto tipo de combustible con un precio altamente fluctuante, el PML también reflejará dichas fluctuaciones. No todos los combustibles producen efectos oscilatorios en el PML, tal es el caso de los combustibles nucleares y el carbón, los cuales son generalmente estables y solo sufren cambios substanciales en el largo plazo.

c) Interrupciones no planificadas:

La incertidumbre de carga puede provocar grandes fluctuaciones en el PML, es decir, la oferta menor que la demanda o cambios rápidos en la demanda, producen incrementos (oscilaciones) en los precios. Cuando lo anterior se combina con una indisponibilidad de generación en horas pico, los picos en los PML pueden ser muy altos. Un aumento en la carga del sistema debe ser satisfecho con unidades con costos marginales más altos, lo cual agrava el problema.

d) Límites de transmisión (congestión):

Cuando los flujos de potencia programados no pueden fluir debido a la capacidad de las líneas de transmisión, el PML de lado de la carga de un segmento congestionado podría ser altamente volátil, ocasionado porque la generación de bajo costo no puede transmitirse a las cargas.

e) Producción de energía hidroeléctrica:

La generación de energía eléctrica con centrales hidroeléctricas es económica en ciertas épocas del año, respecto con la disponibilidad del recurso, en los periodos restantes del año, la falta del recurso hídrico se compensa con unidades térmicas, lo cual puede afectar al PML.

f) Poder de Mercado:

El ejercicio del Poder de Mercado por parte de los Participantes del Mercado es una forma de manipulación de los PML que causa un incremento en la volatilidad en los precios. "Los Derechos Financieros de Transmisión" (DFT), son una herramienta que protege a sus titulares de los riesgos derivados de la volatilidad. Le otorgan a su titular el derecho y la obligación de cobrar o pagar la diferencia que resulte del valor de los Componentes de Congestión Marginal de los Precios Marginales Locales en dos Nodos P; un nodo de origen y un nodo de destino porque son instrumentos financieros que brindan protección contra los riesgos de la congestión en el Precio Marginal Local (PML) la cual se deriva de las limitaciones físicas de la red.

g) Manipulación del Mercado:

Los Participantes del Mercado pueden causar volatilidad artificial en los PML a través de:

i) Modificando la información real de sus cargas, los participantes pueden sub asignar o sobre asignar su demanda, la carga no programada puede cambiar significativamente el precio de la energía cuando las reservas son insuficientes y por;

ii) Realizando comportamiento estratégico a través de la retención de generación en los periodos de altos precios en el mercado de energía, incrementando con esto la volatilidad de los PML. Ambos comportamientos conducen a una situación poco confiable en la operación del sistema eléctrico.

2.3 Inteligencia artificial

El hombre se ha aplicado a sí mismo el nombre científico de "hombre sabio" (homo sapiens) como una valoración de la trascendencia de sus habilidades mentales, tanto para la vida cotidiana como para el propio sentido de identidad. Los esfuerzos de la Inteligencia Artificial (IA), por su parte, se enfocan en lograr la comprensión de entidades inteligentes. Por ello, una de las razones de su estudio y análisis es aprender acerca de los propios seres humanos, pero a diferencia de la Filosofía y de la Psicología, los esfuerzos de la IA están encaminados tanto a la construcción de entidades inteligentes, como a su comprensión. (Ponce, 2010).

Según (Haugeland, 1985), la inteligencia artificial es "la interesante tarea de lograr que las computadoras piensen… máquinas con mente, en su amplio sentido literal". Otra definición, según (George F. Luger & Stubblefield, 1993) la inteligencia artificial es "la rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente". Así, la IA puede entenderse como la parte de la ciencia que se ocupa del diseño de sistemas de computación inteligente, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la compresión del lenguaje, el aprendizaje, el razonamiento y la resolución de problemas.

2.3.1 Redes neuronales artificiales

Las Redes Neuronales Artificiales (RNA) se definen como sistemas de mapeos no lineales cuya estructura se basa en principios observados en los sistemas nerviosos de humanos y animales. Constan de un número grande de procesadores simples ligados por conexiones con pesos. Las unidades de procesamiento se denominan neuronas. Cada unidad recibe entradas de otros nodos y genera una salida simple escalar que depende de la información local disponible, guardada internamente o que llega a través de las conexiones con pesos. Pueden realizarse muchas funciones complejas dependiendo de las conexiones. (Ponce, 2010)

2.3.2 Fundamento biológico:

Una neurona biológica es una célula especializada en procesar información. Está

compuesta por el cuerpo de la célula (soma) y dos tipos de ramificaciones: el axón y las dendritas. La neurona recibe las señales (impulsos) de otras neuronas a través de sus dendritas y transmite señales generadas por el cuerpo de la célula a través del axón. La Figura 2.1 muestra los elementos de una red neuronal natural. En general, una neurona consta de un cuerpo celular más o menos esférico, de 5 a 10 micras de diámetro, del que salen una rama principal, el axón y varias ramas más cortas que corresponden a las dendritas. Una de las características de las neuronas es su capacidad de comunicarse. En forma concreta, las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina e integra y emite señales de salida. El axón transmite dichas señales a los terminales axónicos, los cuales distribuyen la información. (Ponce, 2010)

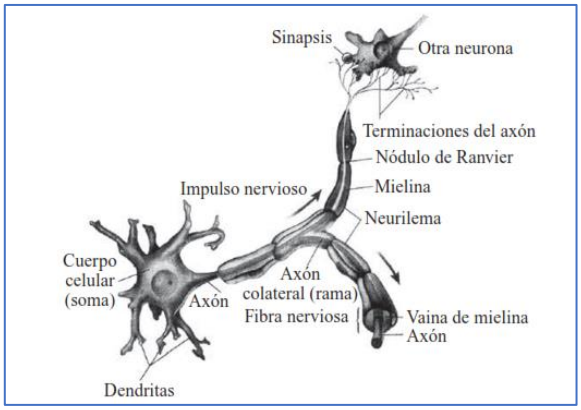


Figura 2.1. Neurona natural típica.

2.3.3 Topologías de las redes neuronales:

Según (Ponce, 2010), dos de las topologías más usadas, de acuerdo con las diferencias en la manera de realizar las conexiones, son:

a) **Redes de propagación hacia delante (feed - foward).** El flujo de información de las entradas a las salidas es exclusivamente hacia delante, extendiéndose por capas múltiples de unidades, pero no hay ninguna conexión de retroalimentación. En la Figura 2.2 se muestra una red de propagación hacia delante de 4 capas, con "*x*" entradas y "*y*" salidas.
b) **Redes Neuronales Recurrentes.** Contienen conexiones de retroalimentación, lo que puede derivarse en un proceso de evolución hacia un estado estable en el que no haya cambios en el estado de activación de las neuronas, permitiendo a la red procesar

secuencias de datos y capturar dependencias temporales en los datos de entrada. Véase Figura 2.3

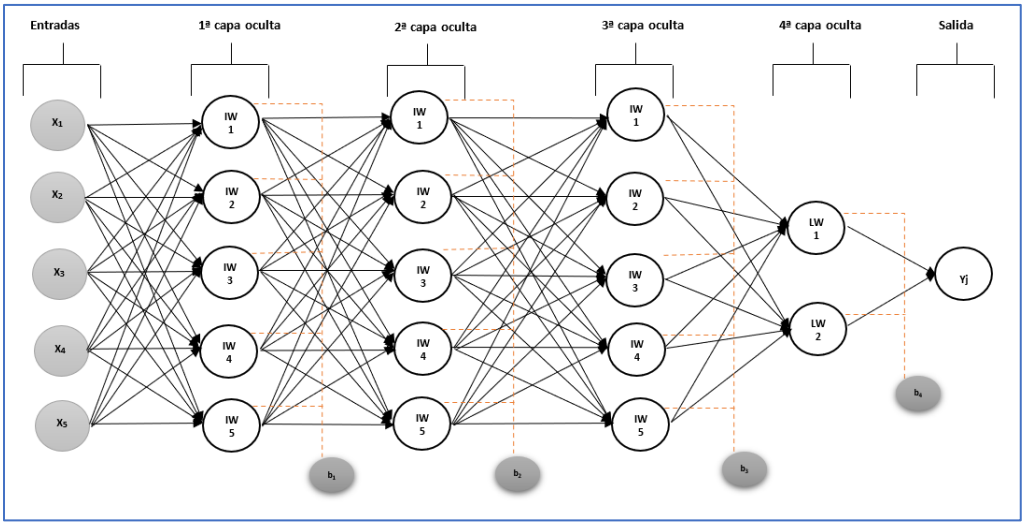


Figura 2.2. RNA de propagación hacia adelante

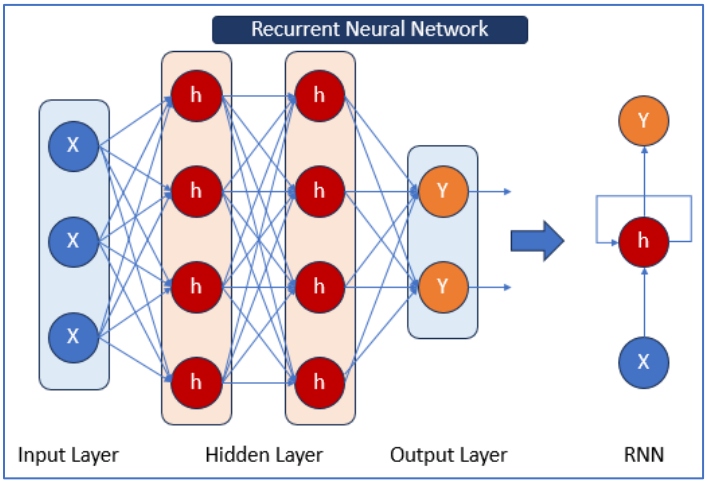


Figura 2.3 RNR (Red Neuronal Recurrente) donde la propagación forma un ciclo.

La metodología de las redes neuronales artificiales, de acuerdo con (Ponce, 2010) consta de tres pasos generales:

1. **Entrenamiento:** Etapa en la que la red "aprende", es decir modifica los pesos sinápticos para minimizar el error, pudiéndose modificar la red si es necesario.
2. **Prueba:** Se validan los resultados de la red (última capa) con valores reales.

3. **Predicción:** Ya que el modelo ha sido validado y aceptado su grado de correlación, el modelo se utiliza para predecir salidas desconocidas a partir de entradas conocidas.

2.3.4 Ventajas y desventajas

Con respecto a las ventajas y desventajas del uso de las RNA en comparación con otras técnicas, (Ponce, 2010) comenzando con las ventajas son:

- Las RNA pueden sintetizar algoritmos a través de un proceso de aprendizaje.
- Para utilizar la tecnología neural no es necesario conocer los detalles matemáticos. Sólo se requiere estar familiarizado con los datos del trabajo.
- La solución de problemas no lineales es uno de los fuertes de las RNA.
- Las RNA son robustas, pueden fallar algunos elementos de procesamiento, pero la red continúa trabajando; esto es contrario a lo que sucede en programación tradicional.

Las desventajas de las redes neurales son:

- Las RNA se deben entrenar para cada problema. Además, es necesario realizar múltiples pruebas para determinar la arquitectura adecuada. El entrenamiento es largo y puede consumir varias horas de la computadora (CPU).
- Debido a que las redes se entrenan en lugar de programarlas, éstas necesitan muchos datos.
- Las RNA representan un aspecto complejo para un observador externo que desee realizar cambios. Para añadir nuevo conocimiento es necesario cambiar las iteraciones entre muchas unidades para que su efecto unificado sintetice este conocimiento.

2.3.5 Aprendizaje automático

El Machine Learning en inglés, es una rama de la Inteligencia Artificial que se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. El desarrollador no tendrá que sentarse a programar por horas tomando en cuenta todos

los escenarios posibles ni todas las excepciones posibles. Lo único que hay que hacer es alimentar el algoritmo con un volumen gigantesco de datos para que el algoritmo aprenda y sepa qué hacer en cada uno de estos casos.

El Deep Learning es una técnica perteneciente al uso de redes neuronales de Inteligencia Artificial que con una base de datos robusta del fenómeno a analizar se estructura una red de neuronas artificiales que se entrenan para agudizar su predicción o clasificación de los datos, ya sea porque se haya programado el modelo o más interesantemente porque el propio sistema haya aprendido a modelarlas. Esto es muy útil para el pronóstico y optimización de sistemas complejos de múltiples variables que pueden representarse numéricamente para el cálculo de la red neuronal. Dado que la mayoría de los fenómenos de nuestra realidad siguen un comportamiento no lineal, sugieren que son problemas multidimensionales y entonces el objetivo es aproximar un algoritmo que capture lo más posible el comportamiento del fenómeno estudiado. Véase Figura 2.4:

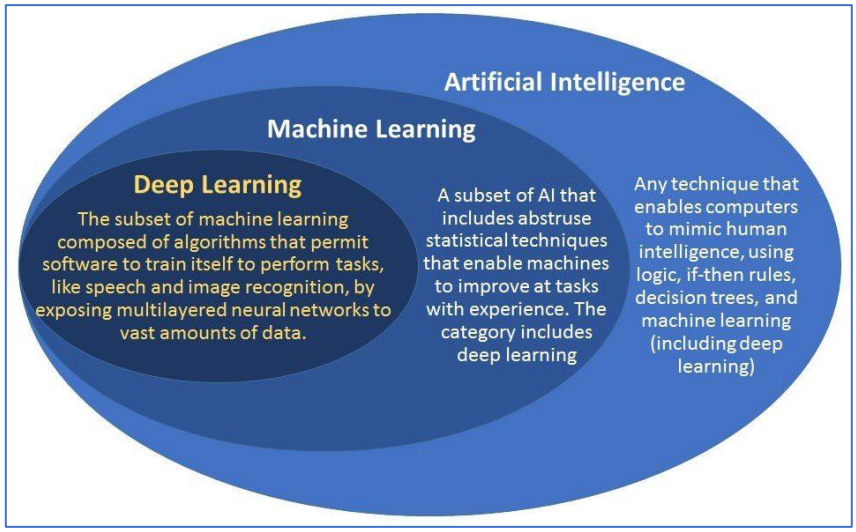


Figura 2.4. Deep Learning dentro de las técnicas de la I.A.

Estas técnicas entrenan y aprenden a partir de los datos donde hemos entrado en una tendencia a acumular más y más datos lo que se ha denominado en Big data. Se suele referenciar al proceso de análisis desde el momento en que son capturados los datos hasta que son transformados en conocimiento y para esto se requieren técnicas potentes como es el Deep Learning la cual es una versión potenciada de las redes neuronales; una familia de algoritmos que

han dado un nuevo resurgir al campo del machine Learning y por tanto al campo de la inteligencia artificial como se observa en la Figura 2.5

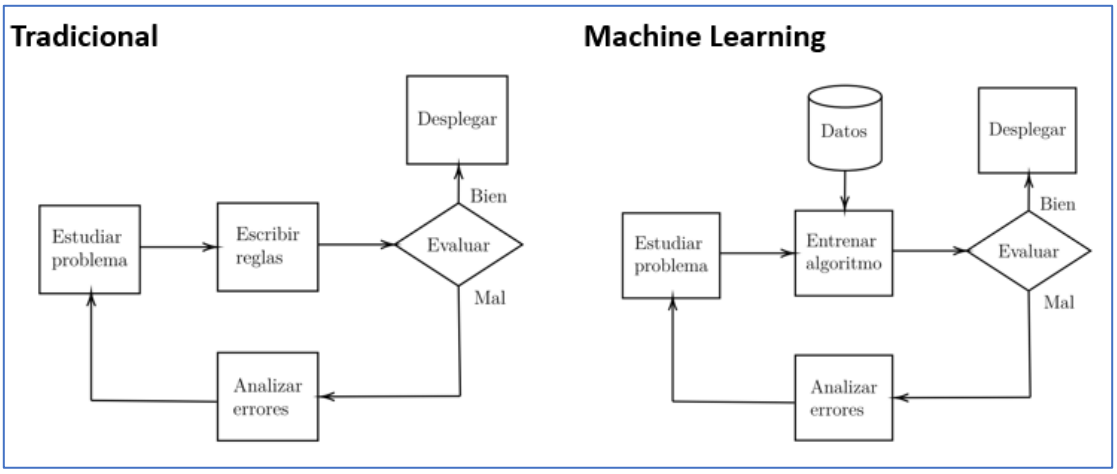


Figura 2.5. Comparativa de algoritmos de las redes neuronales tradicionales vs redes neuronales de Machine Learning

2.3.6 Métodos de aprendizaje de los algoritmos de Machine Learning

De acuerdo con (Moreno, Armengol, Béjar, Belanche, & Cortés, 1998), una clasificación de los métodos de aprendizaje en IA, considerando el tipo de estrategia y las ayudas que recibe un sistema de aprendizaje, son (ver Figura 2.6):

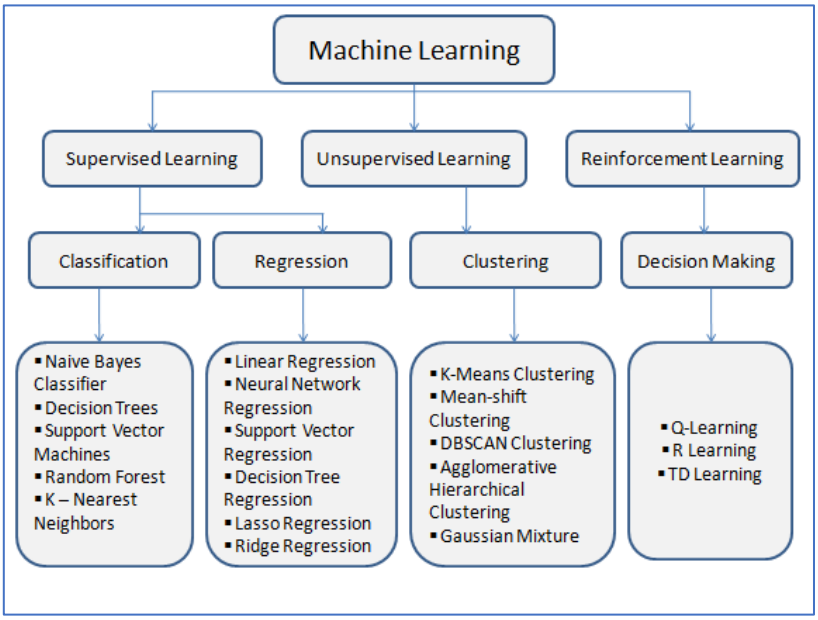


Figura 2.6. Tipos de algoritmos del Machine Learning

2.3.6.1 Aprendizaje supervisado

La suposición fundamental de este tipo de método es que los ejemplos proporcionados como entrada son necesarios para cumplir las metas del aprendizaje. Es como aprender con un profesor; en este tipo de método se dan ejemplos y se especifica de que concepto lo son. En este tipo de aprendizaje hay dos algoritmos:

a) **Algoritmo de clasificación.** Se espera que el algoritmo nos diga a qué grupo pertenece el elemento en estudio. El algoritmo encuentra patrones en los datos que le damos y los clasifica en grupos, luego compara los nuevos datos y los ubica en uno de los grupos y es así como puede predecir de que se trata.
b) **Algoritmo de regresión.** En este método lo que se espera es un número. No lo ubica en un grupo, sino que devuelve un valor específico.

2.3.6.2 Aprendizaje no supervisado

Son un enfoque en el aprendizaje automático donde los algoritmos analizan datos sin la presencia de etiquetas o información de salida previamente conocida. El objetivo principal es descubrir patrones, estructuras o relaciones intrínsecas en los datos sin depender de instrucciones externas. En lugar de trabajar con un conjunto de datos etiquetado, donde el algoritmo se guía por las respuestas correctas, en el aprendizaje no supervisado, el modelo explora y encuentra información inherente, como similitudes, diferencias o agrupaciones, de manera autónoma. Los métodos comunes de aprendizaje no supervisado incluyen el clustering (agrupamiento) y la reducción de dimensionalidad.

2.3.6.3 Aprendizaje mediante refuerzos

Este método de aprendizaje está a medio camino entre los dos anteriores. Al sistema se le proponen problemas que debe solucionar. El aprendizaje se realiza únicamente por prueba y error maximizando una recompensa, es un refuerzo positivo si el algoritmo consigue un avance en

dirección al objetivo y puede ser un refuerzo negativo que es cuando el algoritmo ahora no consiguió alcanzar su objetivo y entonces recibe un refuerzo que le indica que esa época es incorrecta y la deje de hacer, promoviendo que se dirija a las épocas donde consigue acercarse al objetivo y consiga refuerzos positivos.

2.3.7 Fases del Machine Learning

Según (Sandoval & Judith, 2018), las fases para el desarrollo del M.L. son: fase de entrenamiento y fase de prueba.

2.3.7.1 Fase de entrenamiento.

En esta fase se tiene una cantidad enorme de datos, de la cual se separa una parte para entrenar al algoritmo y darle toda esta información para que encuentre los patrones necesarios y después pueda hacer predicciones, se validan los pesos de cada neurona aportando su cálculo de una suma ponderada que contrastará con la función de costo en la capa final y repitiéndose las veces que se hayan programado, (véase Figura 2.7):

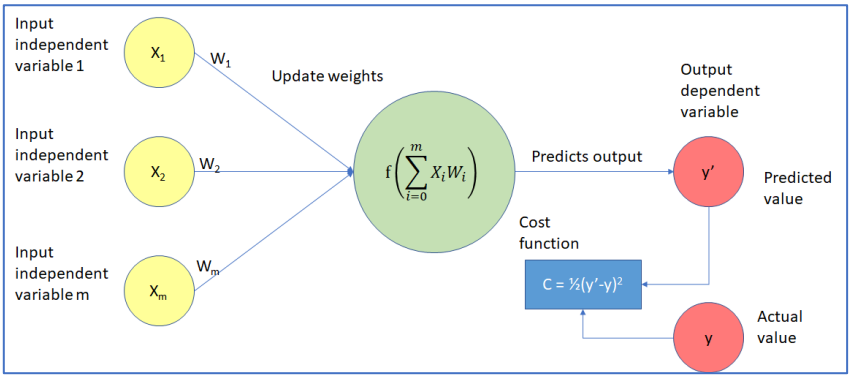


Figura 2.7. Entrenamiento de una neurona artificial.

El proceso de entrenamiento también es crucial para el desarrollo de modelos predictivos eficientes. En este paso, se dividió el conjunto de datos en tres partes esenciales: entrenamiento, validación y prueba. Esta división es fundamental para evaluar la efectividad del modelo en diferentes escenarios y para evitar el sobreajuste. La división de datos es en:

- **Conjunto de Entrenamiento:** Utilizado para "enseñar" al modelo. Aquí, el algoritmo ajusta sus parámetros internos para hacer las mejores predicciones posibles basadas en los datos de entrada.

- **Conjunto de Validación:** Se emplea para ajustar los hiperparámetros del modelo y validar su desempeño durante el entrenamiento. Este conjunto ayuda a asegurar que el modelo no solo se ajuste a los datos de entrenamiento, sino que también generalice bien a nuevos datos.
- **Conjunto de Pruebas:** Reservado para evaluar el rendimiento final del modelo. Esta evaluación se realiza después de que el modelo ha sido entrenado y validado, proporcionando una medida del desempeño del modelo en datos no vistos previamente.

Cada algoritmo seleccionado fue entrenado y validado utilizando estas divisiones. Esto implicó ajustar los modelos al conjunto de entrenamiento y evaluar su desempeño con "*TimeSeriesSplit*" de "*scikit-learn*" que itera este proceso para mejorar la precisión del modelo.

2.3.7.2 Fase de prueba.

El resto de los datos que quedan, se van a usar para hacer las pruebas. Así le podemos hacer preguntas al algoritmo y evaluar si las respuestas están bien o mal, y saber si está aprendiendo o no. Si vemos que no coinciden los datos, tendremos que agregar más datos o cambiar el método que estamos utilizando véase la Figura 2.8. Pero si se observa que hay entre un 80% a 90% de respuestas correctas, podemos decir que hay un buen grado de aprendizaje y poder utilizar ese algoritmo véase la Figura 2.9

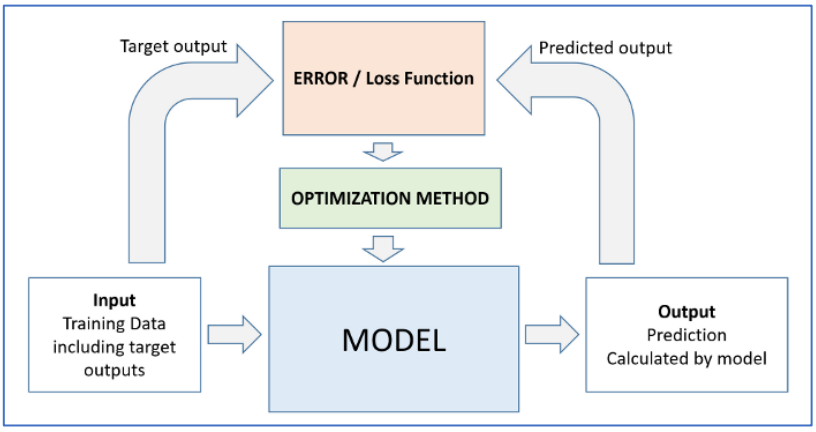


Figura 2.8. Fase de prueba de un modelo de redes neuronales.

Hoy en día, la Inteligencia Artificial y el Aprendizaje Automático están en una etapa de constante crecimiento, serán seguramente los ejes principales para el desarrollo de la ciencia de la computación y la humanidad donde estarán fundidas, no sólo hardware y software, sino también varias tecnologías, tales como nanotecnología, computación cuántica, automatización, entre otras. (Díaz Ramírez, 2021).

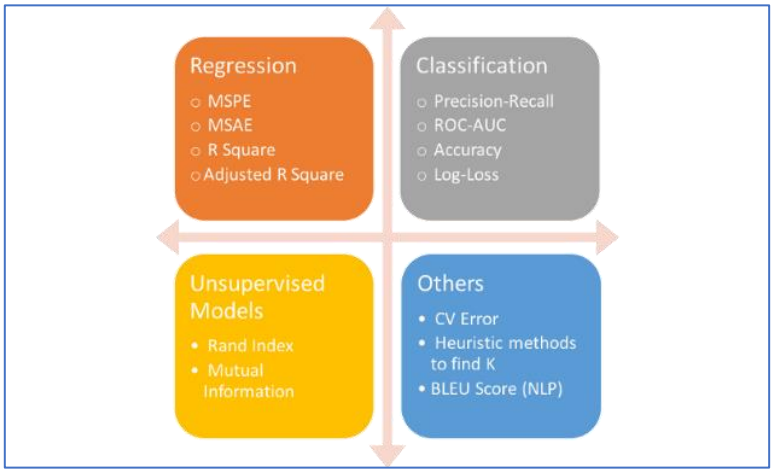


Figura 2.9. Métricas de desempeño para validar los modelos de Machine Learning.

2.4 Métricas de los errores en pronostico

1) **MAE (Error Absoluto Medio):** Mide la magnitud promedio de los errores en un conjunto de pronósticos, sin considerar su dirección. Es la media de las diferencias absolutas entre las predicciones y las observaciones reales, proporcionando una idea de la magnitud de los errores de pronóstico.
2) **MSE (Error Cuadrático Medio):** Calcula el promedio de los cuadrados de los errores. El cuadrado de los errores significa que los errores más grandes tienen un efecto desproporcionadamente grande en el MSE, lo que puede ser útil para resaltar problemas significativos en el modelo de pronóstico.
3) **RMSE (la Raíz del Error Cuadrático Medio):** Se calcula como la raíz cuadrada de la media de los errores al cuadrado entre las predicciones del modelo y los valores observados. El RMSE proporciona una medida de la dispersión de estos errores, expresando la magnitud promedio de las discrepancias entre las predicciones y los datos reales.
4) **MAPE (Error Porcentual Absoluto Medio):** Ofrece una perspectiva de los errores como un porcentaje; esto es útil para comparar la precisión de modelos de diferentes escalas.

2.5 Árboles de Decisión para Regresión

Estos modelos son fundamentales en el aprendizaje supervisado y se eligieron por su capacidad para manejar datos no lineales y su facilidad de interpretación. En Python, se implementaron utilizando bibliotecas como "*scikit-learn*", que proporciona herramientas eficientes para el análisis de datos y la modelización estadística. Para construir y entrenar un árbol de decisión, se siguieron los pasos convencionales, que incluyen importar la biblioteca necesaria, crear una instancia del modelo y ajustar el modelo a los datos de entrenamiento. La estructura resultante permite realizar predicciones en nuevos conjuntos de datos.

2.6 Regresión Polinomial de Segundo Grado

Para capturar las complejas relaciones no lineales entre las variables y el precio marginal local. La implementación de este modelo en Python permite una perspectiva más matizada que la regresión lineal simple. Este método implica transformar las características originales en términos polinomiales y ajustar el modelo resultante a los datos de entrenamiento. A través de la introducción de términos cuadráticos, el modelo puede capturar patrones no lineales en los datos. Sin embargo, es esencial considerar la posible aparición de sobreajuste al seleccionar el grado del polinomio por lo que se sugiere que sean al cuadrado para evitar el sobre ajuste. La Regresión Polinomial de Segundo Grado ofrece, así, una herramienta flexible para modelar relaciones no lineales en datos complejos.

2.7 SARIMA (Seasonal AutoRegressive Integrated Moving Average)

Este modelo enfocado a series temporales para incorporar tanto las relaciones lineales como estacionales en los datos. En Python, se utilizó la biblioteca "*statsmodels*" para su implementación, aprovechando su capacidad para modelos estadísticos complejos. Las órdenes para los términos Auto regresivos, Integrados y de Promedio Móvil se denotan como ((p, d, q)) y se definen de la siguiente manera:

- **p (Orden de AR):** Representa la cantidad de términos auto regresivos en el modelo, indicando cuántas observaciones pasadas se tienen en cuenta al predecir la siguiente.

- **d (Orden de diferenciación):** Indica la cantidad de veces que se ha aplicado la diferenciación para hacer estacionaria la serie temporal. La diferenciación es una técnica para estabilizar las fluctuaciones en los datos.
- **q (Orden de MA):** Refleja la cantidad de términos de promedio móvil, que representan la relación entre una observación y un error residual de una observación anterior.

Las órdenes para los términos Estacionales ((P, D, Q, S)) se definen asi:

- **P (Orden de SAR):** Indica la cantidad de términos estacionales auto regresivos en el modelo, mostrando la influencia de las observaciones estacionales pasadas en las predicciones actuales.
- **D (Orden de diferenciación estacional):** Representa la cantidad de veces que se ha aplicado la diferenciación estacional para hacer estacionaria la serie temporal estacional.
- **Q (Orden de SMA):** Indica la cantidad de términos estacionales de promedio móvil en el modelo, describiendo la relación entre una observación y el error residual estacional de una observación anterior.
- **S (Longitud del ciclo estacional):** Especifica la duración del ciclo estacional, indicando cuántas observaciones conforman un ciclo completo.

Estos parámetros de órdenes se determinan típicamente mediante bibliotecas como "*Auto Arima"* que utilizan técnicas automatizadas para encontrar los mejores valores según el análisis de los datos de series temporales.

2.8 LSTM (Long Short-Term Memory)

La estructura fundamental de una LSTM incluye celdas de memoria con puertas que regulan el flujo de información, permitiendo que la red retenga información relevante a lo largo de secuencias temporales extensas. Cada celda de memoria puede aprender activamente qué información retener, olvidar o escribir, lo que las hace particularmente efectivas para modelar patrones complejos y dependencias a largo plazo en series temporales. La implementación de redes LSTM se facilita mediante bibliotecas especializadas como "TensorFlow" y "Keras" en Python. Estas bibliotecas ofrecen interfaces de alto nivel que permiten construir, entrenar y compilar modelos de manera eficiente.

2.9 LSTM Combinado con PROPHET

Esta combinación innovadora buscó aprovechar las fortalezas de LSTM en el aprendizaje de secuencias temporales y la capacidad de PROPHET, centrado en la adición de componentes como tendencias y patrones estacionales, demuestra ser efectivo para modelar la complejidad de datos temporales. Se destaca por su capacidad para capturar dependencias a largo plazo, se crea un enfoque híbrido y robusto que aprovecha lo mejor de ambas técnicas.

Cada uno de estos modelos fue cuidadosamente seleccionado y diseñado para encontrar la mejor estrategia de modelado para predecir con precisión los precios marginales locales.

2.10 Conclusión

El capítulo 2 contiene los conceptos para la construcción del marco teórico necesario para el pronóstico del Precio Marginal Local (PML). Desde una comprensión profunda del PML dentro del contexto del Mercado Eléctrico Mayorista (MEM), hasta la metodología y modelos aplicados en el proyecto, destacando las etapas cuidadosamente estructuradas para abordar efectivamente la complejidad y el carácter dinámico de los precios de la energía.

CAPÍTULO III. METODOLOGÍA

3.1 Introducción

El presente capitulo es la aplicación práctica de la metodología, donde los conceptos técnicos se transforman en acciones concretas y se materializan en soluciones analíticas reales. Este capítulo detalla el proceso de implementación de la metodología, comenzando desde el manejo inicial de los datos y algoritmos, hasta la construcción y ajuste de los modelos predictivos.

3.2 Pasos de la Metodología

A continuación, se muestra la metodología del modelo inteligente basado en Machine Learning, la Figura 3.1 simplifica el proceso de construcción del modelo de ML:

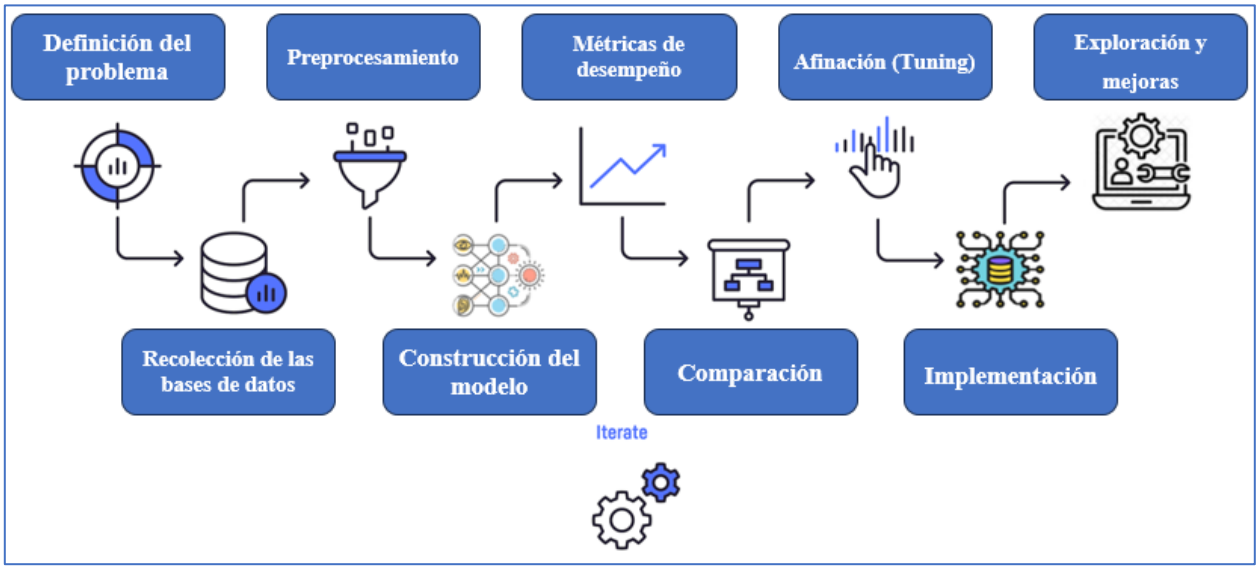


Figura 3.1. Proceso de construcción de un modelo de Machine Learning.

El proceso para aplicar aprendizaje automático puede variar dependiendo del problema específico, las herramientas disponibles y los datos con los que se dispongan. Sin embargo, a nivel general, estos son los pasos para seguir:

3.2.1 Definición del problema:

Es importante tener una comprensión clara de lo que se busca resolver, y dado que el PML

es un valor numérico, continuo y de características de una serie temporal, entonces; estamos ante un problema de regresión para un pronóstico del PML de nodos interconectados de la red nacional. La solución requiere de un enfoque de aprendizaje supervisado para su predicción precisa, con un horizonte de predicción de hasta 48 horas. Esto no solo implica la comprensión de los datos históricos, sino también la capacidad de modelar sus variables y las fluctuaciones dentro de estos datos.

3.2.2 Recolección de las bases de datos:

En este paso, se realiza la recopilación de dos tipos de datos cruciales: los registros horarios del Precio Marginal Local (PML) y las variables climatológicas, ambos en formato "csv". Los datos del PML son obtenidos del Centro Nacional de Control de Energía (CENACE), donde se presentan de manera numérica y tabulada. Además, se recolectan datos climatológicos provenientes de la Comisión Nacional del Agua (CONAGUA), lo que implica un desafío adicional debido a la variabilidad en la operatividad de las estaciones climatológicas y la consecuente dificultad en la obtención de datos consistentes y completos.

3.2.3 Preprocesamiento

Este paso es crucial para garantizar la calidad y utilidad de los datos en la modelización. Se empleó Python en un ambiente para diseñar código llamado "Jupyter", un lenguaje de programación poderoso en el análisis de datos, para realizar el preprocesamiento. El proceso comenzó con la lectura y unificación de los archivos en un *DataFrame*, una estructura de datos bidimensional y flexible de Pandas, ideal para manejar y analizar conjuntos de datos complejos.

3.2.4 Construcción del modelo

En esta etapa, la selección y aplicación de diversos algoritmos de aprendizaje automático y modelado estadístico fueron cruciales para abordar las características particulares y la complejidad de los datos del PML. Se utilizó el lenguaje de programación Python, reconocido por su robustez y amplia biblioteca de herramientas para análisis de datos y aprendizaje automático, para implementar y evaluar los modelos de:

1. Arboles de decisión para regresión
2. Regresión polinominal de segundo grado
3. SARIMA
4. LSTM (Long Short-Term Memory)
5. LSTM con PROPHET

3.2.5 Métricas de desempeño

En esta etapa, el enfoque se centra en la evaluación rigurosa de los modelos de pronóstico desarrollados. Se utilizan las métricas de desempeño estandarizadas para medir la exactitud de modelos de regresión por medio de las bibliotecas de Sciklearn. Estas métricas incluyen:

1. MAE (Error Absoluto Medio)
2. MSE (Error Cuadrático Medio)
3. RMSE (la Raíz del Error Cuadrático Medio)
4. MAPE (Error Porcentual Absoluto Medio)
5. Validación con Time Series Split
6. Walk-Forward Validation

La aplicación de estas métricas y técnicas de validación específicas garantiza que el modelo no solo sea preciso y fiable en la predicción del PML para un nodo específico, sino que también tenga la robustez y adaptabilidad necesarias para ser aplicado en diferentes nodos del sistema energético nacional.

3.2.6 Comparación del modelo

Es llevar a cabo una comparativa entre diferentes algoritmos de Machine Learning que han sido desarrollados para el pronóstico de PML´s (no existe un modelo que funcione mejor para todo tipo de problema, es necesario hacer una búsqueda y probar distintos algoritmos para el conjunto de datos).

Para lograr una comparación objetiva y cuantitativa, se propone una tabla comparativa donde se presentan las métricas de rendimiento clave, incluyendo el Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE), la Raíz del Error Cuadrático Medio (RMSE) y el

Error Porcentual Absoluto Medio (MAPE) para cada modelo. Esta tabla permite no solo visualizar el rendimiento de cada modelo en términos de precisión de predicción, sino también entender sus fortalezas y limitaciones en el contexto específico del pronóstico de PML.

3.2.7 Afinación (Tuning)

En esta fase, se lleva a cabo un exhaustivo proceso de afinación y optimización de hiperparámetros para los modelos basados en redes neuronales LSTM y la combinación de LSTM con PROPHET. El objetivo es refinar estos modelos para maximizar su precisión y eficiencia en la predicción de los PML.

- **Análisis de Sensibilidad en Redes LSTM:** Se experimenta con dos arquitecturas diferentes de redes LSTM, realizando modificaciones manuales en sus hiperparámetros. Esta tarea se repite en 10 iteraciones diferentes para cada arquitectura, registrando meticulosamente los cambios en las métricas de MAPE y MAE para cada configuración. Este análisis de sensibilidad permite identificar las configuraciones más efectivas y cómo los ajustes específicos de hiperparámetros influyen en el rendimiento del modelo.
- **Búsqueda Grid en LSTM Secuencial + PROPHET:** Para el modelo híbrido de LSTM con PROPHET, se implementa un enfoque más sistemático a través de una búsqueda grid. Esta técnica de optimización explora automáticamente una amplia gama de combinaciones de hiperparámetros, permitiendo identificar la configuración óptima que ofrece los mejores resultados en términos de precisión predictiva.

Este proceso de ajuste de hiperparámetros mejora significativamente el desempeño de los modelos en la tarea de pronóstico, además de proporcionar curvas de aprendizaje valiosos sobre el aprendizaje de las redes neuronales.

3.2.8 Implementación

Esta fase realiza la aplicación de los cinco algoritmos en 28 nodos representativos de los tres sistemas de interconexión en México, se lleva a cabo un análisis utilizando los árboles de

decisión para regresión y matrices de correlación. Este análisis tiene como objetivo identificar qué variables tienen mayor influencia en la predicción de precios en los distintos nodos. Los árboles de decisión, a pesar de haberlos utilizado en un contexto de regresión, demuestran ser una herramienta valiosa no solo para predecir precios, sino también para entender las relaciones subyacentes entre las variables.

3.2.9 Exploración y mejoras

En esta etapa complementaria, se crea una Interfaz Gráfica de Usuario (GUI) como una herramienta visual e interactiva con el analista para ayudar en la toma de decisiones, previamente a correr el código; usando "*Tkinter*" se propone una herramienta para visualizar los datos en una gráfica que le permite seleccionar diferentes periodos y guardarlos, si así se desea. Una vez visualizados se regresa al menú para correr el código *Python* al oprimir el botón que ejecuta el código en entorno "*Jupyter*" notebook. Finalmente se realiza la elaboración del documento de tesis y la presentación de resultados.

El Machine Learning es un proceso iterativo. Con base en los resultados y el *feedback* de ideas, es común regresar a pasos anteriores en la metodología para hacer ajustes y mejoras. Es importante tener en cuenta que estos pasos pueden variar o mezclarse dependiendo del proyecto cuando las circunstancias sean complejas o por factibilidad.

3.3 Definición del problema (panorama inicial)

Partiendo de que en México existen más de 2,300 nodos en la república, y estos nodos pertenecerán a uno de los 3 sistemas de interconexión eléctrica que hay en el país (SIN, BCA y BCS) (CENANCE, 2022) y el objetivo esencial es predecir a 48 horas hacia adelante el Precio Marginal Local de un nodo, entonces para fines prácticos, se mostrarán la información y resultados del nodo "07LPZ-115" (nodo de La Paz, Baja California Sur) el cual se eligió por tener datos del año 2023; El interés es conocer el comportamiento a corto plazo o en una hora determinada del día para el mercado, entonces el año más reciente podría ser más relevante que cualquier otro. Y dado que los modelos de Machine Learning soportan grandes cantidades de

datos para aprender, se consideran datos de las 720 horas recientes para que los modelos "aprendan" del comportamiento histórico pero reciente de los precios. Los valores del precio marginal son de manera horaria y en pesos mexicanos para el Mega Watt/hora ($/MWh).

Además, se integran los datos meteorológicos; se obtienen de la página de CONAGUA (Comisión Nacional del Agua) gracias a sus estaciones de monitoreo climatológico disponibles y a que estos datos están disponibles en formato "Excel" y en "csv". Los archivos de CONAGUA vienen limitados en cuanto a datos faltantes; por lo que se requerirá de una imputación de datos como un pretratamiento de estos para coincidir en la granularidad horaria para la base de datos que el modelo utilizará (los datos del PML y los datos meteorológicos).

Los archivos de datos, tanto de los registros del Precio Marginal Local (PML) como de las condiciones climáticas relevantes para la zona en estudio, se almacenan en el mismo directorio que el código Python encargado de realizar el análisis y pronóstico. Este código implementará algoritmos avanzados que se entrenarán con base en estos registros, extrayendo patrones y correlaciones críticas para la predicción del PML. Siguiendo la metodología detallada en este capítulo, se procederá con la aplicación de estos algoritmos y la posterior evaluación de su desempeño a través de métricas específicas de error, las cuales serán determinantes para la selección del modelo más adecuado y preciso.

3.4 Recolección de las bases datos

Los datos tienen la siguiente estructura tabular al recolectarla de las fuentes (CENACE y CONAGUA), véase Figura 3.2.

claveNodo	fechaHoraMDA	hora	precioMargi	componente	compontent	componente	sistema	claveParticipante
07LPZ-115	08/06/2023	7	4397.3	4429.03	-31.73	0	BCS	G001
07LPZ-115	08/06/2023	8	4517.39	4579.89	-62.5	0	BCS	G001
07LPZ-115	08/06/2023	9	4316.99	4392.87	-75.88	0	BCS	G001
07LPZ-115	08/06/2023	10	2429.25	2493.51	-64.26	0	BCS	G001
07LPZ-115	08/06/2023	11	2443.68	2514.68	-71	0	BCS	G001
07LPZ-115	08/06/2023	12	3661.65	3768.8	-107.15	0	BCS	G001
07LPZ-115	08/06/2023	13	2363.45	2433.08	-69.64	0	BCS	G001
07LPZ-115	08/06/2023	14	3747.66	3860.91	-113.26	0	BCS	G001
07LPZ-115	08/06/2023	15	2400.82	2469.91	-69.1	0	BCS	G001
07LPZ-115	08/06/2023	16	4502.2	4578.81	-76.61	0	BCS	G001
07LPZ-115	08/06/2023	17	5402.36	5468.45	-66.09	0	BCS	G001
07LPZ-115	08/06/2023	18	4793.22	4861.46	-68.24	0	BCS	G001
07LPZ-115	08/06/2023	19	5028.13	5079.19	-51.06	0	BCS	G001
07LPZ-115	08/06/2023	20	5649.88	5703.69	-53.81	0	BCS	G001
07LPZ-115	08/06/2023	21	5424.54	5480.77	-56.23	0	BCS	G001
07LPZ-115	08/06/2023	22	6042.31	6087.6	-45.29	0	BCS	G001
07LPZ-115	08/06/2023	23	6060.43	6087.26	-26.84	0	BCS	G001
07LPZ-115	08/06/2023	24	5930.55	5974.96	-44.42	0	BCS	G001

Figura 3.2 Datos originales del Precio Marginal Local

La “clave del nodo” es un nombre asignado al nodo y es una clave alfanumérica de iniciales y de la tensión que maneja el nodo. La “fecha Hora MDA” es la fecha de registro en aaaa-mm-dd. La “hora” corresponde a la hora del día en que se registró el dato. El “Precio Marginal Local” es la variable objetivo al cual definiremos como la variable dependiente en el análisis estadístico y de pronóstico.

Las variables “componente de energía”, “componente de pérdidas” y “componente de congestión” son variables independientes en nuestro análisis y pronóstico; el “componente de energía” es el precio de lo que cuesta generar la electricidad en ese nodo, el “componente de pérdida” corresponde al precio de lo que se pierde de energía por la distribución y otras fugas y por último el “componente de congestión” es el costo o precio de la electricidad que dada la emergencia o limitada capacidad de las líneas de transmisión se eleva el precio por la dificultad de la congestión de la red de transmisión en ese nodo. La variable “sistema” corresponde a las siglas de alguno de los sistemas de interconexión, como SIN (Sistema Interconectado Nacional), BCA (sistema Baja California Norte) y BCS (sistema Baja California Sur). Y finalmente la variable “clave del Participante” corresponde al usuario calificado dentro del sistema MEM (Mercado Eléctrico Mayorista) que puede comprar la energía de forma autorizada.

En la Figura 3.3 se presentan los datos climatológicos recolectados:

Fecha Local	Temperatura del Aire (°C)	Precipitación (mm)	Humedad relativa (%)	Presión Atmosférica (hpa)	Radiación Solar (W/m²)	Dirección del Viento (grados)	Rapidez de viento (km/h)	Dirección de ráfaga (grados)	Rapidez de ráfaga (km/h)
01/03/2023 00:00	15.1	0	40	1012.3		131	2.88	169	5.04
01/03/2023 00:10	14.9	0	37	1012.2		154	3.24	234	5.76
01/03/2023 00:15	25.8	0	68	1011.4		123	15	116	25.8
01/03/2023 00:20	14.8	0	35	1012.2		172	2.88	225	5.4
01/03/2023 00:30	15	0	34	1012.4		185	3.24	229	5.76
01/03/2023 00:40	14.9	0	37	1012.4		183	2.88	222	4.32
01/03/2023 00:45	25.3	0	71	1011.3		123	16.2	122	27.6
01/03/2023 00:50	14.9	0	37	1012.4		185	3.24	223	5.4
01/03/2023 01:00	14.8	0	38	1012.3		182	3.6	234	8.64
01/03/2023 01:10	14.6	0	36	1012.4		163	5.76	224	9.36
01/03/2023 01:15	24.9	0	74	1010.9		124	15.8	110	28.2
01/03/2023 01:20	14.4	0	39	1012.4		161	4.68	208	9.36
01/03/2023 01:30	14.4	0	41	1012.3		160	4.68	228	8.64
01/03/2023 01:40	14.4	0	42	1012.3		167	5.4	209	8.28
01/03/2023 01:45	24.6	0	76	1010.7		123	15.7	120	27
01/03/2023 01:50	14.4	0	43	1012.2		162	5.04	203	8.64
01/03/2023 02:00	14.3	0	43	1012.2		154	5.04	206	9

Figura 3.3 Datos originales climatológicos

Tenemos las siguientes variables meteorológicas:

1. Dirección del viento (grados)

2. Dirección de ráfaga (grados)
3. Rapidez del viento (km/h)
4. Rapidez de ráfaga (km/h)
5. Temperatura del aire (°C)
6. Humedad relativa (%)
7. Presión atmosférica (hpa)
8. Precipitación (mm)
9. Radiación solar (W/m^2)

Lo correspondiente es realizar el procesamiento de los datos para el manejo adecuado de su información analizando sus características para poder aplicar correctamente pronósticos con Machine Learning.

3.5 Preprocesamiento

La importancia de una preparación de los datos previo al análisis estadístico radica en que las bases de datos tienen espacios anormales (faltantes), y a veces se requiere eliminar valores atípicos y/o corregir errores en los datos. Es necesario revisar que la información a utilizar este completa, sin datos erróneos o nomenclaturas extrañas antes de probar algún modelo. Lo que se utiliza es el ambiente de "Jupyter notebook" que es un ejecutor de código que sirve para editar y visualizar tanto el código como los resultados de análisis que se realizan dentro de estos editores de código. Se inicia con la siguiente ilustración donde lo primero por hacer es siempre importar las bibliotecas que ayudan a realizar operaciones y demás recursos para el análisis y pronostico en nuestro script (código), véase la Figura 3.4:

```python
#Módulo para el cálculo numérico y el análisis de datos
import numpy as np
#Módulo para el análisis de datos tabulares
import pandas as pd
#Módulo para la visualización de datos y gráficos estadísticos
import seaborn as sns
# Módulo para la creación de gráficos 2D
import matplotlib.pyplot as plt
#Módulo para construir y entrenar RNA
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

#Módulo para Machine Learnig (utilerias para preprocesar, entrenar y evaluar modelos)
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay, RocCurveDisplay, classification_report

# Módulo para la visualización de datos y gráficos estadísticos interactivos
import plotly.express as px
```

Figura 3.4 Bibliotecas de lectura, análisis y procesamiento de datos.

Para coincidir en la granularidad horaria de la base de datos integrada por ambos archivos (los datos del PML y los datos meteorológicos), para el mismo periodo y registros horarios del PML con todas las demás variables, se aplica el comando que importe “pandas” para la lectura de los archivos “csv” y “Excel” (o “csv” como este caso), identificando la columna del tiempo, esto es porque se crea un nuevo dataset que combinará ambas bases de datos en uno solo, pero como en el archivo “csv” climatológicos, los registros vienen en un formato diez-minutesimal, se requiere promediar los valores y convertirlos a horas del día. Véase Figura 3.5:

```
In [1]: import pandas as pd

        # Lectura de archivos CSV:
        df1 = pd.read_csv("07LPZ-115.csv", parse_dates=['fechaHoraMDA'])

        # Lectura de archivos CSV con encoding diferente:
        df2 = pd.read_csv('LA PAZ_2023_03.csv', parse_dates=['Fecha Local'], encoding='latin1')

In [2]: # identificar las fechas del archivo con datos meteorológicos porque es la más limitada en las fechas:
        fecha_minima = df2['Fecha Local'].min()
        fecha_maxima = df2['Fecha Local'].max()

In [3]: # hacer que la fecha del archivo del PML coincida con el minimo y maximo del archivo de datos meteorológicos:
        df1_filtrado = df1[(df1['fechaHoraMDA'] >= fecha_minima) & (df1['fechaHoraMDA'] <= fecha_maxima)]

In [4]: # hacer que los datos meteorológicos tengan la misma granularidad que en el PML (en horas):
        df2_resampled = df2.set_index('Fecha Local').resample('H').mean().reset_index()

In [5]: # integrar en un mismo dataframe los datos de ambos archivos:
        PMLTEMP_df = pd.merge(df1_filtrado, df2_resampled, left_on='fechaHoraMDA', right_on='Fecha Local', how='inner')
```

Figura 3.5. Preprocesamiento de datos del PML:

A continuación se observa si existen datos faltantes en la base de datos, con

“print(PML.isnull().sum())”, véase la siguiente Figura 3.6:

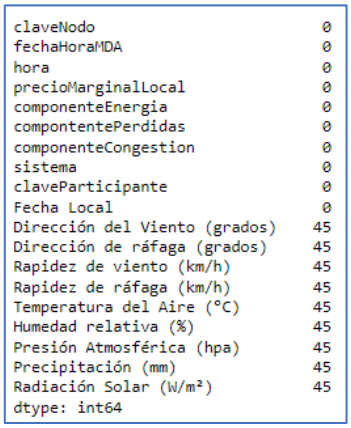


Figura 3.6. Variables disponibles del CENACE y CONAGUA

Se puede observar que 45 horas no hay registros en todas las variables meteorológicas, esto sucede porque no siempre se puede monitorear continuamente el clima en un lugar como se aprecia en esta estación meteorológica.

3.5.1 Discriminación de variables.

En el desarrollo de modelos predictivos avanzados, como los que se utilizan en este proyecto para pronosticar el Precio Marginal Local, una etapa crítica es la discriminación de variables. Esta fase implica identificar y seleccionar aquellas variables que tienen un impacto significativo en la variable objetivo, en este caso, el PML. El proceso de discriminación de variables es fundamental en Machine Learning porque se busca la reducción de la complejidad al eliminar variables irrelevantes o menos importantes. En el contexto de este proyecto, la discriminación de variables se realiza con un enfoque informado tanto por la teoría subyacente como por un análisis empírico. Esto implica considerar las características específicas de la generación de energía en México, como la importancia creciente de la energía eólica y solar, y el impacto de variables ambientales como la velocidad del viento y la radiación solar en la generación de energía.

A continuación, se describe el proceso de selección de variables para este estudio, justificando la inclusión o exclusión de ciertas variables en función de su relevancia para predecir el PML y su congruencia con el contexto energético de México.

Las primeras 4 variables son esenciales para parques eólicos de generación de energía, México ha hecho esfuerzos significativos para expandir su capacidad de generación de energía eólica, pero la proporción total de generación eólica con respecto a otras fuentes todavía no es dominante. El país ha establecido varios parques eólicos, especialmente en regiones como Oaxaca, Tamaulipas y Yucatán (CENANCE, 2022). Por lo tanto, es buena opción considerar solo una de las 4, la cual, si a mayor rapidez del viento generan más energía, es lógico elegir la variable “rapidez del viento” sobre las demás. Misma selección se considera para las variables de precipitaciones, radiación solar y temperatura del aire; para México, que cuenta con una importante capacidad de generación de energía solar y donde la temperatura y la precipitación pueden tener un impacto significativo en la demanda energética.

Como ya se mencionó se debe descartar las variables que se consideran innecesarias porque por experiencia se sabe que no aportan valor al PML, por lo que se ingresa el siguiente comando; *PML_recent.drop(["claveNodo", "fechaHoraMDA", "hora", "sistema", "claveParticipante", "Dirección del Viento (grados)", "Dirección de ráfaga (grados)", "Rapidez de ráfaga (km/h)", "Humedad relativa (%)", "Presión Atmosférica (hpa)", "Fecha Local"], axis=1, inplace=True)*

Aquí se describe que la base de datos se llamará “PML_recent” el cual se descartan las variables "claveNodo", "fechaHoraMDA", "hora", "sistema", "claveParticipante", "Dirección del Viento (grados)", "Dirección de ráfaga (grados)", "Rapidez de ráfaga (km/h)", "Humedad relativa (%)", "Presión Atmosférica (hpa)"y "Fecha Local", por qué sus valores se sabe que no tienen mayor relación con el PML más que como referencia horaria para algunos y en otros casos como los datos meteorológicos, se seleccionaron las variables más importantes para la tecnología de las plantas de generación. Véase la siguiente Figura 3.7, que muestra cómo se encuentra al final del pretratamiento a nuestra base de datos ya lista para los análisis y pronósticos:

	precioMarginalLocal	componenteEnergia	compontentePerdidas	componenteCongestion	Temperatura del Aire (°C)	Precipitación (mm)	Radiación Solar (W/m²)	Rapidez de viento (km/h)
24	2156.15	2186.28	-30.13	0.0	18.987500	0.0	0.0	11.9700
25	2111.00	2131.38	-20.37	0.0	17.987500	0.0	0.0	11.2075
26	2097.44	2116.38	-18.94	0.0	17.287500	0.0	0.0	10.5800
27	2085.23	2102.98	-17.75	0.0	16.312500	0.0	0.0	8.6625
28	2097.78	2115.33	-17.55	0.0	15.400000	0.0	0.0	8.2500
...	...	...	...	...	...	...	...	...
739	5170.10	5197.95	-27.85	0.0	21.216667	0.0	21.8	14.4000
740	4762.51	4792.67	-30.16	0.0	20.333333	0.0	0.0	11.7600
741	4718.05	4755.53	-37.48	0.0	19.983333	0.0	0.0	9.4800
742	4574.42	4603.20	-28.78	0.0	19.733333	0.0	0.0	8.7000
743	4326.22	4366.83	-40.61	0.0	19.466667	0.0	0.0	7.2600

720 rows × 9 columns

Figura 3.7. Data set de los modelos de regresión

Aquí hay una breve descripción y cómo cada variable podría influir en el precio marginal local de energía:

Radiación solar (W/m2): México tiene un gran potencial solar, y la radiación solar directamente afecta la producción de energía en las plantas solares fotovoltaicas. Una mayor radiación solar implicaría una mayor producción de energía solar, lo que podría influir en el precio de la energía.

Temperatura del aire (°C): La temperatura puede influir en la demanda de energía. Por ejemplo, días extremadamente calurosos pueden aumentar el uso de aires acondicionados, lo que lleva a un aumento en la demanda de electricidad. Por otro lado, en las raras ocasiones en que hace frío, podría haber un aumento en la demanda de calefacción en ciertas regiones. Esta fluctuación en la demanda puede influir en el precio de la energía.

Precipitación (mm): Las lluvias pueden tener un doble efecto. Por un lado, pueden disminuir la radiación solar que llega a los paneles solares, afectando la producción de energía solar. Por otro lado, si hay producción hidroeléctrica en la región, las lluvias pueden influir en el almacenamiento y flujo de agua, afectando la generación de energía.

Rapidez del Viento (km/h): La inclusión de la velocidad del viento es esencial, Aunque esta región no es predominantemente conocida por su generación eólica a gran escala como otras áreas de México, la velocidad del viento sigue siendo un indicador relevante.

Con respecto a los datos del PML donde tenemos los componentes del PML (energía, congestión y perdida) se sabe que estos elementos son los que integran en conjunto al PML de cualquier nodo, por lo que aquí una descripción de las razones:

Componente de energía: Representa el costo de generar electricidad y es el componente principal del PML. En La Paz, esto incluye la generación local y la energía importada, siendo relevante la mezcla de fuentes de energía utilizadas, como la solar, la térmica y, en menor medida, la eólica.

Componente de congestión: Este componente refleja los costos asociados con las limitaciones en la capacidad de transmisión. En áreas como La Paz, que están relativamente aisladas del Sistema Interconectado Nacional, la congestión puede ser un factor significativo, especialmente si la generación local no es suficiente para satisfacer la demanda y se requiere una mayor importación de energía.

Componente de pérdida: Se refiere a la energía perdida durante la transmisión y distribución desde el punto de generación hasta el nodo. Dado que La Paz es una región geográficamente aislada, las pérdidas pueden ser mayores en comparación con nodos más conectados al sistema principal, afectando así el PML.

Lo siguiente es determinar un horizonte de análisis de la serie de datos (serie de tiempo), aquí radica una de las ventajas que tiene el Machine Learning en Python que es capaz de usar una gran cantidad de datos para hacer cálculos y resultados de manera rápida y con buenos gráficos para análisis al usar bibliotecas de tratamiento y visualización de datos (como “Pandas”, “Keras”, “statsmodels”, etc.). Sin embargo, no es útil, ni recomendable tener datos muy antiguos o atípicos como se ha identificado hasta ahora, por lo que conviene establecer primero, un tiempo de análisis estadístico. Dado que se busca predecir un precio marginal hacia adelante en el futuro, necesitamos analizar los datos más recientes y en un periodo razonable, por lo que los datos de un mes es más que suficiente para gráficos y cálculos pues disponemos de 720 registros (24 registros * 30 días) como se logra ver en la Figura 3.8.

```
PML_recent = PML.tail(720)
PML_recent
```

	claveNodo	fechaHoraMDA	hora	precioMarginalLocal	componenteEnergia	componentePerdidas	componenteCongestion	sistema	claveParticipante
37583	07LPZ-115	2023-05-10 00:00:00.000	1	4952.61	4975.59	-22.97	0.0	BCS	G001\r
37584	07LPZ-115	2023-05-10 01:00:00.000	2	4848.43	4883.20	-34.77	0.0	BCS	G001\r
37585	07LPZ-115	2023-05-10 02:00:00.000	3	4270.87	4313.30	-42.43	0.0	BCS	G001\r
37586	07LPZ-115	2023-05-10 03:00:00.000	4	4576.85	4627.71	-50.86	0.0	BCS	G001\r
37587	07LPZ-115	2023-05-10 04:00:00.000	5	4577.38	4627.71	-50.34	0.0	BCS	G001\r
...	...	...	...	...	...	...	...	...	...
38298	07LPZ-115	2023-06-08 19:00:00.000	20	5649.88	5703.69	-53.81	0.0	BCS	G001\r
38299	07LPZ-115	2023-06-08 20:00:00.000	21	5424.54	5480.77	-56.23	0.0	BCS	G001\r
38300	07LPZ-115	2023-06-08 21:00:00.000	22	6042.31	6087.60	-45.29	0.0	BCS	G001\r
38301	07LPZ-115	2023-06-08 22:00:00.000	23	6060.43	6087.26	-26.84	0.0	BCS	G001\r
38302	07LPZ-115	2023-06-08 23:00:00.000	24	5930.55	5974.96	-44.42	0.0	BCS	G001\r

720 rows × 9 columns

Figura 3.8 Selección de los 720 registros más actuales de la base de datos.

3.5.2 Análisis estadístico.

3.5.2.1 Descomposición y análisis de la serie temporal

A continuación, se desglosan y analizan los componentes básicos de la serie temporal del PML. Esto implica crear gráficos que nos permiten visualizar no solo el PML en sí, sino también sus elementos constituyentes: la "tendencia", la "estacionalidad" y los "residuos". Para los análisis se acude a dos enfoques: la “descomposición clásica” que descompone una serie temporal en tres componentes principales: tendencia, estacionalidad y residuo. La tendencia representa la dirección general en la que los datos están cambiando a lo largo del tiempo, la estacionalidad captura patrones periódicos y repetitivos, y el residuo contiene la variación aleatoria o el ruido restante. Esta técnica es ampliamente utilizada y está implementada en varias bibliotecas, como “*statsmodels*”. El segundo enfoque: “descomposición STL” (*Seasonal and Trend decomposition using Loess*), es una técnica más avanzada que también descompone una serie temporal en tendencia, estacionalidad y residuo, pero utiliza un enfoque de suavizado local

(*Loess*) para estimar estas componentes. STL es especialmente útil cuando hay estacionalidad o cambios en la variabilidad a lo largo del tiempo. Esta técnica también está implementada en "*statsmodels*". véase la Figura 3.9.

```
# Realizar la descomposición clásica de series de tiempo con un período de estacionalidad de 2
decomposition = sm.tsa.seasonal_decompose(PML_recent['precioMarginalLocal'], model='additive', period=2)

# Realizar la descomposición STL con un período de estacionalidad de 3 horas
stl = STL(PML_recent['precioMarginalLocal'], seasonal=3)
result = stl.fit()
```

Figura 3.9 Enfoque clásico y STL para el análisis y descomposición de las series temporales.

En esta línea de código *"decomposition = sm.tsa.seasonal_decompose (PML_recent ['precioMarginalLocal'], model='additive', period=2)"* descompone la serie de tiempo del PML usando un modelo aditivo en el periodo que sea indicado como el 2 en la Figura 3.9. En el caso de STL tenemos "*STL(PML_recent['precioMarginalLocal'], seasonal=3):"* esto crea un objeto de descomposición STL utilizando la serie temporal de precios marginales locales *"(PML_recent ['precioMarginalLocal'])"* como la serie de entrada y especifica un período de estacionalidad de 3 horas. El parámetro *seasonal* en la función STL debe ser un número impar mayor o igual a 3. Esto se debe a que el algoritmo STL utiliza una ventana móvil para calcular la estacionalidad, y esta ventana debe tener un tamaño impar para garantizar que haya un punto central. Esto ayuda a calcular correctamente la estacionalidad para cada punto en la serie temporal.

Ahora bien, se aplica la descomposición de la serie al probar con 8, 12, 24 y 48, refiriéndonos a las horas en que se buscan patrones estacionales en los datos en la descomposición clásica, como se ve en la Figura 3.10:

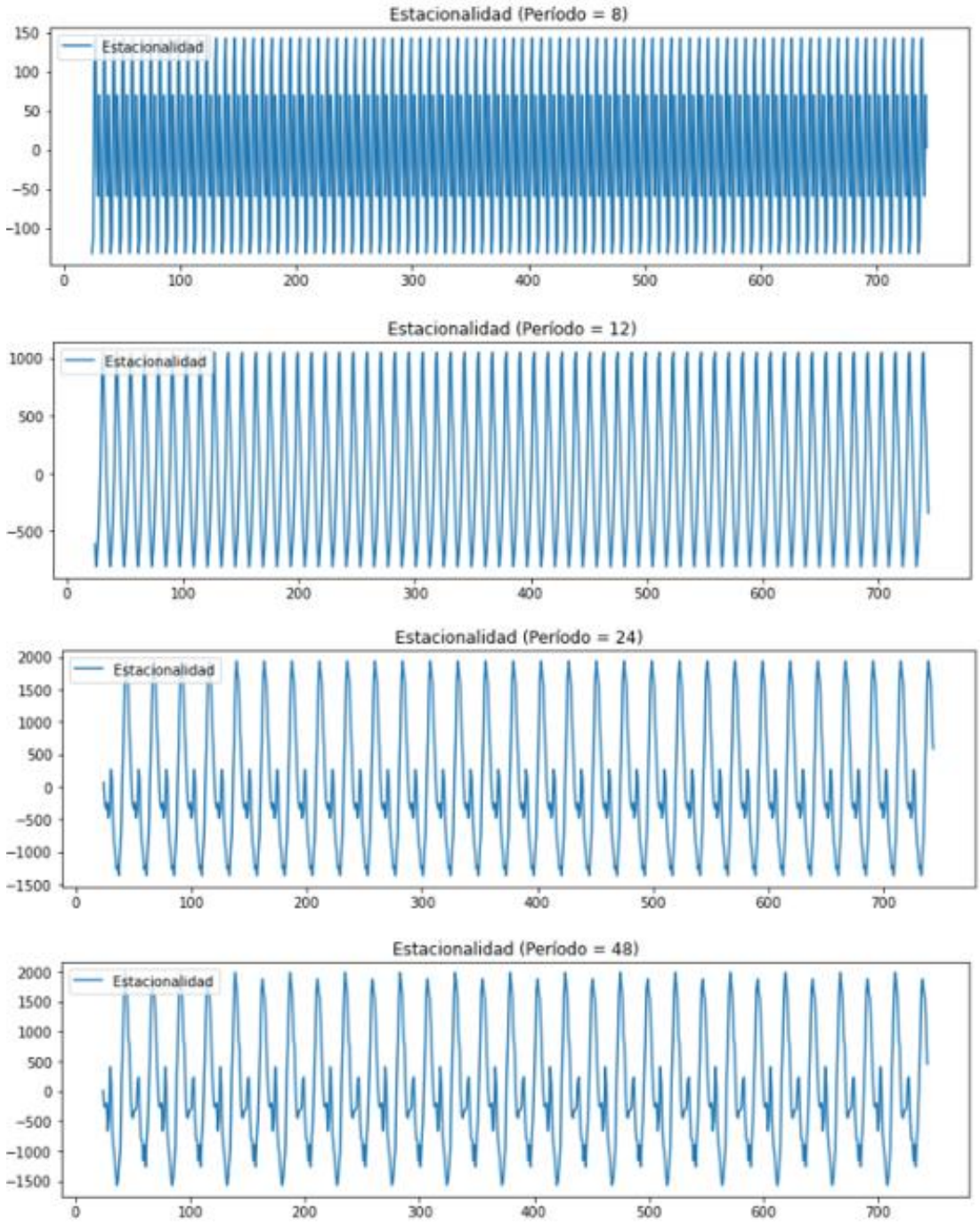


Figura 3.10 Análisis clásico de la estacionalidad mediante descomposición de series de tiempo.

La Figura 3.10 tiene 4 graficas correspondientes a las horas 8, 12, 24 y 48, busca identificar el patrón oscilante típico que normalmente adopta una forma ondulada; En el caso del período de 8 horas, se observa un patrón oscilante con una influencia interna, representada como otra oscilación dentro del ciclo de 8 horas. Esto sugiere la presencia de una estacionalidad específica dentro de este intervalo. Además, esta oscilación se encuentra dentro de un rango de $-120.^{00} a $150.^{00} pesos, siendo la menor en comparación con los otros tres períodos analizados en

la Figura 3.10 a continuación, se observa la gráfica del periodo de 12 horas en la cual se aprecia una oscilación constante sin patrones internos discernibles. El rango de oscilación abarca desde $-800.^{00} hasta $1,000.^{00} pesos, lo que sugiere la presencia de una estacionalidad regular cada 12 horas en los datos analizados. En los casos de 24 y 48 horas, se observa un comportamiento similar en ambos periodos, lo que sugiere la presencia de patrones estacionales más refinados dentro del periodo de 24 horas o más; esto se evidencia por las pequeñas oscilaciones que se presentan dentro de los periodos de 24 o 48 horas, respectivamente. Además, ambos periodos muestran una longitud de oscilación similar, que va desde $-1500.^{00} hasta $2,000.^{00} pesos. Por lo tanto, asumir una estacionalidad de 12 horas parece ser más adecuado, ya que se identifica que captura el comportamiento general que define a los precios marginales locales en este conjunto de datos. Véase Figura 3.10

Por otro lado, se complementa el análisis con una comparación con STL y como se sabe, debemos hacer impares a las horas, para seguir el mismo experimento, se consideran las siguientes horas; 7, 11, 23, y 47. En la Figura 3.11 observamos que se identifican las fechas de los datos y que, al enfocarse en una estacionalidad de 7 horas, se identifican patrones de oscilación que varían constantemente. Esto sugiere que los datos exhiben una estacionalidad cambiante cada 7 horas, con oscilaciones internas más amplias al principio de la serie. Sin embargo, hacia la mitad de los datos, el rango de oscilación tiende a reducirse. Este hallazgo indica que el enfoque de STL es efectivo para capturar las estacionalidades de los datos, aunque también señala la dificultad de generalizar la serie. En consecuencia, definir si la estacionalidad corresponde realmente a un periodo de 7 horas puede resultar un desafío. También es notable que a medida que aumenta el período de análisis, los datos tienden a generalizarse más, lo que podría implicar una pérdida de esta cualidad distintiva. Se sugiere que los datos exhiben una estacionalidad diaria, pero no más allá de las 23 horas, ya que a partir de ese punto la estacionalidad tiende a representarse más generalizada. Esto sugiere que los datos tienen una estacionalidad diaria que se aproxima a las 11 horas, ya que cerca de las 23 horas se observa una generalización. Además, es interesante notar que en todos los períodos de análisis se exhiben un rango de oscilación similar, que va desde los $-2,000.^{00} hasta los $2,000.^{00} pesos en cualquier período analizado. Véase Figura 3.11

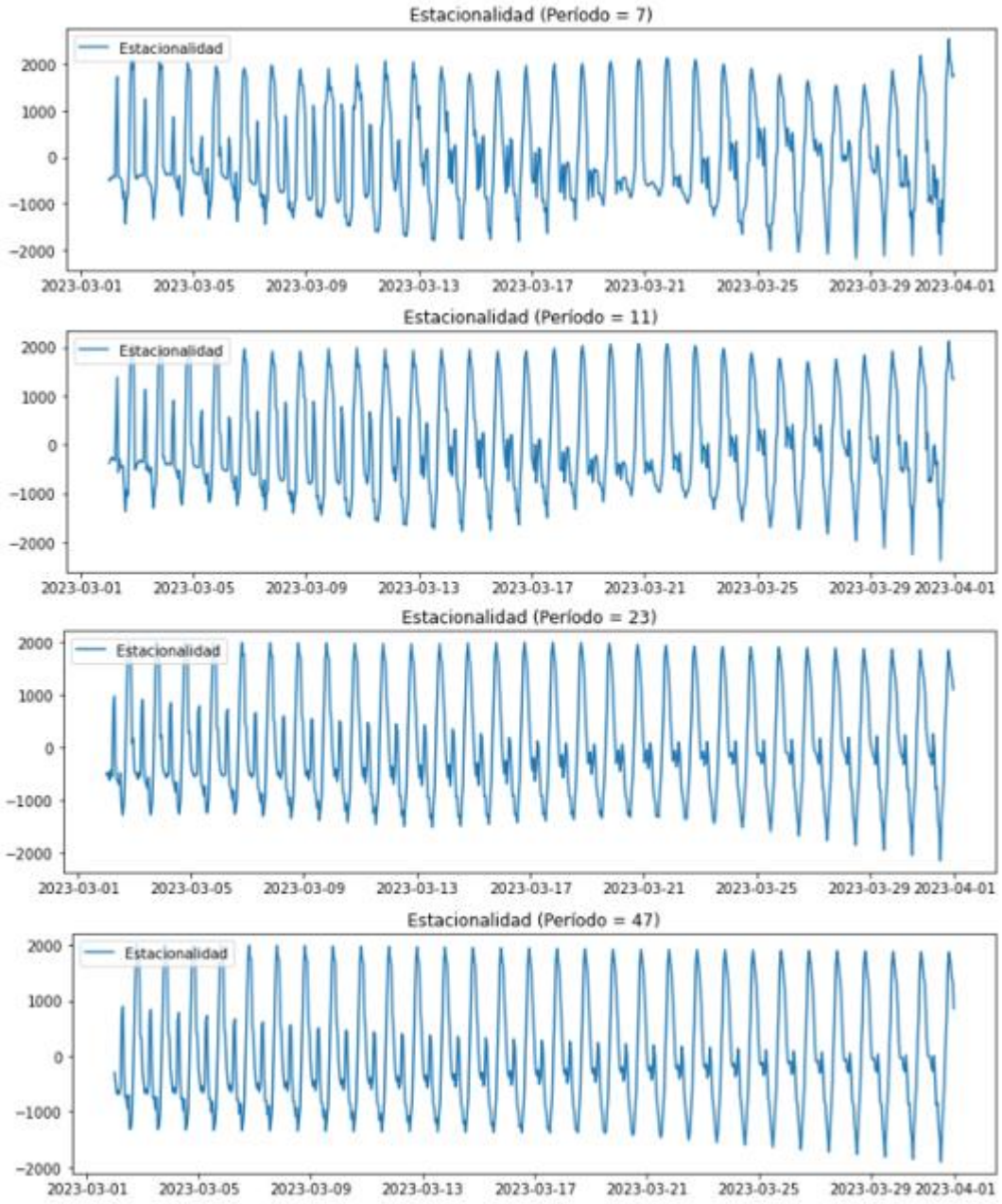


Figura 3.11 Análisis STL de la estacionalidad mediante descomposición de series de tiempo.

En este caso específico se optaría por el enfoque clásico de descomposición de series temporales por los siguientes aspectos:

- Mayor claridad en la identificación de la estacionalidad: El enfoque clásico muestra una identificación más clara y consistente de la estacionalidad, especialmente en un período alrededor de 12 horas. Las observaciones revelan un rango de oscilación definido y una ausencia de cambios significativos en otros períodos.

- Coherencia en los resultados: A diferencia del enfoque STL, que exhibe variaciones en la detección de la estacionalidad entre diferentes horas analizadas, el enfoque clásico ofrece resultados más uniformes y coherentes en la identificación de los patrones estacionales.
- Simplicidad y familiaridad: El enfoque clásico es ampliamente utilizado y comprendido en el análisis de series temporales, lo que lo convierte en una opción más accesible y fácil de interpretar.

Por lo tanto, realizado un análisis previo de 8 horarios diferentes para determinar el periodo estacional que representa a la serie temporal de los datos del nodo 07LPZ-115, ahora el análisis a los 720 registros que equivalen a un mes de datos se explicarán sus comandos para la descomposición de la serie temporal a continuación:

- "*trend*": La tendencia de la serie de tiempo.
- "*seasonal*": La estacionalidad de la serie de tiempo.
- "*residual*": Los residuos o la parte de la serie de tiempo que no se explica por la tendencia o la estacionalidad.

Para la creación de gráficos para cada componente se crea una figura con cuatro sub gráficos (usando "*matplotlib.pyplot*", importado como "*plt*"), cada uno mostrando un componente diferente de la descomposición:

- El primer gráfico (ax1) muestra la serie de tiempo original del PML.
- El segundo gráfico (ax2) muestra la tendencia extraída.
- El tercer gráfico (ax3) muestra la estacionalidad.
- El cuarto gráfico (ax4) muestra los residuos.

Finalmente el reglón que dice "*plt.tight_layout()*" es un comando que se utiliza para ajustar automáticamente los parámetros de la subtrama para que la figura se vea ordenada, y "*plt.show():*" muestra la figura con todos los sub gráficos. Véase Figura 3.12

```python
#biblioteca para  estadísticas y modelado estadístico en Python
import statsmodels.api as sm

# Realizar la descomposición clásica de series de tiempo
decomposition = sm.tsa.seasonal_decompose(PML_recent['precioMarginalLocal'], model='additive', period=12)

# Obtener los componentes
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

# Crear las gráficas de los componentes
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(10, 12))

ax1.plot(PML_recent['precioMarginalLocal'], label='Componentes del PML')
ax1.legend(loc='upper left')
ax1.set_title('Precio Marginal Local Original')

ax2.plot(trend, label='Tendencia')
ax2.legend(loc='upper left')
ax2.set_title('Tendencia')

ax3.plot(seasonal, label='Estacionalidad')
ax3.legend(loc='upper left')
ax3.set_title('Estacionalidad')

ax4.plot(residual, label='Residuales')
ax4.legend(loc='upper left')
ax4.set_title('Residuales')

plt.tight_layout()  # Asegura que las gráficas no se superpongan
plt.show()
```

Figura 3.12. Comandos de creación de la serie y sus componentes temporales del PML.

Estos componentes de las series temporales nos ofrecen una visión más clara de los patrones subyacentes en los datos. Las gráficas representan los registros más recientes del PML en el nodo 07LPZ-115 durante marzo (720 horas del mes), con el eje "**x**" reflejando el tiempo y el eje "**y**" indicando los valores en pesos ($), variando según el componente representado. En la Figura 3.13, se ilustran los precios originales en pesos ($). Sin embargo, en la Figura 3.14, la unidad de pesos ($) se ajusta automáticamente por "*matplotlib*", abarcando $200.^{00}$ a $5,000.^{00}$ para la tendencia global, $-800.^{00}$ a $1,000.^{00}$ para la estacionalidad diaria, y finalmente, $-2,000.^{00}$ a $2,000.^{00}$ para los valores sin explicación por la estacionalidad y tendencia; estos últimos representan los aspectos atípicos o volátiles del PML.

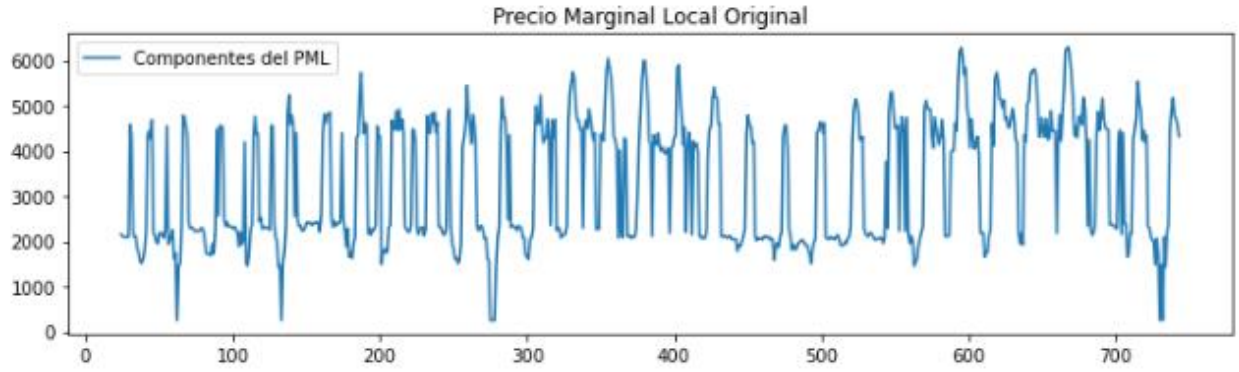


Figura 3.13 Serie de tiempo del PML

Entonces la gráfica muestra el comportamiento original de los datos del PML y podemos identificar una oscilación entre los $2,000.^{00}$ y los $6,000.^{00}$ pesos del PML del nodo de La Paz en el último mes (recordar que los 720 datos en el eje "x" corresponden al mes de marzo) con notable oscilación diaria subiendo y bajando entre este rango de precios. (ver Figura 3.13).

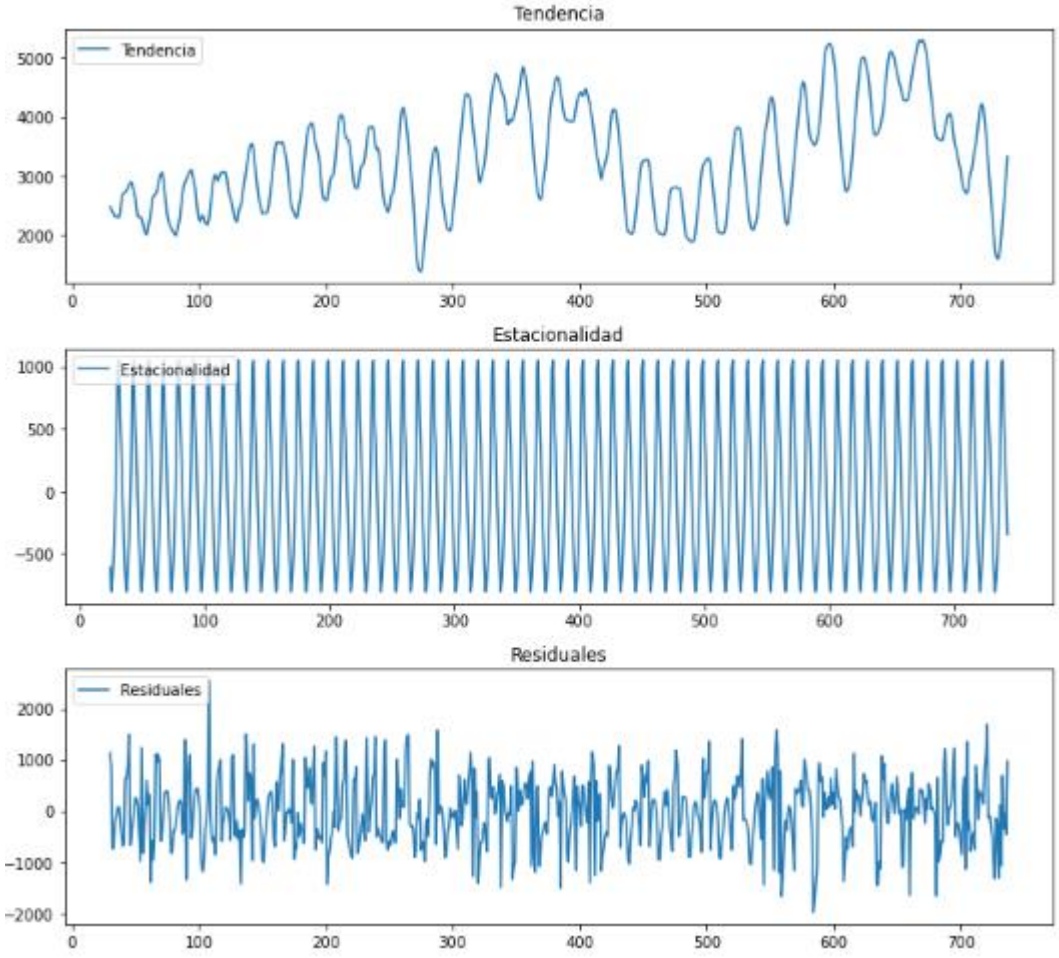


Figura 3.14 Componentes de la serie de tiempo; Tendencia, estacionalidad y residuales del nodo 07LPZ-115

La Figura 3.14 muestra la tendencia de la serie suavizando los patrones del precio real para notar solo la dirección hacia la que se mueve el PML notando mejor esa tendencia generalizada hacia la baja al final de la serie, la tendencia captura los cambios a largo plazo y oscila entre valores similares a la serie original (en este caso, entre $200.^{00}$ a $5,000.^{00}$ pesos). La estacionalidad: En una descomposición aditiva, muestra las fluctuaciones regulares alrededor de la tendencia. Estas fluctuaciones pueden ser positivas o negativas en relación con la tendencia. Por ejemplo, si durante ciertas horas del día el precio tiende a estar $100.^{00}$ pesos por encima de

la tendencia, se tendrá un valor estacional de +100 para esas horas. De manera similar, si durante ciertas horas el precio tiende a estar $100.^{00} pesos por debajo de la tendencia, se tendrá un valor estacional de -100 para esas horas. Por eso vemos valores oscilando entre $-800.^{00} y $1,000.^{00}, ya que capturan las fluctuaciones cíclicas en torno a la tendencia. Los residuales: los residuales representan lo que no ha sido capturado, ni por la tendencia, ni por la estacionalidad. Estos valores son los más volátiles ya que reflejan las irregularidades, eventos inusuales o el "ruido" de la serie. El rango de $-2,000.^{00} a $2,000.^{00} (los residuales de la serie) indica que después de eliminar la tendencia y la estacionalidad, todavía hay variaciones en el precio que pueden ser significativas, pero estas variaciones no siguen un patrón claro (tendencial o estacional). Véase Figura 3.14.

3.5.2.2 Estadísticos de tendencia central y de dispersión.

A continuación, se analizan sus estadísticos descriptivos de las variables, asi como sus valores de correlación aplicando código Python, que nos arroja los datos estadísticos de tendencia central, dispersión y correlación respectivamente. Para ello se usa el siguiente bloque de código que está preparando y visualizando un análisis de las columnas numéricas de la “dataframe” “*PML_recent*”, después de haber eliminado ciertas columnas no relevantes para el análisis.

El análisis y procesamiento de datos puede continuar con el siguiente comando que nos muestre de estos 720 datos sus estadísticos descriptivos: *mean_val = PML_recent[col].mean()* , *median_val = PML_recent[col].median(), std_dev = PML_recent[col].std(), skewness = skew(PML_recent[col]), kurt = kurtosis(PML_recent[col])* y mostrando los gráficos con un diagrama de caja usando *fig.add_trace(go.Box(y=PML_recent[col], name=col)), fig.show()*. Véase Figura 3.15:

```python
# Para cada columna numérica en PML_recent, calcula e imprime estadísticas, luego crea un boxplot
fig = go.Figure()

# Filtrar columnas numéricas
numerical_cols = PML_recent.select_dtypes(include=[np.number]).columns

# Itera a través de cada columna numérica.
for col in numerical_cols:
    # Calcula la media, mediana, desviación estándar, simetría y curtosis para la columna actual.
    mean_val = PML_recent[col].mean()
    median_val = PML_recent[col].median()
    std_dev = PML_recent[col].std()
    skewness = skew(PML_recent[col])
    kurt = kurtosis(PML_recent[col])

    # Imprime las estadísticas calculadas para la columna actual.
    print(f"Estadísticas para {col}:")
    print(f"Media: {mean_val:.2f}")
    print(f"Mediana: {median_val:.2f}")
    print(f"Desviación estándar: {std_dev:.2f}")
    print(f"Simetría: {skewness:.2f}")
    print(f"Curtosis: {kurt:.2f}")
    print("-----" * 5)

    # Añade un boxplot para la columna actual a la figura.
    fig.add_trace(go.Box(y=PML_recent[col], name=col))

    # Muestra la figura con todos los boxplots.
fig.show()
```

Figura 3.15 Comandos de obtención de estadísticos descriptivos del PML en un "*Boxplot*".

Para *"numerical_cols = PML_recent.select_dtypes(include =[np.number]).columns"* se busca filtrar todas las columnas que contienen datos numéricos en "PML_recent". Esto es útil para realizar análisis estadísticos específicos en solo las columnas numéricas. Entonces para iterar a través de cada columna numérica dentro del bucle "*for*", el código realiza las siguientes acciones para cada columna numérica en "*PML_recent*":

- Calcula estadísticas básicas como la media, mediana, desviación estándar, simetría (skewness) y curtosis.
- Imprime estas estadísticas para cada columna numérica. La simetría y la curtosis son medidas que indican la forma de la distribución de los datos (simetría respecto a la media y la "altitud" de la distribución, respectivamente).

Después el código generará un Boxplots, usando; "*fig.add_trace(go.Box(y= PML_recent[col], name=col))*": Añade un "boxplot" para cada columna numérica en una figura y con; "fig.show()": Muestra la Figura 3.16 con todos los boxplots. Los boxplots son útiles para visualizar la distribución de los datos, incluyendo la mediana, los cuartiles y los valores atípicos.

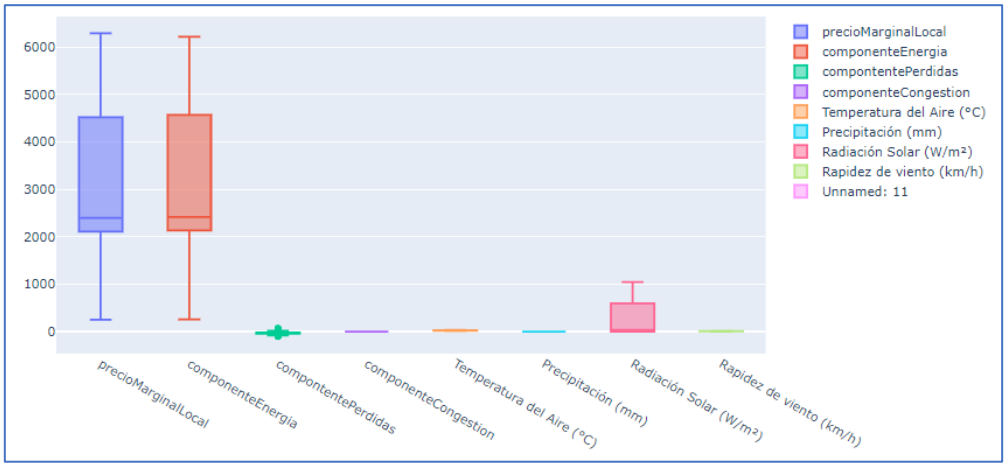


Figura 3.16 Gráfica de caja de estadísticos descriptivos de las variables del PML

Véase en la Figura 3.16 a las variables de nuestra "*dataset*" en un diagrama de cajas en donde si se superpusieran una encima de otra se reconocerá aquella variable similar al PML, la cual es el "componente de energía". Observemos que las demás son de magnitudes cercanas a cero, sin embargo, de aquellas variables destaca la radiación solar con valores de 0 hasta los 100 W/m^2.

```
------------------------
Estadísticas para precioMarginalLocal:
Media: 3281.02
Mediana: 2398.14
Desviación estándar: 1372.82
Simetría: 0.20
Curtosis: -1.35
------------------------
Estadísticas para componenteEnergia:
Media: 3312.82
Mediana: 2418.64
Desviación estándar: 1371.16
Simetría: 0.18
Curtosis: -1.37
------------------------
Estadísticas para componentePerdidas:
Media: -31.81
Mediana: -33.50
Desviación estándar: 23.35
Simetría: 0.69
Curtosis: 3.18
------------------------
Estadísticas para componenteCongestion:
Media: 0.00
Mediana: 0.00
Desviación estándar: 0.00
Simetría: 0.00
Curtosis: -3.00
------------------------
------------------------
Estadísticas para Temperatura del Aire (°C):
Media: 21.98
Mediana: 21.40
Desviación estándar: 5.48
Simetría: 0.18
Curtosis: -0.77
------------------------
Estadísticas para Precipitación (mm):
Media: 0.00
Mediana: 0.00
Desviación estándar: 0.00
Simetría: 0.00
Curtosis: -3.00
------------------------
Estadísticas para Radiación Solar (W/m²):
Media: 541.90
Mediana: 540.50
Desviación estándar: 317.24
Simetría: nan
Curtosis: nan
------------------------
Estadísticas para Rapidez de viento (km/h):
Media: 7.87
Mediana: 7.56
Desviación estándar: 4.49
Simetría: 0.24
Curtosis: -0.92
------------------------
```

Figura 3.17 Estadísticos descriptivos del nodo 07LPZ-115

A continuación, la información de la Figura 3.17
Precio marginal local:

- Media: El valor promedio del PML es de $3,281.[02]
- Mediana: La mediana, que representa el valor central de la distribución, es $2,398.[14]

- Desviación estándar: Indica la dispersión de los datos alrededor de la media; en este caso, es $1,372.82, lo que sugiere una variabilidad considerable.
- Simetría: La simetría es 0.20, lo que indica cierta asimetría positiva.
- Curtosis: La curtosis es -1.35, lo que sugiere una distribución relativamente plana en comparación con una distribución normal.

Componente de Energía:

- Las interpretaciones son similares a las de Precio Marginal Local.

Componente de Perdidas:

- Media negativa (-31.81), lo que sugiere que en promedio hay pérdidas.
- La distribución tiene una simetría de 0.69, indicando cierta asimetría positiva.
- La curtosis es 3.18, lo que sugiere una distribución más puntiaguda (colas más pesadas) en comparación con una distribución normal.

Componente de Congestión:

- La media y la mediana son $0.00, y la desviación estándar también es $0.00, lo que indica que los datos están constantemente en cero.
- La curtosis es -3.00, indicando colas extremadamente ligeras.

Temperatura del Aire (°C):

- La temperatura media es 23.01°C, con una desviación estándar de 4.73, lo que indica cierta variabilidad en las temperaturas.

Precipitación (mm):

- Todos los valores son cero, lo que sugiere que no hay precipitación en los datos proporcionados.

Radiación Solar (W/m²):

- La simetría y la curtosis son informadas como "nan" (no es un número), lo que podría deberse a la presencia de valores atípicos o a la faltantes en los datos.

Rapidez de viento (km/h):

- La velocidad promedio del viento es 9.10 km/h, con una desviación estándar de 3.76, indicando cierta variabilidad.

3.5.2.3 Matriz de correlación.

Los estadísticos describen las características de los datos y es conveniente hacer un análisis de la correlación de estos para medir la inferencia de cada variable con las demás y darnos una mejor comprensión de la base de datos de este nodo en específico, véase Figura 3.18:

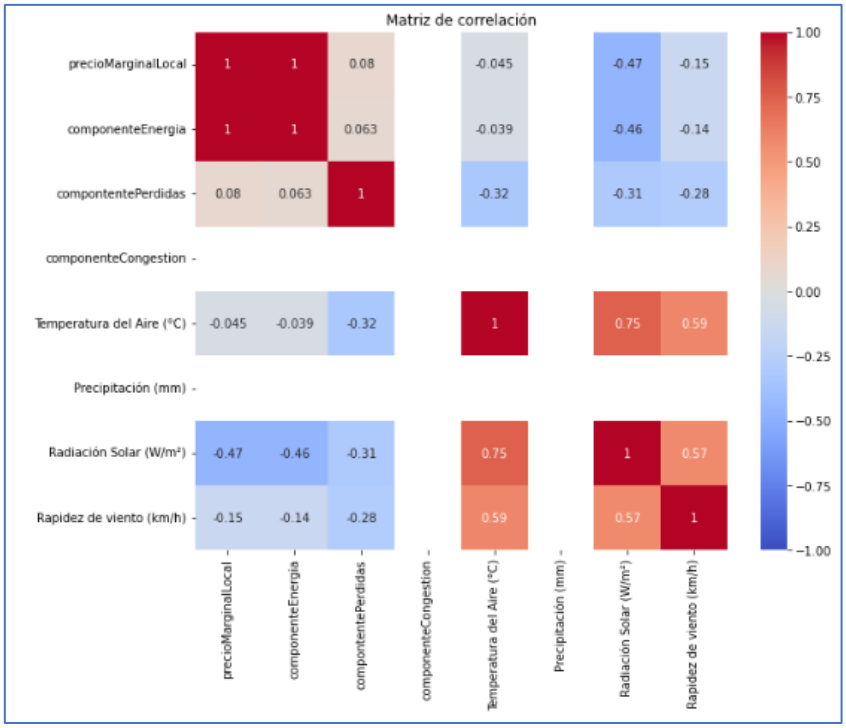


Figura 3.18 Diagrama de correlación múltiple de las variables del nodo 07LPZ-115

Como se observa en el diagrama de caja que la variable dependiente (PML) tiene una variable similar llamada “componente de energía” y ahora en el análisis de correlación, aparecen altamente correlacionadas. Veamos los resultados uno por uno de la Figura 3.18.

- ***Precio Marginal Local y componente de energía:*** Tienen una correlación extremadamente alta (0.999856). Esto significa que cuando uno de estos valores aumenta, el otro tiende a aumentar en una proporción similar y viceversa. Es posible que estén relacionados de manera causal al 100%.
- ***Precio Marginal Local y componente de perdidas:*** Tienen una correlación muy baja (0.079693). Esto sugiere que cuando el *precio Marginal Local* aumenta, las pérdidas también tienden a hacerlo, pero en muy poco.
- ***Componente de congestión:*** Su correlación con otras variables es inexistente debido a que tiene valores en cero. Sin embargo, es interesante notar que aparece con el nombre de

NaN (Not a Number) lo que en programación es un valor que no tiene un numero o que su resultado no es un número y por lo tanto se declara como NaN, sin embargo, también es porque el valor es cero en los datos y cálculos que su resultado al aplicar una correlación múltiple (como es en nuestro caso) resulta innecesario y también sin influencia en las demás variables. Nótese que también sucede con la variable de precipitación.

- ***Rapidez de viento (km/h):*** Tiene correlaciones inversamente proporcionales al PML con r = -0.147283 lo que sugiere que la velocidad del viento tiene poco impacto directo en el PML, pero en sentido inversamente proporcional.
- ***Temperatura del Aire (°C):*** Tiene correlación moderada y negativa con -0.044740 con respecto al precio marginal local, pero si tiene más correlación con respecto a la Radiación Solar (W/m²) (0.747658). Esto tiene sentido ya que, en días soleados, la temperatura tiende a ser más alta.
- ***Precipitación (mm):*** Similar al componente de congestión podemos ver que la precipitación tiene nula correlación con las variables en la matriz porque sus valores son todos ceros, dando una relación inherente con ninguna y reconociéndola como NaN.
- ***Radiación Solar (W/m²):*** Aparte de su relación con la temperatura del aire, tiene correlaciones bajas o negativas con las demás variables. La correlación negativa con *componente de energía* (-0.461390) y *precio Marginal Local* (-0.466080) podría sugerir que, en días de alta radiación solar, estos componentes tienden a disminuir ligeramente.

En resumen, las variables que más destacan en términos de correlaciones son el "*Precio Marginal Local"* y "*componente de energía"* debido a su alta correlación. El *"componente de pérdidas"* muestra una correlación débil de aproximadamente un 8%. Las demás correlaciones son bajas, inexistentes o hasta inversamente negativas, lo que sugiere que no tienen un impacto significativo entre sí o quizás que las relaciones podrían ser más complejas y no lineales. A excepción de la variable de radiación solar, al ser una variable con datos muy altos que alcanza los 1,045 W/m^2 podría influir fuertemente en el PML, pero que no necesariamente al subir la radiación solar sube al mismo tiempo el PML como aquí se ha determinado. Cabe destacar que la correlación no implica causalidad; es decir, aunque dos variables estén correlacionadas, una no necesariamente causa el cambio en la otra. Más bien eso se determina con la observación y experiencia de la naturaleza del precio marginal local. Es lógico que aparezcan NaN en la matriz dado que los estadísticos descriptivos arrogaban ceros en sus medidas de tendencia central y dispersión, y en la gráfica Boxplot (Figura 3.16.).

3.5.2.4 Sesgo y curtosis

Por último, el análisis de sesgo y curtosis pertenecientes al nodo 07LPZ-115, se observan los resultados en la siguiente Figura 3.19:

```
from scipy.stats import skew, kurtosis

data = {
    'Sesgo': PML_recent.apply(skew),
    'Curtosis': PML_recent.apply(kurtosis)
}
sesgo_curtosis_df = pd.DataFrame(data)
print(sesgo_curtosis_df)
```

```
                          Sesgo  Curtosis
precioMarginalLocal   -0.360725  0.208425
componenteEnergia     -0.407350  0.244890
compontentePerdidas    0.448298  0.616752
componenteCongestion   0.000000 -3.000000
```

Figura 3.19 Sesgo y curtosis del nodo 07LPZ-115

Sesgo (*Skewness*): Describe la asimetría de una distribución. Un sesgo positivo significa que la cola de la distribución se extiende más hacia la derecha, mientras que un sesgo negativo significa que se extiende más hacia la izquierda. En el contexto de los datos:

- **Precio Marginal Local:** Tiene un sesgo negativo (-0.360725), lo que indica que la distribución tiene una cola que se extiende ligeramente hacia la izquierda. Esto podría significar que hay más observaciones con precios marginales locales más bajos que altos.
- **Componente de energía:** Similar al anterior, tiene un sesgo negativo, lo que indica una cola que se extiende hacia la izquierda.
- **Componente de perdidas:** Con un sesgo positivo, indica que hay más observaciones con valores altos.
- **Componente de congestión:** Un sesgo de 0.0 significa que esta variable es simétrica, aunque ya sabemos que todos sus valores son ceros.

Curtosis (*Kurtosis*): Describe la "agudeza" o "achatamiento" de una distribución. Una curtosis positiva indica una distribución más puntiaguda que la distribución normal, mientras que una negativa indica una distribución más achatada. Una distribución normal tiene una curtosis de 0. Entonces en el contexto de los datos:

- **Precio Marginal Local** y **componente de energía**: Ambas variables tienen una curtosis ligeramente positiva, lo que sugiere que la distribución es un poco más puntiaguda que una distribución normal, con más valores cerca de la media y menos en las colas.
- **Componente de perdidas**: Tiene una curtosis mayor que las anteriores, esto implica que los eventos extremos pueden ser un poco más comunes que en una distribución normal.
- **Componente de congestión**: Una curtosis de -3 en realidad indica que todos los valores son iguales (en este caso, 0). Técnicamente, la curtosis para una constante es indefinida, pero algunos software o métodos la establecen en -3 por convención.

Estos estadísticos describen nuestras variables de esta manera: Al precio marginal local, tiene una ligera asimetría hacia la izquierda, con una distribución un poco más puntiaguda que una distribución normal, sugiere que la mayoría de los precios oscilan alrededor de un valor central y que hay menos variabilidad extrema en los precios.

Por otro lado, de las variables independientes; el componente de congestión es una constante 0, no aporta información al modelo. Las otras dos variables; el componente de energía muestra una tendencia a valores más bajos y una variabilidad de precios regular, mientras que el componente de pérdidas muestra una tendencia a valores más altos y una ligera propensión a experimentar eventos de precios extremos. Las demás variables climatológicas al tener faltantes en sus registros el código marca errores pues siempre deben existir valores numéricos para hacer los cálculos estadísticos.

3.5.3 Imputación de datos.

Lo correspondiente es realizar una imputación de los datos faltantes considerando la naturaleza de las variables. Véase Figura 3.20:

```python
# Importación del imputador KNN de la biblioteca scikit-learn.
from sklearn.impute import KNNImputer

# Imputación de valores faltantes en la columna 'Precipitación (mm)' con 0.0.
# Esto podría indicar que la ausencia de un valor se interpreta como ninguna precipitación.
PML['Precipitación (mm)'].fillna(0.0, inplace=True)

# Calcula la media de la columna 'Temperatura del Aire (°C)' para usarla en la imputación.
media_temp = PML['Temperatura del Aire (°C)'].mean()
# Imputación de valores faltantes en 'Temperatura del Aire (°C)' usando la media calculada.
PML['Temperatura del Aire (°C)'].fillna(media_temp, inplace=True)

# Calcula la mediana de la columna 'Rapidez de viento (km/h)' para usarla en la imputación.
mediana_viento = PML['Rapidez de viento (km/h)'].median()
# Imputación de valores faltantes en 'Rapidez de viento (km/h)' usando la mediana calculada.
PML['Rapidez de viento (km/h)'].fillna(mediana_viento, inplace=True)

# Creación de un objeto imputador KNN con 5 vecinos como parámetro.
imputer = KNNImputer(n_neighbors=5)
# Lista de columnas que serán imputadas usando KNN.
cols_to_impute = ['Temperatura del Aire (°C)', 'Precipitación (mm)', 'Rapidez de viento (km/h)', 'Radiación Solar (W/m²)']
# Aplicación de la imputación KNN a las columnas seleccionadas.
PML[cols_to_impute] = imputer.fit_transform(PML[cols_to_impute])

# Comando para verificar si aún quedan valores faltantes en el DataFrame.
print(PML.isnull().sum())
```

Figura 3.20 Imputación de datos faltantes en la base de datos del nodo 07LPZ-115

Precipitación (mm):

- Imputación realizada: Se reemplazaron los valores NaN con 0.0.
- Razón: Es probable que la ausencia de datos en la precipitación indique que no hubo precipitación durante ese periodo. Esto es una suposición común en series temporales meteorológicas.

Temperatura del Aire (°C):

- Imputación realizada: Se reemplazaron los valores NaN con la media de la temperatura del aire.
- Razón: Se menciona que, en una localidad específica, la temperatura del aire tiende a rondar un promedio con algunas variaciones estacionales y diurnas. Por lo tanto, reemplazar con la media puede ser una aproximación razonable.

Rapidez de viento (km/h):

- Imputación realizada: Se reemplazaron los valores NaN con la mediana de la rapidez del viento.

- Razón: se consideró a la mediana como una medida robusta que puede ser menos sensible a valores extremos que el promedio. Si hay variaciones extremas en la rapidez del viento, la mediana podría ser una elección más representativa.

Radiación Solar (W/m²):

- Imputación realizada: Durante las horas de noche (específicamente entre las 19:00 y las 06:00), se reemplazaron los valores NaN en Radiación Solar con 0.0.
- Razón: Durante las horas de noche, no hay radiación solar, por lo que cualquier valor faltante en este periodo debe corresponder a un valor de 0.

Imputación usando KNNImputer:

- Imputación realizada: Para cualquier otra columna numérica con valores faltantes, se utiliza el método de imputación KNN (K-Nearest Neighbors). En este caso, se ha elegido 5 como el número de vecinos más cercanos para imputar el valor faltante.
- Razón: El KNNImputer considera el valor de los 'k' vecinos más cercanos para imputar un valor faltante. Es un enfoque más sofisticado y puede ser efectivo cuando la estructura de datos tiene alguna correlación entre características. Se ha optado por este método para tratar de capturar relaciones más complejas en los datos que no se pueden manejar simplemente mediante imputaciones basadas en estadísticas descriptivas.

Finalmente, se verifica si aún hay valores faltantes en el conjunto de datos, lo que permitiría validar la efectividad del proceso de imputación o identificar columnas que requieran un tratamiento adicional en la Figura 3.21.

```
# Comando para verificar si aún quedan valores faltantes en el DataFrame.
print(PML_recent.isnull().sum())
```

```
precioMarginalLocal          0
componenteEnergia            0
compontentePerdidas          0
componenteCongestion         0
Temperatura del Aire (°C)    0
Precipitación (mm)           0
Radiación Solar (W/m²)       0
Rapidez de viento (km/h)     0
dtype: int64
```

Figura 3.21 Dataset obtenida después del preprocesamiento de datos del nodo 07LPZ-115

3.6 Construcción del modelo:

En esta etapa de la metodología, se deciden los algoritmos de pronósticos; se presentan 5 de vanguardia y enfocados al aprendizaje de los datos con las bibliotecas de Python y que estructuren los datos a las características del Machine Learning.

Aunque tradicionalmente la estadística descriptiva puede revelar las generalidades que describen a los datos, la mejor forma de poder proponer un modelo de pronóstico es con las bondades de las redes neuronales y el aprendizaje de máquina. Los algoritmos son los siguientes:

3.6.1 Árbol de decisión para regresión (Decision Tree Regressor):

Este modelo se puede implementar utilizando la biblioteca *Scikit-learn* de Python. Los árboles de decisión son útiles por su facilidad de interpretación y su capacidad para manejar relaciones no lineales. Se ajustan a los datos de entrenamiento, aprendiendo a predecir los valores de PML basándose en las variables de entrada. Luego, se valida y ajusta para mejorar su precisión y evitar el sobreajuste.

El código selecciona e identifica las variables asi: *"x = PML [PML.columns.drop ("precioMarginalLocal")]"* para seleccionar todas las columnas del DataFrame 'PML' excepto 'Precio Marginal Local', que se utilizarán como características (features) para el modelo. Las características son los inputs que el modelo usará para aprender a hacer predicciones. "*y = PML["precioMarginalLocal"]*": Aquí se selecciona la columna 'Precio Marginal Local' como la variable objetivo (target). Esta es la variable que el modelo intentará predecir, en este caso, el precio marginal local, véase Figura 3.22.

```
# Selecciona todas las columnas del DataFrame 'PML' excepto 'precioMarginalLocal' para las características (features).
x = PML[PML.columns.drop("precioMarginalLocal")]

# Selecciona la columna 'precioMarginalLocal' del DataFrame 'PML' para la variable objetivo (target).
y = PML["precioMarginalLocal"]


# Calcular índices para los cortes
corte_train = int(len(x) * 0.8)
corte_valid = corte_train + int(len(x) * 0.1)

# Dividir datos en entrenamiento, validación y prueba
train_x, train_y = x.iloc[:corte_train], y.iloc[:corte_train]
valid_x, valid_y = x.iloc[corte_train:corte_valid], y.iloc[corte_train:corte_valid]
test_x, test_y = x.iloc[corte_valid:], y.iloc[corte_valid:]
```

Figura 3.22 Comandos para identificar la variables y subconjuntos en el data set.

Después se hace la división de datos en Entrenamiento, Validación y Prueba: Primero, se calculan los índices para realizar los cortes entre los conjuntos de entrenamiento, validación y prueba. Se decide asignar el 80% de los datos para entrenamiento (*corte_train*), el 10% para validación (*corte_valid*), y el 10% restante para prueba. Luego, se realiza la división efectiva de los datos: "*train_x, train_y*": Conjunto de entrenamiento que el modelo usará para aprender. Contiene el 80% de los datos. "*valid_x, valid_y*": Conjunto de validación utilizado para ajustar hiperparámetros y realizar evaluaciones intermedias del modelo. Contiene el 10% de los datos después del conjunto de entrenamiento. "*test_x, test_y*": Conjunto de prueba para la evaluación final del modelo. Este conjunto no se utiliza durante el entrenamiento y sirve para evaluar cómo se desempeñaría el modelo en datos no vistos previamente. También comprende el 10% restante de los datos. véase Figura 3.22

La cardinalidad de cada subconjunto se visualiza en la Figura 3.23; en este caso, se divide el conjunto de datos en un conjunto de entrenamiento con 504 muestras, un conjunto de validación con 108 muestras y un conjunto de prueba con 108 muestras (504+108+108=720) y al número 13 en cada conjunto correspondiente a las 13 características (variables).

```
# Visualizar las dimensiones de los conjuntos
print("Dimensiones de entrenamiento:", train_data.shape)
print("Dimensiones de validación:", validation_data.shape)
print("Dimensiones de prueba:", test_data.shape)
```

```
Dimensiones de entrenamiento: (504, 13)
Dimensiones de validación: (108, 13)
Dimensiones de prueba: (108, 13)
```

Figura 3.23 Cardinalidad de los subconjuntos; entrenamiento, validación y prueba.

```
# Entrenar el modelo de árbol de decisión
tree_model = DecisionTreeRegressor(random_state=44)
tree_model.fit(train_x, train_y)

# Hacer predicciones para las próximas 48 observaciones
x_future_tree = test_x.iloc[-48:]
predictions_future_tree = tree_model.predict(x_future_tree)

# Obtener las fechas para las últimas 48 observaciones y las "predicciones futuras"
dates_recent_48 = dataset_original["fechaHoraMDA"].iloc[-48:]
dates_future = pd.date_range(start=dates_recent_48.iloc[-1], periods=49, freq='H')[1:]  # Genera 48 fechas futuras

# Calcular los promedios de las próximas 48 horas
future_hours = 48
hourly_averages_tree = []

for i in range(future_hours):
    if i + 24 < len(predictions_future_tree):
    # Calcular el promedio de las predicciones del árbol para la hora actual y la misma hora del día anterior
        average = (predictions_future_tree[i] + predictions_future_tree[i + 24]) / 2
        hourly_averages_tree.append(average)
    else:
    # Si no hay suficientes datos para el promedio, usa NaN
        hourly_averages_tree.append(np.nan)

# Crear una figura para el gráfico
plt.figure(figsize=(12, 6))
```

Figura 3.24 Estructura del modelo de árbol de decisión para regresión.

En la Figura 3.24 se proponen los siguientes comandos:

- Preparación de Datos para la Visualización: Se obtienen las fechas correspondientes a las últimas 48 observaciones y se generan 48 fechas futuras. Esto se utiliza para etiquetar los ejes en el gráfico.
- Cálculo de Promedios Horarios: Se calculan promedios para las próximas 48 horas. Si hay datos disponibles para la hora actual y la misma hora del día anterior, se calcula el promedio de ambos; de lo contrario, se usa NaN. Esto añade un elemento de análisis temporal al modelo.
- Creación del Gráfico: Se usa Matplotlib para crear un gráfico que compara los valores reales de las últimas 48 horas "*(test_y.iloc[-48:])"* con las predicciones del árbol de decisión para las próximas 48 horas *"(predictions_future_tree)"* y los promedios horarios calculados.

3.6.2 Regresión polinomial de segundo grado

También usando *Scikit-learn*, este modelo extiende la regresión lineal permitiendo capturar relaciones no lineales entre las variables. Es particularmente útil cuando los patrones de datos sugieren una tendencia cuadrática. Al igual que con los árboles de decisión, se entrena, valida y prueba para asegurar la generalización del modelo.

```
# Preparar los datos; selección de la variable de interés
x = PML[PML.columns.drop("precioMarginalLocal")]
y = PML["precioMarginalLocal"]

# Calcular índices para los cortes
corte_train = int(len(x) * 0.8)
corte_valid = corte_train + int(len(x) * 0.1)

# Dividir datos en entrenamiento, validación y prueba
train_x, train_y = x.iloc[:corte_train], y.iloc[:corte_train]
valid_x, valid_y = x.iloc[corte_train:corte_valid], y.iloc[corte_train:corte_valid]
test_x, test_y = x.iloc[corte_valid:], y.iloc[corte_valid:]

# Transformar características para regresión polinomial
transformer = PolynomialFeatures(degree=2, include_bias=False)
x_poly = transformer.fit_transform(train_x)
test_x_poly = transformer.transform(test_x)
```

Figura 3.25 Comandos de identificación de las variables y subconjuntos del modelo de regresión polinominal

En la Figura 3.25 se observa los comandos iniciales del código para regresión polinominal, y similarmente al código de árbol de decision, aquí tambien se identifican las variables dependientes y al PML se le identifica como el objetivo. Tenemos: *"x = PML [PML.columns.drop ("precioMarginalLocal")]":* Selecciona todas las columnas del DataFrame PML excepto precioMarginalLocal para utilizarlas como características (*features*). *"y = PML ["precioMarginalLocal"]":* Selecciona la columna Precio Marginal Local del DataFrame PML como la variable objetivo (*target*).

Después se hace la división del Conjunto de Datos: Se calculan índices para dividir el conjunto de datos en entrenamiento, validación y prueba, basándose en porcentajes (80% para entrenamiento, 10% para validación, y 10% para prueba). "*train_x, train_y*": Conjuntos de entrenamiento. *"valid_x, valid_y"*: Conjuntos de validación, *"test_x, test_y"*: Conjuntos de prueba.

Aquí es necesario diferenciar con el algoritmo de árboles puesto que se recurre a una transformación de Características para Regresión Polinomial: *"transformer = PolynomialFeatures (degree=2, include_bias=False)":* Se crea una instancia de "*PolynomialFeatures*" para transformar las características. Se especifica un grado polinomial de 2, lo que implica que las características se elevarán al cuadrado y se crearán interacciones entre ellas. *"include_bias=False",* indica que no se añadirá un término de sesgo (coeficiente igual a 1, manteniendo solo las características que aportan información única al modelo).

Después *"x_poly = transformer.fit_transform(train_x)":* Transforma el conjunto de entrenamiento train_x aplicando la transformación polinomial. Esto implica elevar las características al cuadrado y crear interacciones entre ellas. *"test_x_poly = transformer.transform (test_x)"*: Transforma el conjunto de prueba (*test_x*) utilizando la misma transformación aplicada al conjunto de entrenamiento.

```
# Implementación de TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=10)

# Aquí transformamos toda la matriz X, ya que TimeSeriesSplit dividirá por nosotros
transformer = PolynomialFeatures(degree=2, include_bias=False)
x_poly_all = transformer.fit_transform(x)

# Entrenar el modelo
model = LinearRegression().fit(x_poly, train_y)

# Hacer predicciones para las próximas 48 observaciones
x_future_poly = test_x_poly[-48:]
predictions_future_poly = model.predict(x_future_poly)

# Calcular los promedios de las próximas 48 horas
future_hours = 48
hourly_averages_poly = []

for i in range(future_hours):
    if i + 24 < len(predictions_future_poly):
        # Calcular el promedio de las predicciones para la hora actual y la misma hora del día siguiente
        average = (predictions_future_poly[i] + predictions_future_poly[i + 24]) / 2
        hourly_averages_poly.append(average)
    else:
        # Si no hay suficientes datos para el promedio, usa NaN
        hourly_averages_poly.append(np.nan)
```

Figura 3.26 Estructura del modelo de Regresión Polinominal.

De acuerdo con la Figura 3.26, observamos la implementación de "*TimeSeriesSplit*": *"TimeSeriesSplit (n_splits=10)":* Se crea un objeto de "TimeSeriesSplit" para una validación cruzada específica de series temporales. Aquí, los datos se dividen en 10 partes consecutivas. Este método es particularmente útil para series temporales, ya que respeta el orden temporal de los datos.

Dado que ya se hizo la transformación de características, tenemos *"transformer = PolynomialFeatures(degree=2, include_bias=False)":* Se crea un transformador para expandir las características originales a términos polinómicos de grado 2, excluyendo el término de sesgo. Después en *"x_poly_all = transformer.fit_transform(x)":* Transforma todas las características "x" en características polinómicas. Continuando con la Figura 3.36 se cubre tambien el

entrenamiento del modelo: *"model = LinearRegression().fit(x_poly, train_y)":* Se entrena un modelo de regresión lineal utilizando solo el conjunto de entrenamiento transformado.

Se seleccionan los últimos 48 datos del conjunto de prueba transformado *"(test_x_poly[-48:])"* y realiza predicciones para estas observaciones. Después se calculan los promedios para cada hora usando la predicción actual y la correspondiente 24 horas después, para un total de 48 horas. Si no hay suficientes datos para calcular el promedio (es decir, no hay datos para "la misma hora del día siguiente"), se utiliza NaN.

Para poder graficar el pronóstico del modelo de regresión polinominal, se hace el siguiente bloque de código y se visualizan los valores:

```
# Obtener fechas para las últimas 48 horas y las próximas 48 horas
dates_current_poly = dataset_original["fechaHoraMDA"].iloc[-48:]
dates_future_poly = pd.date_range(start=dates_current_poly.iloc[-1], periods=future_hours, freq='H')

# Crear una figura para el gráfico
plt.figure(figsize=(12, 6))

# Graficar los valores reales de las últimas 48 horas
plt.plot(dates_current_poly, test_y.iloc[-48:], marker='.', label="Valores Reales de las Últimas 48 Horas", color='blue')

# Graficar las predicciones para las próximas 48 horas
plt.plot(dates_future_poly, predictions_future_poly, marker='o', label="Predicciones para las Próximas 48 Horas", color='red',

# Graficar los promedios de las próximas 48 horas
plt.plot(dates_future_poly, hourly_averages_poly, marker='x', label="Promedios para las Próximas 48 Horas", color='green',

# Configurar título, etiquetas y leyenda
plt.title("Valores Reales vs Predicciones vs Promedios (Últimas 48 Horas y 48 Horas a Futuro)")
plt.xlabel("Fecha y Hora")
plt.ylabel("Precio Marginal Local")
plt.legend()

# Rotar las fechas para una mejor visualización
plt.xticks(rotation=45)

# Mostrar el gráfico
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figura 3.27 Comando para graficar los pronósticos del modelo de Regresión Polinominal

En la Figura 3.27 está el código que genera un gráfico para visualizar y comparar los valores reales del "Precio Marginal Local" de las últimas 48 horas con las predicciones y los promedios calculados para las próximas 48 horas, basados en un modelo de regresión lineal con características polinómicas.

A continuación, se describe cada parte del código: *"dates_current_poly:"* Recoge las fechas y horas de las últimas 48 observaciones del conjunto de datos original.

"dates_future_poly:" Genera un rango de fechas y horas para las próximas 48 horas, comenzando desde la última fecha y hora de "*dates_current_poly*".

En cuanto a la configuración del Gráfico se tiene: *"plt.figure(figsize=(12, 6))"*: Crea una figura para el gráfico con un tamaño específico. Se grafican los valores reales del Precio Marginal Local para las últimas 48 horas. Se grafican las predicciones del modelo para las próximas 48 horas.
Se grafican los promedios calculados para las próximas 48 horas, basados en las predicciones.

Este gráfico proporciona una representación visual clara de cómo las predicciones y los promedios calculados para el futuro se comparan con los valores reales recientes, lo que es útil para evaluar la precisión y utilidad del modelo de regresión en la predicción de precios marginales locales.

3.6.3 SARIMA:

El componente "*Seasonal*" del SARIMA permite que el modelo tenga en cuenta la estacionalidad en los datos (por ejemplo, patrones que se repiten cada 24 horas, cada 7 días, cada año, etc.). Es importante señalar que para SARIMA, los parámetros (p, d, q, P, D, Q, s) son cruciales y pueden requerir una identificación adecuada, y a veces la optimización iterativa. Es ideal para datos del PML que muestran patrones estacionales y dependencias en valores pasados. Requiere una cuidadosa selección de parámetros y validación para asegurar su efectividad.

Es por práctica que se recurre a crear copias de las bases de datos por lo que en este algoritmo la *data frame* se llamará "PML1DE", véase Figura 3.28.

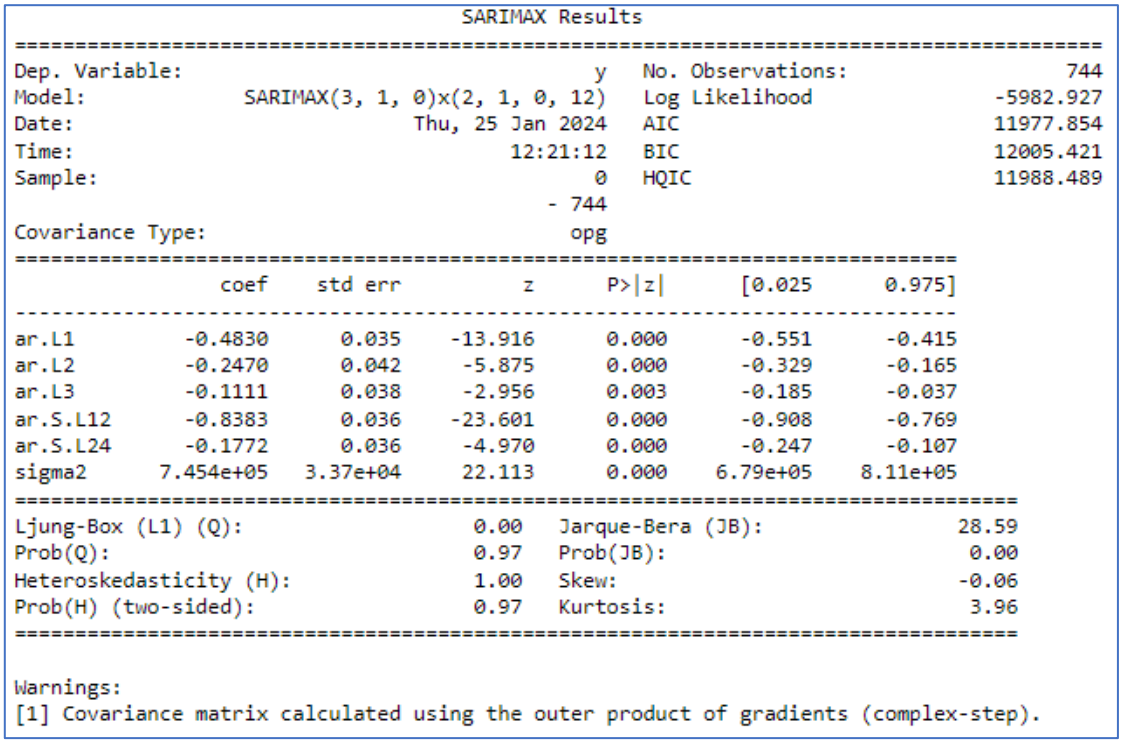
```
# Modelo Auto-Arima
modelo_auto = auto_arima(PML1DE, start_p=0, start_q=0,
                         max_p=3, max_q=3, m=12,
                         start_P=0, seasonal=True,
                         d=1, D=1, trace=True,
                         error_action='ignore',  # no deseo saber si el orden no funciona
                         suppress_warnings=True,  # no deseo ver las advertencias
                         stepwise=True)  # configurar para que sea paso a paso

print(modelo_auto.summary())
```

Figura 3.28 Comandos para la ejecución de Auto-ARIMA

Auto-ARIMA automatiza el proceso de selección de parámetros, probando una gama de diferentes combinaciones y seleccionando el modelo con el mejor ajuste según un criterio predefinido, generalmente el AIC (Criterio de Información de Akaike). Esto no solo ahorra tiempo y recursos, sino que también proporciona una forma objetiva y estadísticamente sólida de seleccionar el mejor modelo para los datos. De acuerdo con el código mostrado en la Figura 3.28, tenemos: *"start_p=0, start_q=0"*: estos son los puntos de partida para el orden AR (p) y el orden MA (q) del componente no estacional del modelo, entonces *"max_p=3, max_q=3"*: son los máximos valores que auto_arima considerará para los órdenes AR y MA no estacionales.

Después *"m=12"*: es el número de periodos en cada temporada. Sugiere que los datos son estacionales cada 12 horas. *"start_P=0, seasonal=True"*: Estos parámetros indican que el modelo debe considerar la estacionalidad y comienza a buscar desde el orden 0 para el componente AR estacional. Seguidamente tenemos *"d=1, D=1"*: estos son los órdenes de diferenciación no estacional y estacional, respectivamente. Especifican cuántas veces los datos deben ser diferenciados para hacerlos estacionarios. Y con "trace=True": imprimirá el progreso de la función mientras ajusta diferentes modelos. Opcionalmente se agrega *"error_action='ignore', suppress_warnings=True"*: para controlar el manejo de errores y advertencias. Aquí, se ignoran los errores en los ajustes de los modelos y se suprimen las advertencias por estética.

Es importante *"stepwise=True"*: ya que esto activa un algoritmo eficiente para buscar el mejor modelo, probando una variedad de combinaciones de parámetros de manera más inteligente en lugar de probar todas las combinaciones posibles. Se imprime el resumen del mejor modelo encontrado donde incluirá los coeficientes del modelo, el valor de AIC (Criterio de Información de Akaike) y otras estadísticas que ayudan a entender el rendimiento y la idoneidad del modelo. Ver Figura 3.29



```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                 y   No. Observations:                  744
Model:             SARIMAX(3, 1, 0)x(2, 1, 0, 12)   Log Likelihood               -5982.927
Date:                            Thu, 25 Jan 2024   AIC                          11977.854
Time:                                    12:21:12   BIC                          12005.421
Sample:                                         0   HQIC                         11988.489
                                              - 744
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4830      0.035    -13.916      0.000      -0.551      -0.415
ar.L2         -0.2470      0.042     -5.875      0.000      -0.329      -0.165
ar.L3         -0.1111      0.038     -2.956      0.003      -0.185      -0.037
ar.S.L12      -0.8383      0.036    -23.601      0.000      -0.908      -0.769
ar.S.L24      -0.1772      0.036     -4.970      0.000      -0.247      -0.107
sigma2      7.454e+05   3.37e+04     22.113      0.000    6.79e+05    8.11e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                28.59
Prob(Q):                              0.97   Prob(JB):                         0.00
Heteroskedasticity (H):               1.00   Skew:                            -0.06
Prob(H) (two-sided):                  0.97   Kurtosis:                         3.96
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Figura 3.29 Resultados del mejor modelo determinado por Auto-ARIMA

Concluido el código de auto ARIMA, se obtuvo la información de la Figura 3.29, donde el "Mejor modelo" identificado es SARIMA (3,1,0) (2,1,0) [12]. Esto indica:

ARIMA (3,1,0):

- AR (Autoregressive) = 3: El modelo utiliza tres términos auto regresivos.
- I (Integrated) = 1: Hay una diferenciación para hacer la serie temporal estacionaria.
- MA (Moving Average) = 0: No se utilizan términos de media móvil.

Componente estacional (2,1,0) [12]:

- AR estacional = 2: Hay dos términos auto regresivos en la parte estacional del modelo.
- Diferenciación estacional = 1: Se realiza una diferenciación estacional.
- MA estacional = 0: No se utilizan términos de media móvil en la parte estacional.
- [12]: Esto indica que el patrón estacional ocurre cada 12 horas.

El código incluye el cálculo de estadísticas del modelo SARIMA:

- *"Log Likelihood":* -5982.927 indica la verosimilitud logarítmica del modelo. Significa que el mejor modelo tiene esa similitud a los datos reales y en el momento en que se tenga algún otro modelo para compararlos, esta magnitud sirve para saber que el más alto es el más capaz de asemejarse a los datos del PML.
- AIC (Criterio de Información de Akaike): 11977.854 es un estimador de la calidad relativa de los modelos estadísticos; cuanto menor sea el AIC, mejor. BIC (Criterio de Información Bayesiano): 12005.421 es otro criterio para la selección del modelo, similar al AIC, pero con una penalización más fuerte por modelos con más parámetros.
- HQIC (Criterio de Información de Hannan-Quinn): 11988.489 es otro criterio para la selección de modelos, que está entre AIC y BIC en términos de penalización por complejidad.

Los coeficientes (ar.L1, ar.L2, ar.L3, ar.S.L12, ar.S.L24) y sus errores estándar, valores z, y p-valores indican la importancia de cada término en el modelo. Los coeficientes con p-valores bajos son estadísticamente significativos.

Pruebas y Diagnósticos:

- Ljung-Box Test (Q): Prueba la autocorrelación residual. Un p-valor alto (como 0.97) sugiere que no hay autocorrelación significativa en los residuos, lo que es bueno.
- Jarque-Bera Test (JB): Prueba la normalidad de los residuos. Un p-valor bajo indica que los residuos no siguen una distribución normal.
- Heteroskedasticity Test (H): Evalúa la homogeneidad de la varianza de los residuos. Un p-valor alto sugiere que la varianza es constante (homocedasticidad).
- Skew y Kurtosis: Miden la asimetría y la forma "puntiaguda" de la distribución de los residuos, respectivamente.

Se concluyó que el modelo SARIMA (3,1,0) (2,1,0) [12] resultó ser el mejor ajuste a la serie de datos, dada la baja AIC y las pruebas estadísticas satisfactorias. Sin embargo, siempre es importante validar estas conclusiones con una comprensión del contexto de los datos y realizar una validación cruzada o pruebas para confirmar el rendimiento del modelo. Ver Figura 3.30

```python
# Dividir el dataset en entrenamiento y prueba
train_size = int(PML1DE.shape[0] * 0.8)
train_data = PML1DE.iloc[:train_size]
test_data = PML1DE.iloc[train_size:]

# Entrenar el modelo en los datos de entrenamiento
arima_model = SARIMAX(train_data, order=modelo_auto.order, seasonal_order=modelo_auto.seasonal_order)
arima_result = arima_model.fit()

# predicciones en los datos de prueba
arima_pred = arima_result.predict(start=train_size, end=train_size + len(test_data) - 1, typ="levels").rename("ARIMA Predictions"

# Mostrar las predicciones con numeración
print("Predicciones ARIMA:")
for idx, pred in enumerate(arima_pred, 1):
    print(f"{idx}. {pred}")
print("\n------------------------\n")

# Función para calcular el MAPE
def calculate_mape(true_values, predictions):
    # Filtrar ceros en los valores verdaderos para evitar divisiones por cero
    mask = true_values != 0
    return (np.fabs((true_values[mask] - predictions[mask]) / true_values[mask])).mean() * 100

# Evaluación del modelo
evaluate_model(test_data, arima_pred)

# Calcular y mostrar el MAPE
mape = calculate_mape(test_data.values.flatten(), arima_pred.values)
print(f"MAPE: {mape}")
```

Figura 3.30 Estructura del modelo de SARIMA

El código que se observa en la Figura 3.30 realiza varios pasos importantes para ajustar un modelo SARIMA y evaluar su rendimiento:

1. **Dividir el Dataset en Entrenamiento y Prueba:** Se calcula el tamaño del conjunto de entrenamiento como el 80% del total de los datos. Los datos se dividen en dos conjuntos: uno para entrenar el modelo (*train_data*) y otro para probarlo (*test_data*).
2. **Entrenar el Modelo SARIMA:** Se crea un modelo SARIMA usando la librería *statsmodels* (por default asume que es SARIMAX importado de *"statsmodels .tsa.statespace.sarimax"*). El modelo se ajusta con los datos de entrenamiento. Los órdenes del modelo (tanto para la parte ARIMA como para la parte estacional) se toman del modelo auto_arima previamente ajustado; (3,1,0) (2,1,0) [12].
3. **Realizar Predicciones en los Datos de Prueba:** Se hacen predicciones para el período correspondiente al conjunto de prueba. Estas predicciones se renombran como *"ARIMA Predictions"* para su fácil identificación.
4. **Mostrar las Predicciones:** Se imprime cada predicción con su correspondiente número de orden. En el capítulo de resultados se podrán apreciar los gráficos correspondientes.
5. **Función para Calcular el MAPE (Mean Absolute Percentage Error):** Se define una función para calcular el MAPE, que es una medida común para evaluar la precisión de las

predicciones en modelos de series temporales. Esta función evita la división por cero al excluir los valores verdaderos que son cero.

6. **Evaluación del Modelo:** Se evalúa el modelo comparando las predicciones con los datos reales de prueba. Finalmente se calcula el MAPE utilizando la función definida anteriormente y se imprime su valor.

3.6.4 LSTM:

Las Redes Neuronales de Memoria a Corto y Largo Plazo (LSTM) son ideales para capturar dependencias a largo plazo en series temporales complejas. Implementadas mediante bibliotecas como "*Keras"* o "*TensorFlow"*, estas redes son adecuadas para pronósticos precisos de PML.

```
# Definir el número de time steps
time_steps = 48

# Iniciar TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=3)
history = []  # Para almacenar la historia de entrenamiento de cada división

# Función para crear un dataset a partir de los datos
def create_dataset(X, y, time_steps=1):
    Xs, ys = [], []
    for i in range(len(X) - time_steps + 1):
        v = X.iloc[i:(i+time_steps), 1:].values  # Excluir la columna de fechas
        Xs.append(v)
        ys.append(y.iloc[i + time_steps - 1])
    return np.array(Xs), np.array(ys)

# Construir y entrenar el modelo para cada división en TimeSeriesSplit
for train_index, val_index in tscv.split(train_data):
    train, val = train_data.iloc[train_index], train_data.iloc[val_index]

    # Aplicar la función de creación de dataset
    X_train, y_train = create_dataset(train, train['precioMarginalLocal'], time_steps)
    X_val, y_val = create_dataset(val, val['precioMarginalLocal'], time_steps)

    # Definir y compilar la arquitectura del modelo
    model = tf.keras.models.Sequential([
        tf.keras.layers.LSTM(units=128, input_shape=(time_steps, X_train.shape[2])),
        tf.keras.layers.Dense(units=1)
    ])
    model.compile(loss='mean_squared_error', optimizer=tf.keras.optimizers.Adam(learning_rate=0.011))

    # Entrenar el modelo y almacenar la historia
    hist = model.fit(X_train, y_train, epochs=50, batch_size=48, validation_data=(X_val, y_val), verbose=1)
    history.append(hist)
```

Figura 3.31 Estructura del modelo LSTM

La arquitectura es una red neuronal con una capa LSTM estándar seguida de una capa densa, véase Figura 3.31. Los detalles de esta arquitectura son:

- **Definición de Time Steps:** *"time_steps = 48"* indica que se utilizarán 48 pasos temporales en cada muestra de entrada para el modelo. Esto significa que cada muestra de entrada tendrá datos de 48 puntos temporales anteriores.
- **Inicialización de TimeSeriesSplit:** *"tscv = TimeSeriesSplit(n_splits=3)"* crea un objeto "*TimeSeriesSplit*" para la validación cruzada en series temporales con 3 divisiones. Esto se utiliza para entrenar y validar el modelo en diferentes segmentos de los datos de entrenamiento.
- **Función para Crear un Dataset:** *"create_dataset"* es una función para transformar los datos en un formato adecuado para entrenar modelos de redes neuronales. Esta función toma una serie temporal y la convierte en un conjunto de muestras, donde cada muestra contiene *"time_steps"* observaciones pasadas para predecir el siguiente valor.
- **Construcción y Entrenamiento del Modelo para Cada División en TimeSeriesSplit:** El bucle "for" itera sobre cada división proporcionada por "*TimeSeriesSplit*". Para cada división, se divide el conjunto de entrenamiento en un subconjunto de entrenamiento y validación. Se aplica la función "*create_dataset*" a estos subconjuntos para obtener los datos en el formato requerido. Se define un modelo secuencial utilizando Keras, con una capa LSTM seguida de una capa densa. El modelo se compila con una función de pérdida de error cuadrático medio y un optimizador "*Adam*". Finalmente, se entrena el modelo utilizando los datos de entrenamiento y validación, y se almacena el historial de entrenamiento.
- **Arquitectura del Modelo:** La capa LSTM con 128 unidades es adecuada para capturar dependencias temporales en los datos. La capa densa al final se utiliza para la predicción de un valor (en este caso, el *"precioMarginalLocal*"). La tasa de aprendizaje del optimizador Adam se establece en 0.011.
- **Entrenamiento del Modelo:** El modelo se entrena por 50 épocas con un tamaño de lote de 48. Se utiliza la validación cruzada para evaluar el rendimiento del modelo en diferentes segmentos del conjunto de entrenamiento.

3.6.5 LSTM Secuencial + Prophet:

Esta combinación une la capacidad de las LSTM para manejar dependencias temporales complejas con la robustez de Prophet para manejar tendencias y estacionalidades en los datos.

Prophet, una herramienta de Facebook es excelente para datos con patrones estacionales fuertes y varios puntos de cambio. Al combinarlo con LSTM, se puede lograr un modelo altamente efectivo y adaptable para pronósticos de series temporales.

```python
def create_dataset_seq2seq(X, y, time_steps=48, future_steps=48):
    Xs, ys = [], []
    for i in range(len(X) - time_steps - future_steps + 1):
        v = X.iloc[i:(i+time_steps)].drop('fechaHoraMDA', axis=1).values
        Xs.append(v)
        ys.append(y.iloc[i+time_steps:i+time_steps+future_steps].values)
    return np.array(Xs), np.array(ys)

time_steps = 48  # las últimas 48 horas como entrada
future_steps = 48  # Predecir las próximas 48 horas

# Iniciar TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=2)
history = []  # Para almacenar la historia de entrenamiento de cada división

for train_index, val_index in tscv.split(train_data):
    train, val = train_data.iloc[train_index], train_data.iloc[val_index]

    X_train, y_train = create_dataset_seq2seq(train, train.precioMarginalLocal, time_steps, future_steps)
    X_val, y_val = create_dataset_seq2seq(val, val.precioMarginalLocal, time_steps, future_steps)

#Estructura de la red:
model_seq2seq = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(units=128, input_shape=(time_steps, X_train.shape[2]), return_sequences=True),
    tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1))  # El "TimeDistributed" es la clave para predecir una secuencia
])
```

Figura 3.32 Estructura del modelo LSTM con Prophet

El bloque de código representado en la Figura 3.32 representa un enfoque más efectivo para modelar series temporales conocido como "secuencia a secuencia" o “secuencial” (*sequence to sequence*, *seq2seq*). Este enfoque es especialmente útil para predecir no solo un solo punto futuro en el tiempo, sino toda una secuencia de puntos futuros:

- **“Función create_dataset_seq2seq”:** Esta función prepara los datos para un modelo seq2seq. Toma una serie de entradas (X) y salidas (y) y las organiza en secuencias de un tamaño definido (*“time_steps”* para entradas, *“future_steps”* para salidas). Por cada iteración, selecciona una ventana de “time_steps” para “X” y una ventana de “future_steps” para “Y”, creando así un conjunto de pares de entrada-salida donde cada entrada está mapeada a una secuencia de salidas futuras en lugar de a un solo valor futuro.
- **Estructura de la red neuronal:** La red utiliza una capa LSTM seguida de una capa *“TimeDistributed”*. La opción *“return_sequences=True”* en la capa LSTM es crucial aquí, ya que permite que la capa LSTM devuelva la secuencia completa de salidas a lo largo del tiempo, en lugar de solo el último punto de tiempo. La capa *“TimeDistributed”* se utiliza para aplicar una capa densa a cada punto de tiempo en la secuencia de salida. Esto permite que la red produzca una secuencia de predicciones de salida, en lugar de una sola predicción.

- **Uso de "TimeSeriesSplit" para la división temporal de los datos:** Al igual que en el primer bloque de código, se utiliza *"TimeSeriesSplit"* para dividir los datos de manera que respete la secuencia temporal. Esto asegura que el modelo se entrene de manera que no se filtre información del futuro hacia el pasado durante el entrenamiento. Este enfoque *"seq2seq"* es especialmente adecuado para predecir series de tiempo donde se requiere una predicción continua para un horizonte de tiempo futuro, como predecir las próximas 48 horas. Al predecir una secuencia completa a la vez, este modelo puede capturar dinámicas y dependencias temporales más complejas en comparación con modelos que predicen un solo punto futuro a la vez.

Una vez hecho esa modificación en LSTM, podremos fusionarlo con el algoritmo de Prophet diseñado por Facebook, que maneja su pronóstico de la siguiente manera, véase Figura 3.33:

```
# Crear DataFrame con fechas futuras y valores pronosticados por el modelo LSTM
future_dates_lstm = pd.date_range(start=last_date, periods=49, freq='H')[1:]
df_lstm_predictions = pd.DataFrame({'ds': future_dates_lstm, 'y': future_predictions_original})

# Crear y entrenar el modelo Prophet con los datos del modelo LSTM
prophet_model_lstm = Prophet(yearly_seasonality=False, daily_seasonality=True)
prophet_model_lstm.fit(df_lstm_predictions)

# Preparar dataframe para predicciones futuras con Prophet
future_lstm = prophet_model_lstm.make_future_dataframe(periods=48, freq='H')

# Hacer predicciones con Prophet usando los datos del modelo LSTM
forecast_lstm = prophet_model_lstm.predict(future_lstm)

# Visualizar predicciones
fig_lstm = prophet_model_lstm.plot(forecast_lstm)
```

Figura 3.33 Comandos de ensamble de los modelos LSTM con Prophet

- **Preparación de Datos para Prophet:** Se adaptaron los datos existentes de precios de energía para su uso en el modelo Prophet. Esto implicó indicar las columnas *"ds"* para las fechas y *"y"* para la variable objetivo (precios de energía) que son los pronósticos del modelo LSTM. Prophet requiere este formato específico para realizar sus predicciones de series temporales. Prophet está diseñado para modelar series temporales univariadas, lo que significa que solo puede manejar una variable objetivo (en este caso, 'y') y una variable de tiempo (en este caso, 'ds'). El algoritmo de Prophet se centra en modelar la tendencia y las estacionalidades de la variable objetivo a lo largo del tiempo.

- **Modelado y Entrenamiento con Prophet:** Se configuró y entrenó un modelo Prophet, excluyendo la estacionalidad anual (*yearly_seasonality=False*) pero incluyendo la diaria (*daily_seasonality=True*). Esta configuración se eligió considerando la naturaleza horaria de los datos de precios de energía. El modelo Prophet aprendió de los patrones históricos para identificar tendencias y estacionalidades en los datos.
- **Generación de Predicciones Futuras:** Se generaron predicciones para un horizonte de 48 horas utilizando Prophet. Para esto, se creó un DataFrame *"future_lstm"* que contenía las fechas futuras para las predicciones. Prophet produjo predicciones para este período, abordando las tendencias y estacionalidades identificadas.
- **Visualización de Resultados:** Las predicciones se visualizaron mediante las funciones integradas de Prophet, proporcionando intervalos de confianza del 95% (por default), sobre las predicciones y de solicitarlo también puede dar los componentes de tendencia y estacionalidad del modelo.

La elección de un algoritmo de pronóstico sobre otro depende de muchos factores, como la naturaleza de los datos, la cantidad de datos disponibles, las características o variables explicativas consideradas, entre otros. Eventualmente hay que probar los algoritmos descritos previamente para medir sus errores de pronóstico y determinar el mejor modelo entre ellos.

3.7 Métricas de desempeño

En cada uno de los algoritmos es crucial el entrenamiento y la validación cruzada (*timeseriesplit*), así como también la experimentación para optimizar su rendimiento. Por lo tanto; se presentan los resultados de entrenamiento y las métricas obtenidas.

3.7.1 Validación del modelo de Árbol de decisión para regresión

En esta etapa es importante obtener y observar el desempeño del modelo, particularmente del algoritmo de árboles, en donde la siguiente figura representa gráficamente el buen o mal pronóstico del modelo durante el entrenamiento con la particularidad de que se procuró respetar la temporalidad de los datos permitiendo hacer una validación cruzada con "*timeseriesplit*", pero considerando la secuencia de los datos y al MAPE para la medición de los pronósticos. Véase la Figura 3.34

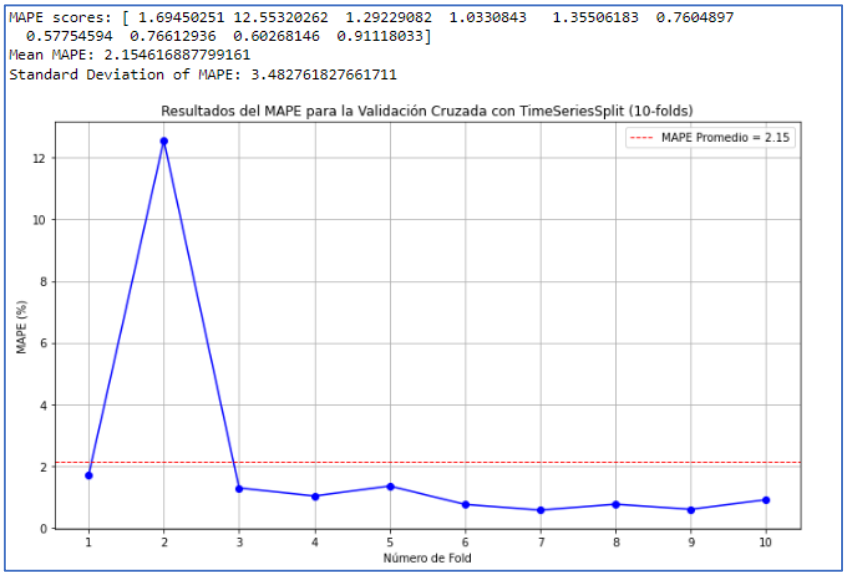


Figura 3.34 Validación del modelo de árbol de decisión para regresión con "*timeseriesplit*"

Podemos observar en la Figura 3.34 como el pronóstico no se ajustó bien a los datos de entrenamiento para el segundo fold, lo que sugiere que el modelo tenía un aprendizaje equivocado al principio, pero fue aprendiendo del segundo y primer fold, y pudo recuperar su eficiencia en el tercer fold y así se mantuvo hacia adelante. El MAPE medio es de 2.1546%, lo que indica que, en promedio, el modelo tiene un error de aproximadamente 2.15% en la predicción de los precios. La desviación estándar de 3.4827, sin embargo, es relativamente alta en comparación con el MAPE medio, lo que refuerza la observación de una variabilidad significativa en el rendimiento del modelo; quizás debido al segundo fold pues está claro que no funcionó bien el modelo, pero en general al considerar a todos los fold resulta en MAPE´s menores al 10%, y se observa un desempeño aceptable.

3.7.2 Métricas de los errores del modelo de árbol de decisión para regresión

En esta sección veremos sus errores para predecir de acuerdo con su MAE, MSE, RMSE, R^2 y MAPE, finalmente, el desempeño del modelo se evalúa usando diversas métricas de regresión. Véase Figura 3.35

```
Métricas de evaluación en el conjunto de prueba:
MAE: 23.773194444444453
MSE: 6247.490331944442
RMSE: 79.04106737604472
R^2: 0.9966527919225916
MAPE: 0.8284484885323669
------------------------------------------------
```

Figura 3.35 Métricas de desempeño del modelo de árboles de decisión para regresión.

En el contexto del rango de valores de la base de datos del Precio Marginal Local (PML), que oscila entre $2,000.^{00}$ y $6,000.^{00}$ pesos, las métricas de evaluación del modelo de árbol de decisión para regresión presentadas sugieren un alto grado de precisión y fiabilidad:

- **Error Absoluto Medio (MAE): $23.^{77}$ pesos:** Considerando el amplio rango de valores en la base de datos, un MAE de aproximadamente $23.^{77}$ pesos es relativamente bajo. Esto indica que, en promedio, las predicciones del modelo desvían ligeramente del valor real, lo cual es significativo dada la escala de los valores manejados.
- **Error Cuadrático Medio (MSE): 6247.49 y Raíz del Error Cuadrático Medio (RMSE): 79.04:** A pesar de que estos valores pueden parecer elevados a primera vista, es importante contextualizarlos con respecto al rango de los datos. Un RMSE de 79.04, que representa aproximadamente el 1.3% del rango total ($4,000.^{00}$ pesos), sugiere que la mayoría de las predicciones están cercanas a los valores reales.
- **Coeficiente de Determinación (R^2): 0.9967:** Un valor de R^2 cercano a 1, como el observado, implica que el modelo logra explicar casi la totalidad de la variabilidad en los datos. Este es un indicador robusto de un ajuste de modelo excepcional.
- **Error Porcentual Absoluto Medio (MAPE): 0.8284%:** Un MAPE por debajo del 1%, especialmente en el contexto del rango de precios manejado, es notablemente bajo. Esto sugiere que el porcentaje promedio de error en las predicciones es menor al 1%, lo que denota una alta precisión en las estimaciones del modelo.

La grafica de la Figura 3.36 nos da clara representación de los valores reales vs los pronosticados y como las métricas traducen lo que la gráfica representa. El gráfico incluye tres series de datos con diferentes marcadores y colores para una fácil distinción, así como títulos, etiquetas y leyendas para mayor claridad. Se ajusta la rotación de las etiquetas del eje x para mejorar la legibilidad.

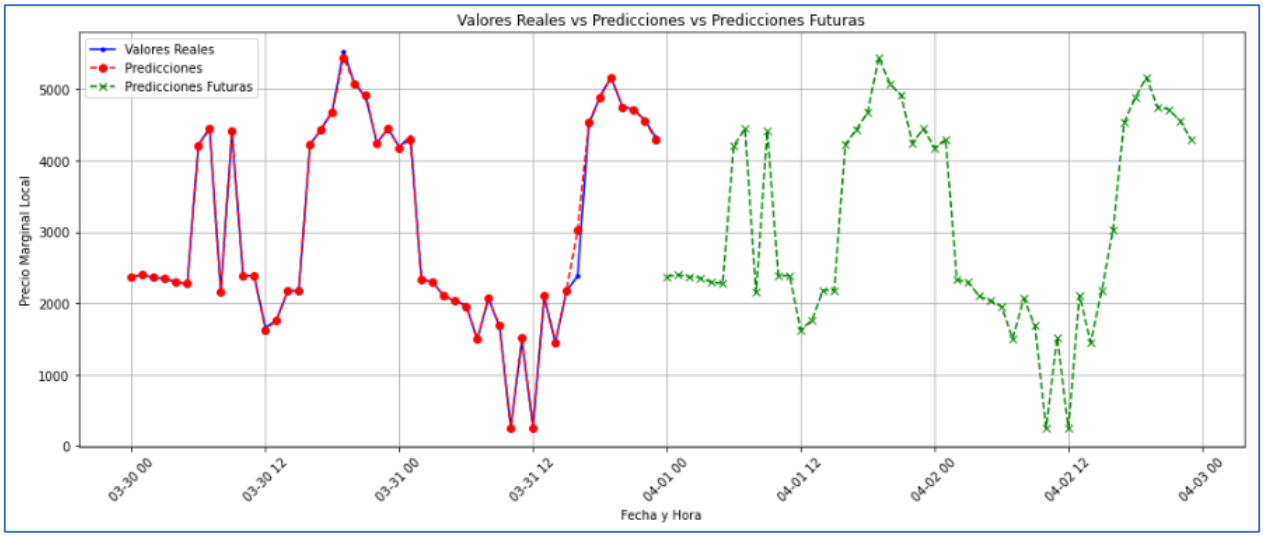


Figura 3.36 Grafica de pronósticos del PML con el modelo de árbol de decisión para regresión.

3.7.3 Interpretación general del modelo de árbol de decisión para regresión

El modelo de árbol de decisión para regresión parece tener un rendimiento excepcional tanto en el conjunto de validación como en el conjunto de prueba, según las métricas obtenidas. No obstante, el algoritmo solo puede predecir un punto hacia adelante con ese desempeño, puesto que al realizar más de un pronóstico este se estancará inevitablemente porque no dispone de datos para ese futuro y después de la primera predicción, el modelo simplemente sigue "reciclando" su predicción más reciente, resultando en predicciones estancadas o una línea recta; este problema no es exclusivo de los árboles de decisión y puede ocurrir con cualquier modelo que no tenga en cuenta la secuencia temporal en sus predicciones.

3.7.4 Validación del modelo de Regresión polinominal de segundo grado

La regresión polinomial de segundo grado fue sometida a una validación cruzada rigurosa utilizando el método *"TimeSeriesSplit"* con 10 divisiones. Esta metodología es particularmente adecuada para series temporales, ya que respeta el orden cronológico de los datos, un aspecto crucial para evitar la contaminación del modelo con información futura durante el entrenamiento. Véase Figura 3.37

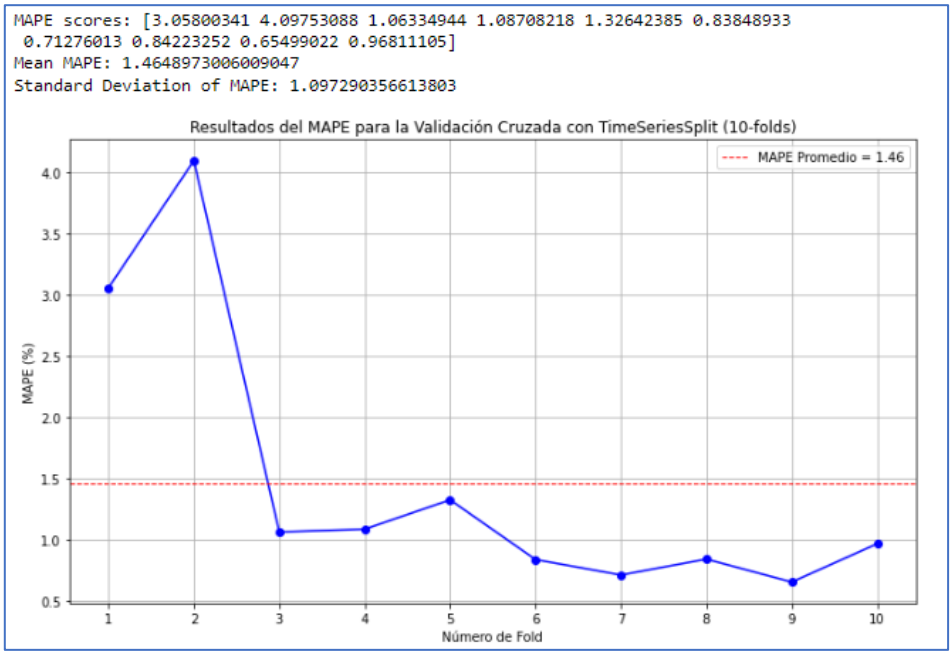


Figura 3.37 Validación del modelo de Regresión polinominal mediante "*timeseriesplit*"

Estos resultados muestran una variabilidad en el desempeño del modelo a lo largo de diferentes segmentos de la serie temporal. El MAPE promedio fue de 1.465% con una desviación estándar de 1.097%. El segundo "*fold*", en particular, muestra un MAPE significativamente más alto (4.098%) en comparación con los otros "*folds*", similar al modelo de árboles anterior.

3.7.5 Métricas de los errores del modelo de Regresión polinominal de segundo grado

A continuación, en la Figura 3.38 están las medidas de los errores de predicción de este modelo son:

```
Métricas de evaluación en el conjunto de prueba:
MAE: 0.003700827215135168
MSE: 3.294324523166005e-05
RMSE: 0.005739620652243496
R^2: 0.99999999998235
MAPE: 0.00018884680450356926
------------------------------------------------
```

Figura 3.38 Métricas de desempeño del modelo de Regresión Polinominal

En el contexto de los precios marginales locales (PML) del mercado eléctrico, donde los valores fluctúan entre \$2,000.^{00} y \$6,000.^{00} pesos, las métricas de rendimiento del modelo de regresión polinomial de segundo grado adquieren una perspectiva interesante. Brevemente se

ofrece una interpretación detallada de cada una de estas métricas:

- **MAE (Error Absoluto Medio) de 0.0037:** Este valor es extremadamente bajo, especialmente si se considera que los precios varían en un rango de varios miles. Sugiere que las predicciones del modelo están muy cerca de los valores reales, lo que es excepcional en un mercado con esta amplitud de variación de precios.
- **MSE (Error Cuadrático Medio) y RMSE (Raíz del Error Cuadrático Medio):** Indica que el modelo hace predicciones muy precisas, con errores cuadráticos mínimos. Sin embargo, dado el amplio rango de precios en el mercado eléctrico, estos valores tan bajos podrían ser una señal de que el modelo está sobreajustado a los datos de entrenamiento.
- **R^2 cercano a 1:** Sugiere que el modelo explica casi toda la variabilidad de los precios.
- **MAPE** casi cero: Un MAPE tan bajo es extraordinario, especialmente en un mercado con variaciones significativas de precios. Esto indica que los errores porcentuales son casi inexistentes, lo cual es inusual en la predicción de series temporales financieras o económicas.

Se puede ver como son los valores pronosticados vs los reales en este modelo con la siguiente Figura 3.39 que el método de regresión es capaz de ajustarse a los valores reales históricos casi a la perfección, pero para datos futuros no podrá dar precios pronosticados sin precios reales, por lo que se recurrió a mover los últimos 48 precios predichos por el modelo ahora hacia el futuro y nos encontramos en la típica situación de modelos de regresión:

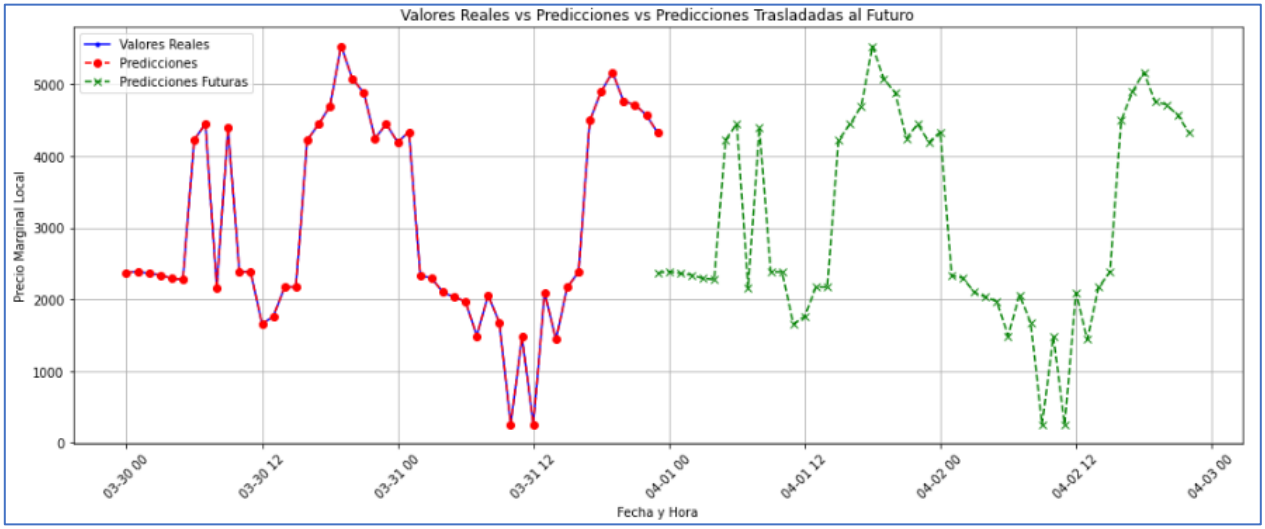


Figura 3.39 Gráfica de pronósticos del modelo de Regresión Polinominal.

3.7.6 Interpretación general del modelo regresión polinominal

La regresión polinómica de segundo grado parece ajustarse excepcionalmente bien a los datos, según estas métricas. De hecho, el rendimiento es aún mejor que el del modelo de árbol de decisión para regresión que analizamos anteriormente, resultan parecidos. Este modelo predice bien datos existentes, pero en realidad no es capaz de predecir con fidelidad el PML a pesar de sus excepcionales métricas, puesto que no capturan la temporalidad ni son capaces de generalizar bien valores volátiles en sus pronósticos a futuro. Como se ilustra en la Figura 3.39, las proyecciones futuras generadas por el modelo no son más que extrapolaciones de los datos históricos. Se agrego en el código que replicara sus predicciones para las siguientes 48 horas. Sin embargo, esta aproximación presupone que los patrones pasados se mantendrán constantes, lo cual es una suposición arriesgada en contextos donde los precios son altamente volátiles y sujetos a numerosos factores externos e internos. Aunque las métricas del modelo indican un alto grado de precisión, es fundamental tener en cuenta estas consideraciones al interpretar su capacidad predictiva en escenarios reales y dinámicos.

3.7.7 Validación del modelo de SARIMA:

La validación del modelo se hizo también con "*timeseriessplit*", obteniendo los gráficos de cada *"fold"* a continuación. Véase Figura 3.40:

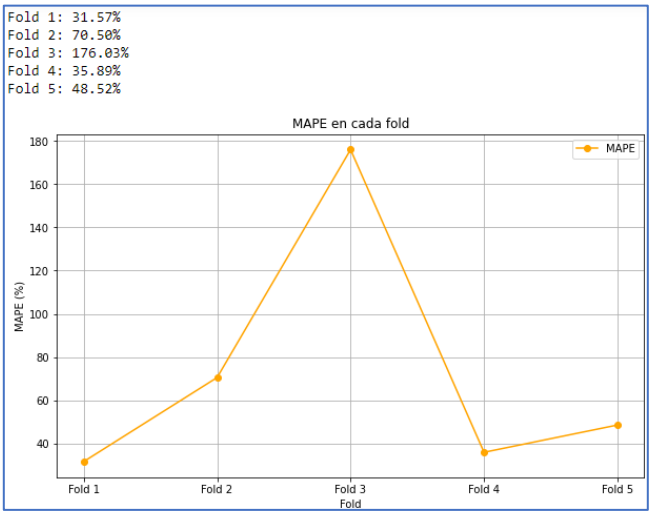


Figura 3.40 Validación del modelo SARIMA con "*timeseriesplit*"

Los valores de MAPE varían ampliamente, con un pico notablemente alto de 176.03%, lo

que indica una baja precisión en esa parte específica del conjunto de datos.

3.7.8 Métricas de los errores del modelo de SARIMA:

Los resultados del modelo SARIMA indica que los pronósticos no son exactos y tienden a equivocarse en alto grado; lo podemos comprender en los siguientes errores, véase Figura 3.41:

```
Métricas de evaluación en el Conjunto de datos:
MAE: 6641.7567363131175
MSE: 58859458.453115
RMSE: 7671.99181784724
MAPE: 292.16%
------------------------------------------------
MAPE: 292.1631100615836
```

Figura 3.41 Métricas de desempeño del modelo SARIMA.

El resultado del modelo se puede ver en un gráfico de los valores reales vs los pronosticados en la siguiente Figura 3.42 donde se puede observar cómo capta el patrón estacional correctamente, pero su tendencia hacia arriba hace que se alejen los pronósticos de los reales y entonces se comprende el porqué de las métricas de los errores tan elevadas:

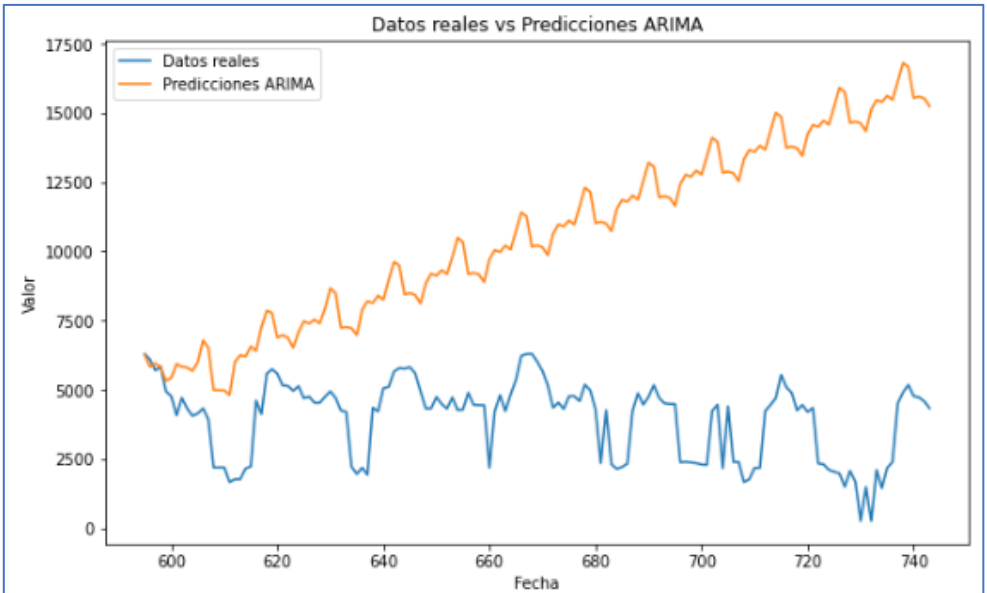


Figura 3.42 Valores reales vs Pronosticados por el modelo SARIMA.

- **MAE (Mean Absolute Error) de 6641.7567:** Dado que el MAE es mayor que el rango entero de precios del PML ($2,000.^{00}$ a $6,000.^{00}$ pesos), esto indica que el error promedio de las predicciones es más grande que los valores reales típicos en sus datos. Esto es un indicativo claro de un modelo que no está funcionando adecuadamente para estos datos.
- **MSE (Mean Squared Error) de 58859458.4531 y RMSE (Root Mean Squared Error) de 7671.9918:** Estas métricas, que son sensibles a los errores grandes debido a su naturaleza cuadrática, también son extremadamente altas. El RMSE es incluso mayor que el MAE, lo que sugiere la presencia de algunos errores de predicción muy grandes.
- **MAPE de 292.16%:** Este valor indica que, en promedio, el error absoluto de sus predicciones es casi tres veces el valor real de los datos. En un contexto donde los valores están en el rango de $2,000.^{00}$ a $6,000.^{00}$ pesos, esto significa que las predicciones son extremadamente imprecisas.

La siguiente gráfica representa los pronósticos, donde se observan predicciones similares a los comportamientos de los precios, pero notablemente menores en magnitud y posiblemente el modelo los suaviza o ajusta entre valores promedios (a pesar de ser el mejor modelo determinado con Auto ARIMA), véase Figura 3.43.

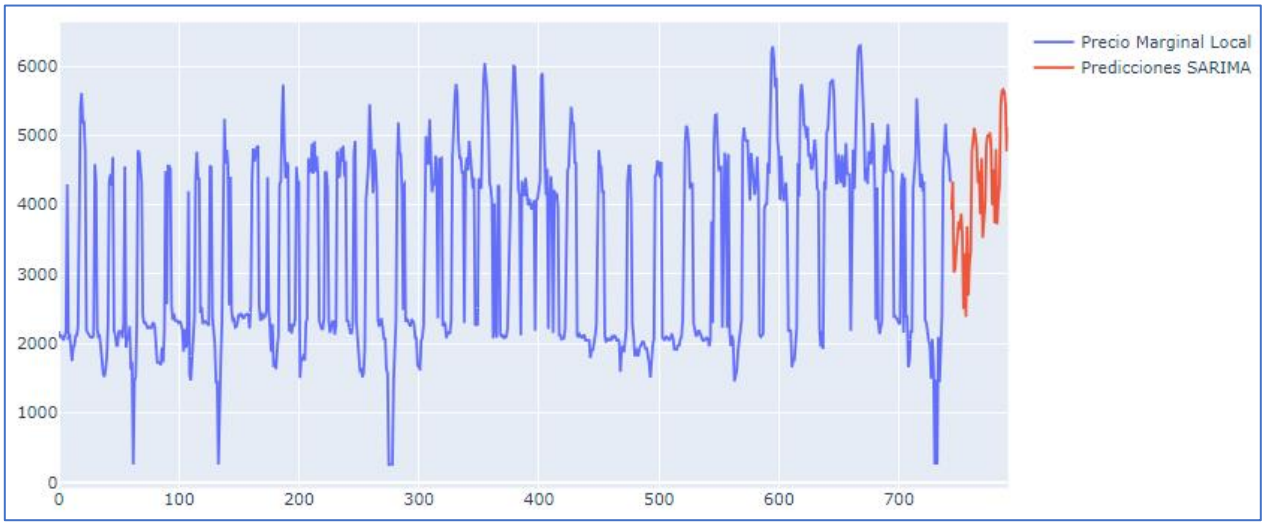


Figura 3.43 Gráfica del pronóstico del PML con el modelo SARIMA.

3.7.9 Interpretación general del modelo SARIMA:

Comparado con los modelos anteriores (árbol de decisión y regresión polinominal), el modelo SARIMA parece tener un rendimiento considerablemente peor en los pronósticos, según

las métricas proporcionadas. Las métricas sugieren que las predicciones del modelo SARIMA tienen errores significativos en comparación con los valores verdaderos.

3.7.10 Validación del modelo de LSTM:

Las métricas de evaluación del modelo LSTM, en el contexto de los precios del Precio Marginal Local (PML) para el nodo 07LPZ-115 con precios entre los $2,000.^{00}$ y $6,000.^{00}$, podremos contextualizar las métricas obtenidas, veamos la Figura 3.44:

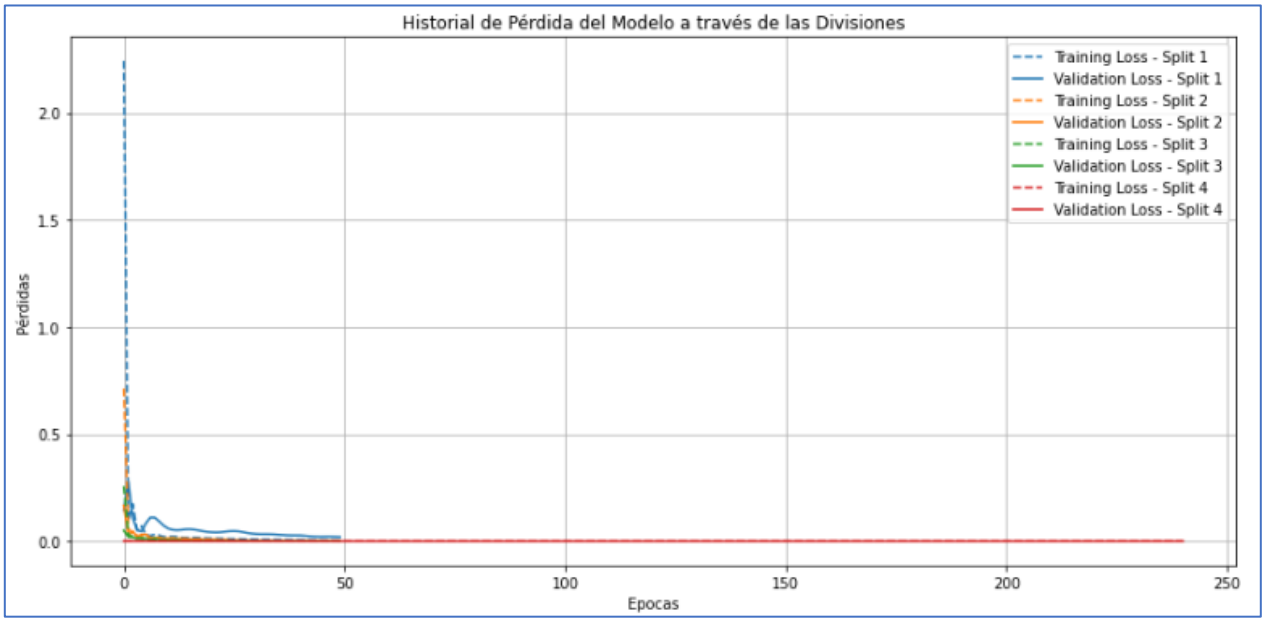


Figura 3.44 Validación del modelo LSTM con "*timeseriesplit*".

Se observa una tendencia a mínimos cercanos a 0 para cada fold de la validación, lo que sugiere que se aproximan mucho los pronósticos del modelo a los reales, más sin embargo estos valores y el número de épocas podría indicar un sobre ajuste. Para controlarlo se consideraron:

- **La validación con Splits (folds)** que ayuda a garantizar que el modelo sea generalizable y no esté sobreajustado a un subconjunto específico de los datos de entrenamiento.
- **Early Stopping:** un "callback" de "EarlyStopping" monitorea la pérdida de validación (val_loss). Si no se observa una mejora en esta métrica durante 200 épocas, el entrenamiento se detiene. Esto evita que el modelo se sobreajuste al continuar aprendiendo de los datos de entrenamiento más allá del punto en el que ya no mejora su rendimiento en los datos de validación.

- **Ajuste del Learning Rate (Tasa de Aprendizaje):** un "*LearningRateScheduler*" para ajustar dinámicamente la tasa de aprendizaje durante el entrenamiento. Si las épocas son menores a 25, se mantendrá la tasa de aprendizaje, pero después de 25 épocas, se reducirá gradualmente esta tasa. Reducir la tasa de aprendizaje puede ayudar a evitar que el modelo se sobreajuste, ya que hace más pequeños y precisos los ajustes a los pesos del modelo a medida que se acerca a la convergencia.
- **Shuffle:** Al habilitar el "shuffle" en el entrenamiento, se asegura que los datos se presenten al modelo en un orden diferente en cada época. Esto ayuda a prevenir el sobreajuste ya que el modelo no aprenderá secuencias específicas de los datos de entrenamiento.

3.7.11 Métricas de los errores del modelo LSTM:

A continuación, las métricas del modelo LSTM, véase **Figura 3.45**.

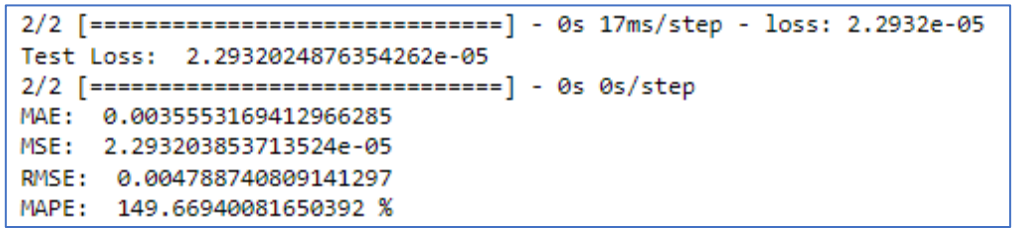


Figura 3.45 Métricas de desempeño del modelo LSTM

En el contexto del Precio Marginal Local (PML), las métricas del modelo LSTM indican lo siguiente:

- **MSE (2.2932e-05) y RMSE (0.0047):** Valores bajos sugieren un alto grado de precisión en las predicciones.
- **MAE (0.0035):** Indica un error absoluto promedio bajo, lo que refleja predicciones precisas.
- **MAPE (149.66%):** Es inusualmente alto, lo cual puede ser causado por valores de la dataset "PML" cercanos a cero o por errores significativos en ciertas predicciones. Posiblemente sea error de predicción en horas donde se toque picos o valles atípicos del PML o porque las variables independientes contienen muchos ceros en ellos.

La siguiente **Figura 3.46** representa los valores reales en color azul, a los pronosticados en color rojo y a los valores pronosticados a futuro en verde, se puede reconocer la destacable

similitud con la que predice el modelo.

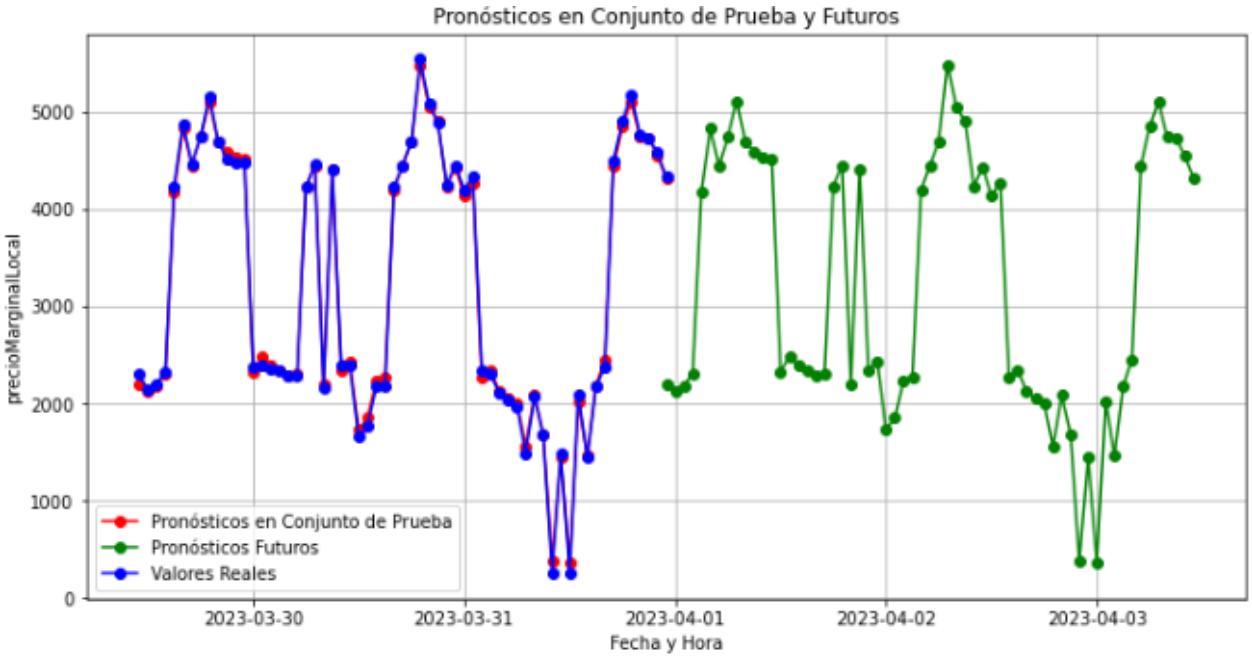


Figura 3.46 Grafica del pronóstico del PML con el modelo LSTM

3.7.12 Interpretación general del modelo LSTM:

Las métricas sugieren que la LSTM tiene un buen rendimiento en los datos, especialmente cuando se compara con el modelo SARIMA anteriormente. El MAPE es el único valor elevado y sugiere que el modelo no es capaz de predecir fielmente. Las razones serian la volatilidad o la presencia de valores cero en las variables, como es en las variables de precipitación y de congestión principalmente.

3.7.13 Validación del modelo LSTM + Prophet:

Para la validación del modelo ensamblado LSTM + Prophet se puede representar en la siguiente Figura 3.47.

```
plt.figure(figsize=(14, 5))
plt.plot(history_model_seq2seq.history['loss'], label='Perdida durante el entrenamiento')
plt.plot(history_model_seq2seq.history['val_loss'], label='perdida en la validacion')
plt.title('Función de pérdida del modelo')
plt.ylabel('Pérdidas')
plt.xlabel('Epocas')
plt.legend()
plt.show()
```

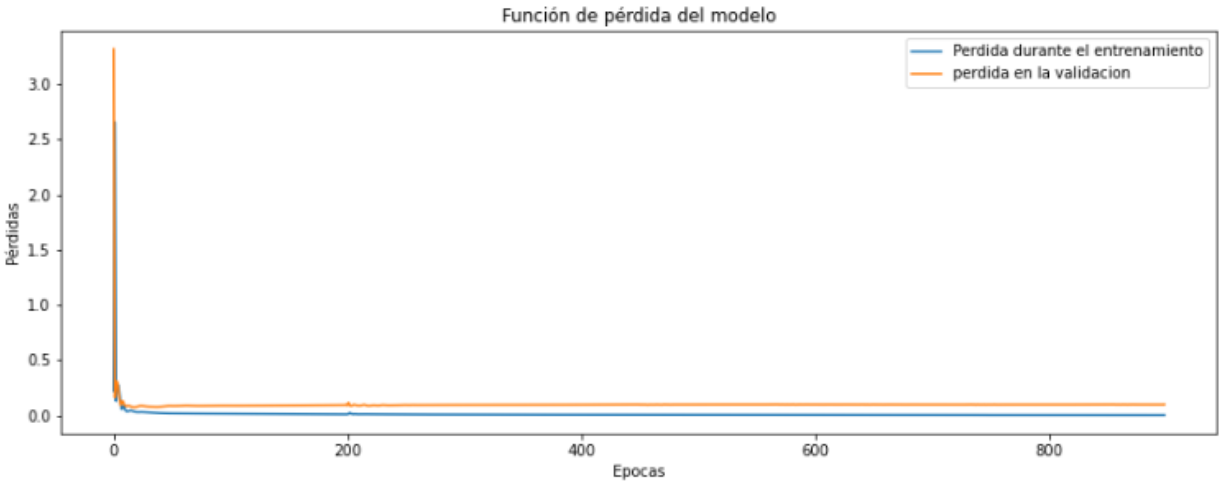


Figura 3.47 Validación del modelo LSTM + Prophet con "timeseriesplit".

El modelo LSTM es básicamente el mismo y la diferencia es la neurona de salida que arroja una secuencia, entonces la representación de sus métricas y validaciones son básicamente las mismas que las anteriormente descritas. No obstante, se presenta la Figura 3.47 para ver como esta version aprende desde tempranas épocas de entrenamiento y validación, estancándose en valores aceptables que evidentemente ya no convergen, lo que sugiere un mínimo local.

3.7.14 Métricas de los errores del modelo LSTM + Prophet:

Las métricas del modelo de ensamble LSTM + Prophet tenemos en la Figura 3.48

```
3/3 [==============================] - 1s 11ms/step
MAE: 0.11455087164886431
MSE: 0.026858547622147375
RMSE: 0.16388577614346944
MAPE: 35.13008442779011%
```

Figura 3.48 Métricas de desempeño del modelo LSTM + Prophet.

- **Mean Absolute Error (MAE) de 0.1146 y Mean Squared Error (MSE) de 0.02686:** Este valor indica un error promedio de 0.1146, dado el rango de PML ($2,000.^{00}$ a $6,000.^{00}$), este error absoluto puede considerarse relativamente pequeño.
- **Root Mean Squared Error (RMSE) de 0.1639:** Aparentemente el modelo capta muy bien los precios del PML durante el entrenamiento.
- **Mean Absolute Percentage Error (MAPE) de 35.13%:** Este porcentaje indica que, en promedio, las predicciones del modelo pueden desviarse un 35.13% del valor real. En el contexto de PML, esto puede ser significativo, especialmente en mercados volátiles pues el error se dispararía a valores aún más altos.

La siguiente grafica observamos que Prophet hace que los pronósticos obedezcan a los patrones de tendencia y estacionalidad diaria, véase Figura 3.49:

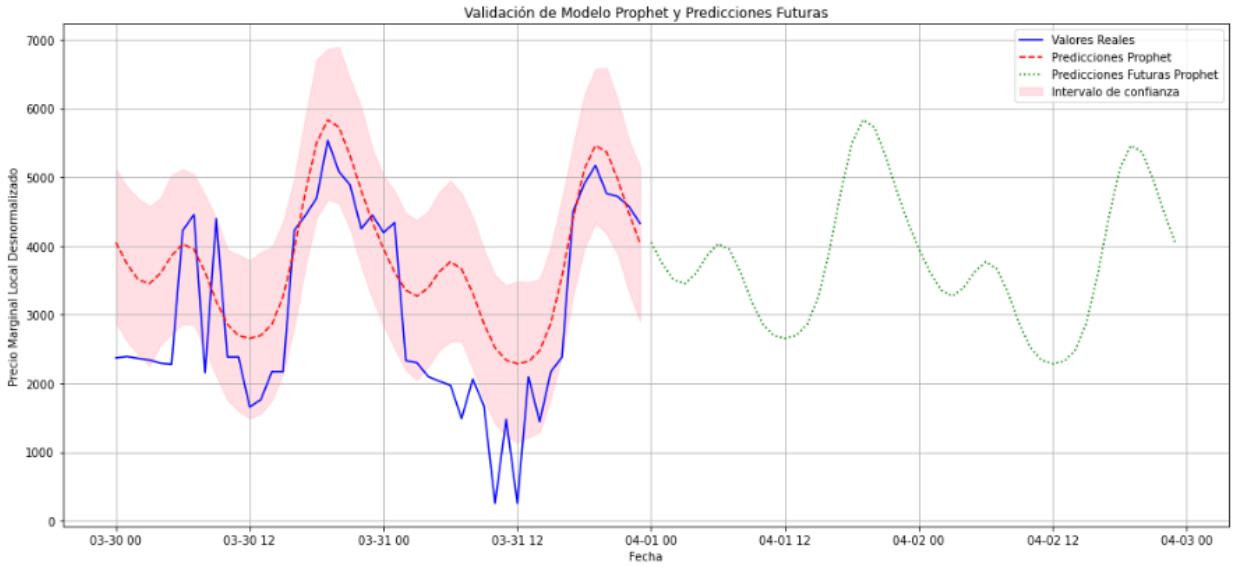


Figura 3.49 Grafica de los pronósticos del modelo LSTM + Prophet.

3.7.15 Interpretación general del modelo LSTM + Prophet:

El modelo combinado LSTM + Prophet muestra un MAPE de 35.13%. Este valor sugiere que, aunque el modelo tiene una capacidad moderada para predecir tendencias generales, puede tener limitaciones al predecir con precisión valores atípicos o picos extremos en los PML. Esta observación se ve respaldada por la Figura 3.49, donde los precios que se encuentran fuera del intervalo de confianza representan estos valores atípicos.

Este comportamiento es particularmente relevante en el contexto del nodo 07LPZ-115, donde los valores extremos pueden ser críticos para la toma de decisiones. La capacidad del modelo para capturar las tendencias generales puede ser útil para ciertas aplicaciones, pero su rendimiento en situaciones de alta variabilidad o con precios extremos debe ser considerado cuidadosamente.

3.8 Comparación de los algoritmos de pronósticos:

El debate del ¿por qué un modelo de decisión es mejor para predecir que otros diseñados para series temporales?, las razones pueden ser la naturaleza de los datos, la cantidad de datos disponibles, las características o variables explicativas consideradas, entre otros, por lo que corresponde hacer una comparativa más detallada al respecto, pero siempre es útil considerar la interpretación visual de los resultados y la comprensión del contexto de los datos.

A continuación, una tabla (véase Figura 3.50) compara las métricas de los modelos para el nodo 07LPZ-115, donde el mejor MAPE es del modelo de RPSG, después del árbol de decisión y en tercer lugar el LSTM + Prophet. Esta tabla es una herramienta valiosa para entender las fortalezas y limitaciones de cada modelo en el contexto específico del nodo 07LPZ-115:

Modelo	MAE	MSE	RMSE	MAPE
Árbol de Decisión para Regresión	23.77	6247.49	79.04	0.83%
Regresión Polinomial (2º Grado)	0.0037	3.2943	0.0057	0.00%
SARIMA	6641.76	58859458.5	7671.99	292.16%
LSTM	0.0035	2.2932E-05	0.00478	149.66%
LSTM + Prophet	0.1145	0.0268	0.1638	35.13%

Figura 3.50 Comparación de las métricas de los modelos.

En general, el rendimiento de un algoritmo sobre otro puede ser el resultado de cómo se ajusta el algoritmo a la estructura subyacente de los datos (si son numéricos o categóricos) la disponibilidad de los datos y la complejidad intrínseca de la problemática a abordar. Es posible que algunos algoritmos sean más adecuados para ciertos tipos de patrones y tendencias que otros. Lo correspondiente es probar distintos algoritmos para poder predecir el PML y una vez seleccionado el mejor de ellos recurrir a una afinación del método para lograr mejores resultados. Por lo que se recurre a aplicar los algoritmos en un solo script de Python; se unificaran los

métodos de pronósticos en un modelo (código de pronóstico) para tener los resultados disponibles para el usuario en un solo código y en una sola corrida compilada.

3.9 Afinación (Tuning):

En la presente sección, abordamos la optimización de hiperparámetros que conlleva al modelo *Long Short-Term Memory* (LSTM), un paso esencial en la consolidación del enfoque analítico. A diferencia de otros modelos implementados en este estudio, el LSTM presenta una complejidad inherente en su arquitectura, lo que requiere un análisis detallado y un ajuste meticuloso de sus hiperparámetros. Este proceso no solo se hizo para mejorar la precisión y reducir el error en las predicciones, sino también para garantizar la estabilidad y la generalización del modelo frente a variaciones en los datos. La selección cuidadosa de hiperparámetros, como el número de capas, unidades por capa, tasa de aprendizaje y otros, en busca de un buen desempeño global del modelo.

3.9.1 Ingeniería de características:

Adicionalmente, al construir un modelo de pronóstico, resulta beneficioso emplear técnicas de ingeniería de características para capturar relaciones no lineales y las interacciones entre las variables. La aplicación de transformaciones como el seno y el coseno a las características temporales es una estrategia común en la ingeniería de características, destinada a identificar patrones cíclicos o senoidales en los datos. Esta técnica es particularmente valiosa para variables como la hora del día, el mes o la época del año, permitiendo una representación más detallada y adaptativa de los patrones temporales que se repiten regularmente en los datos.

El razonamiento detrás de esto es que muchas de estas características temporales tienen una naturaleza cíclica. Al aplicar las funciones seno y coseno a la hora del día, convertimos una variable lineal en dos nuevas características que capturan mejor la periodicidad y la continuidad cíclica de los datos. Por ejemplo, en una escala de tiempo de 24 horas, el salto de 23 a 0 crea una discontinuidad que no refleja la realidad cíclica del tiempo. Las funciones seno y coseno suavizan este salto, representando las 23:00 y 00:00 como puntos cercanos en el ciclo, véase Figura 3.51

```python
from sklearn.preprocessing import MinMaxScaler

# Ingeniería de Características
def feature_engineering(df):
    # Agregar características cíclicas para la hora del día y el día de la semana
    df['hora_sin'] = np.sin(2 * np.pi * df['fechaHoraMDA'].dt.hour / 24)
    df['hora_cos'] = np.cos(2 * np.pi * df['fechaHoraMDA'].dt.hour / 24)
    df['dia_semana_sin'] = np.sin(2 * np.pi * df['fechaHoraMDA'].dt.dayofweek / 7)
    df['dia_semana_cos'] = np.cos(2 * np.pi * df['fechaHoraMDA'].dt.dayofweek / 7)

    # Normalizar las características numéricas usando MinMaxScaler
    scaler = MinMaxScaler()
    numeric_columns = ['precioMarginalLocal', 'componenteEnergia', 'componentePerdidas',
                       'componenteCongestion', 'Temperatura del Aire (°C)', 'Precipitación (mm)',
                       'Radiación Solar (W/m²)', 'Rapidez de viento (km/h)']

    df[numeric_columns] = scaler.fit_transform(df[numeric_columns])

    # Devolver el objeto scaler para su uso posterior
    return df, scaler

# Aplicar ingeniería de características
PML, scaler = feature_engineering(PML)
```

Figura 3.51 Comandos de aplicación de ingeniería de características al dataset "PML".

Esto dará dos nuevas características que capturan el ciclo diario de la hora original. Estas características transformadas tendrán valores que oscilan entre -1 y 1 y describirán un círculo unitario a medida que se avanza en el día.

Este tipo de ingeniería de características puede ayudar a los modelos, especialmente a aquellos que no capturan inherentemente la naturaleza cíclica de ciertos datos (como los modelos lineales), a entender y aprovechar mejor los patrones subyacentes en las características temporales. Además de usar "*MinMaxScaler*" para la normalización de los datos, ya que ayuda a que todas las características contribuyan de manera equitativa al modelo sin que una característica domine debido a su escala más amplia. Estas transformaciones pueden llevar a un mejor rendimiento de los modelos que utilizamos para pronosticar el PML de energía, pero siempre es importante validar el impacto de estas transformaciones en el rendimiento del modelo mediante pruebas de validación cruzada para series temporales.

3.9.2 Validación con Walk-forward.

Ambos enfoques, "*TimeSeriesSplit*" y "*Walk-Forward",* ofrecen perspectivas valiosas. "*TimeSeriesSplit*" evalúa el modelo en diferentes divisiones temporales para obtener una visión más global del rendimiento a lo largo del tiempo. Por otro lado, *Walk-Forward* simula la capacidad de la red para adaptarse continuamente a nuevos datos, emulando un entorno de

producción dinámico. Utiliza ventanas de entrenamiento y validación que avanzan en el tiempo, representando así una simulación más dinámica de la vida real. Al renovar el entrenamiento en cada iteración y ajustar la red a datos más recientes, este enfoque busca mantener la capacidad predictiva de la red en situaciones cambiantes o "nunca vistas" por decirlo de cierta manera, véase Figura 3.52

```
# Definir tamaños para la ventana de entrenamiento y validación
train_window = 150  # Reducido de 200 a 150
val_window = 75     # Reducido de 100 a 75
step_size = 75      # Reducido de 100 a 75
n_splits = (len(train_data) - train_window) // step_size

history = []

for i in range(n_splits):
    start_train = i * step_size
    end_train = start_train + train_window
    start_val = end_train
    end_val = start_val + val_window

    train = train_data.iloc[start_train:end_train]
    val = train_data.iloc[start_val:end_val]

    X_train, y_train = create_dataset(train, train.precioMarginalLocal, time_steps)
    X_val, y_val = create_dataset(val, val.precioMarginalLocal, time_steps)

    # Construcción del modelo (esto reinicializa el modelo en cada split)
    model = tf.keras.models.Sequential([
        tf.keras.layers.LSTM(units=128, input_shape=(X_train.shape[1], X_train.shape[2])),
        tf.keras.layers.Dense(units=1)
    ])

    # Compilación del modelo
    optimizer = tf.keras.optimizers.Adam(learning_rate=0.011)
    model.compile(loss='mean_squared_error', optimizer=optimizer)
```

Figura 3.52 Comandos de validación al modelo con "*walk-forward*"

En la Figura 3.52 se definen los tamaños de las ventanas de entrenamiento "*train_window*" y validación "*val_window*", así como el tamaño del paso "*step_size*" y el número de divisiones "*n_splits"*. Se itera a través de *"n_splits"* que se realizarán a lo largo de los datos temporales basándose en la longitud total de los datos "*len(train_data)"* y el tamaño del paso.

Se utilizan índices "*start_train, end_train, start_val, end_val"* para seleccionar las secciones correspondientes de datos de entrenamiento "*train*" y datos de validación "*val*". Estos índices se actualizan en cada iteración según el tamaño del paso y las longitudes de las ventanas. Se emplea la función "*create_dataset*" para preparar los datos de entrenamiento *"X_train, y_train"* y los datos de validación "*X_val, y_val*" en secuencias adecuadas para el modelo LSTM. Esto implica estructurar los datos en lotes secuenciales que el modelo pueda utilizar para el aprendizaje.

Antes de entrenar el modelo, se muestra la arquitectura del modelo LSTM dentro del bucle *for*. Es importante destacar que el modelo se reinicializa en cada iteración, asegurando que comience desde cero en términos de conocimiento acumulado. El modelo se entrena utilizando los datos de entrenamiento "*X_train, y_train*" y se valida con los datos de validación "*X_val, y_val*". Esto se realiza en cada iteración, y el entrenamiento se realiza sin barajar los datos "*shuffle=False*" para mantener la coherencia temporal. La curva de aprendizaje debe converger en cero llegando a su mínimo, como puede apreciarse en la Figura 3.53.

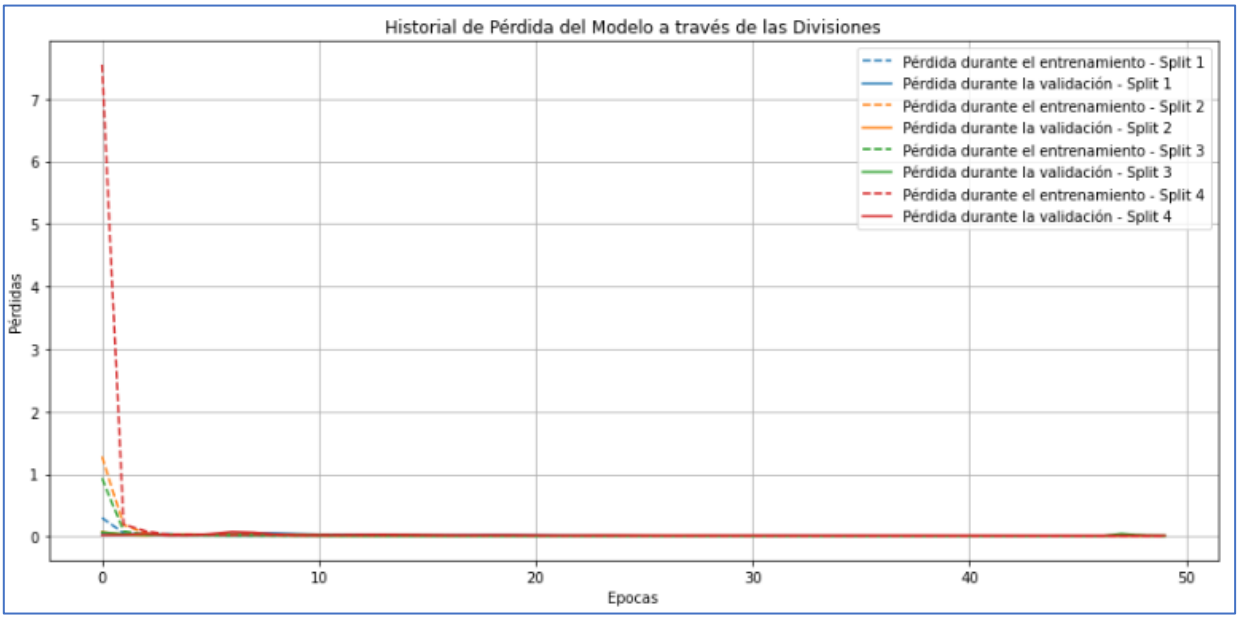


Figura 3.53 Validación del modelo LSTM + Prophet con "*Walk - forward*".

Usar diferentes metodologías de validación en series temporales es una práctica recomendable. En el bloque de código donde se utilizó *TimeSeriesSplit*, el modelo divide el conjunto de datos en k particiones consecutivas manteniendo el orden temporal. Cada partición se utiliza como conjunto de validación mientras que el resto se utiliza para entrenar el modelo. Esto es beneficioso para evaluar el rendimiento del modelo en diferentes segmentos temporales.

Por otro lado, en la validación *walk-forward*, simula más de cerca, cómo se utilizaría el modelo en un entorno de producción. Aquí, el modelo entrena con datos hasta un punto en el tiempo y luego hace predicciones sobre el futuro inmediato. Este enfoque es más parecido a cómo se despliega un modelo en producción (implementación), ya que se actualiza continuamente el conjunto de entrenamiento y se realizan predicciones para el futuro próximo.

Ambos enfoques, *TimeSeriesSplit* y validación *walk-forward*, son útiles, pero tienen objetivos ligeramente diferentes. *TimeSeriesSplit* se centra en evaluar el rendimiento en diferentes segmentos temporales en forma global y ordenada, mientras que la validación *walk-forward* proporciona una perspectiva más dinámica y localizada, al dividir la serie en ventanas y reiniciar el entrenamiento en cada iteración, se emula la capacidad del modelo para adaptarse y aprender de manera continua en entornos cambiantes.

3.9.3 Análisis de sensibilidad:

La sección que vamos a abordar se centra en la mejora del rendimiento de un modelo de pronóstico basado en redes neuronales en el contexto específico de la predicción de precios marginales locales. Este modelo inicial, aunque capaz de proporcionar predicciones aceptables, presenta ciertos límites en cuanto a su capacidad de generalización y precisión. Para abordar esta limitación, se llevó a cabo un análisis de sensibilidad mediante la exploración sistemática de diferentes configuraciones de la arquitectura del modelo y ajustes de hiperparámetros.

El propósito es evaluar variaciones en la complejidad de la red, mediante la adición de capas y ajustes en los hiperparámetros, y como afectan la capacidad del modelo para aprender patrones más intrincados y realizar pronósticos más precisos. Esta estrategia busca determinar si una red neuronal más robusta y compleja puede mejorar significativamente la capacidad de generalización y la precisión de las predicciones en comparación con el modelo inicial.

La red neuronal que identificaremos como “1+1” es una arquitectura que tiene una capa LSTM con 128 unidades y retorna secuencias en lugar de un solo valor. La capa *TimeDistributed* se utiliza para aplicar la capa densa a cada paso de tiempo.

Ahora bien, la red neuronal que identificaremos en este análisis como “2+1” su arquitectura es más compleja: Incluye dos capas LSTM (128 y 64 unidades, respectivamente), capas de “*Batch Normalization*” y “*Dropout*”, y utiliza la técnica de “*Repeat Vector*” para repetir la salida antes de alimentarla a otra capa LSTM en las primeras dos capas, y la capa “*Time Distributed*” para la salida también.

La inclusión de capas de "*Batch Normalization*" y "*Dropout*" en la arquitectura "2+1" tiene un propósito específico en el contexto de la mejora de la robustez del modelo. La capa de *"Batch Normalization"* contribuye a estabilizar el entrenamiento normalizando las activaciones intermedias durante el proceso de aprendizaje, lo que puede ayudar a evitar problemas como el desvanecimiento del gradiente.

Por otro lado, la capa de *"Dropout*" introduce regularización al apagar aleatoriamente neuronas durante el entrenamiento, reduciendo así la posibilidad de sobreajuste y mejorando la generalización del modelo. Además, la técnica de *"RepeatVector"* se emplea para repetir la salida de las capas LSTM antes de alimentarla a otra capa LSTM. Esta repetición proporciona información contextual adicional al modelo, permitiendo una representación más rica de las secuencias de datos. Finalmente, la capa "*TimeDistributed*" aplicada a la salida garantiza que la predicción se realice para cada paso de tiempo, ofreciendo una salida secuencial coherente. Véase la estructura claramente en la Figura 3.54:

```
# Estructura de la red "1 + 1"(Modelo Inicial):
model_seq2seq = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(units=128, input_shape=(time_steps, X_train.shape[2]), return_sequences=True),
    tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1))  # El "TimeDistributed" es la clave para predecir una secuencia
])

# Estructura de la red "2 + 1"(Modelo robusto):
model_seq2seq = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(128, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    tf.keras.layers.BatchNormalization(momentum=0.3),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.LSTM(128, return_sequences=False),
    tf.keras.layers.RepeatVector(future_steps),
    tf.keras.layers.LSTM(64, return_sequences=True),
    tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1))
])
```

Figura 3.54 Arquitecturas de las 2 redes LSTM a someter el análisis de sensibilidad.

La siguiente Figura 3.55 nos da una visión clara de los ajustes que se hicieron para el análisis de sensibilidad, se reconocen 10 escenarios con una red básica y 10 escenarios con una red de doble capa y una neurona de salida, por lo que cada escenario representa una configuración específica de parámetros o condiciones, en total son 20 combinaciones diferentes que representa una instancia única de los parámetros o condiciones que se exploran en el estudio. Los resultados se registraron para su crítica y posterior análisis.

CAPAS	Escenario	MAE	MAPE	
1+1	1	0.0648	68.9365	Red 1+1 con Adam 0.01 - Epochs 300 y batch size 120
	2	0.0672	61.7969	Red 1+1 con Adam 0.01 - Epochs 200 y batch size 55
	3	0.0663	65.3848	Red 1+1 con Adam 0.01 - Epochs 500 y batch size 250
	4	0.0633	48.4952	Red 1+1 con Adam 0.01 - Epochs 180 y batch size 300
	5	0.0512	38.43	Red 1+1 con Adam 0.011 - Epochs 180 y batch size 300
	6	0.0651	84.5352	Red 1+1 con Adam 0.011 - Epochs 180 y batch size 320
	7	0.0587	46.2075	Red 1+1 con Adam 0.012 - Epochs 180 y batch size 300
	8	0.0523	48.4454	Red 1+1 con Adam 0.011 - Epochs 80 y batch size 300
	9	0.0589	55.254	Red 1+1 con Adam 0.011 - Epochs 100 y batch size 300
	10	0.0561	58.5486	Red 1+1 con Adam 0.011 - Epochs 180 y batch size 100
CAPAS	Escenario	MAE	MAPE	
2+1	11	0.0634	43.4756	RED 2+1 con dropout 0.4 + BatchNormalization(momentum=0.4) - Adam 0.011 - Epochs 580 y batch size 380
	12	0.0743	54.8543	RED 2+1 con dropout 0.4 + BatchNormalization(momentum=0.4) - Adam 0.001 - Epochs 600 y batch size 380
	13	0.0565	41.22	RED 2+1 con dropout 0.4 + BatchNormalization(momentum=0.4) - Adam 0.011 - Epochs 800 y batch size 400
	14	0.0688	69.3303	RED 2+1 con dropout 0.3 + BatchNormalization(momentum=0.3) - Adam 0.011 - Epochs 800 y batch size 400
	15	0.0596	51.8946	RED 2+1 con dropout 0.3 + BatchNormalization(momentum=0.3) - Adam 0.011 - Epochs 1,000 y batch size 400
	16	0.0838	69.3173	RED 2+1 con dropout 0.4 + BatchNormalization(momentum=0.3) - Adam 0.011 - Epochs 1,000 y batch size 400
	17	0.0542	45.584	RED 2+1 con dropout 0.3 + BatchNormalization(momentum=0.4) - Adam 0.011 - Epochs 1,000 y batch size 400
	18	0.0646	46.6838	RED 2+1 con dropout 0.3 + BatchNormalization(momentum=0.4) - Adam 0.011 - Epochs 1,100 y batch size 420
	19	0.0596	51.8946	RED 2+1 con dropout 0.3 + BatchNormalization(momentum=0.3) - Adam 0.012 - Epochs 1,000 y batch size 400
	20	0.0774	50.441	RED 2+1 con dropout 0.3 + BatchNormalization(momentum=0.3) - Adam 0.011 - Epochs 1,200 y batch size 450

Figura 3.55 Descripción de las arquitecturas y escenarios del análisis de sensibilidad.

Comienza con los modelos de “1 + 1” y después los modelos “2 + 1”, (véase Tabla 3.1 a **Tabla 3.4**) del lado izquierdo se visualizan las gráficas de los valores reales en color azul y se comparan con los valores pronosticados por la configuración de hiperparámetros experimentada en esa corrida del modelo. Observaremos que se encuentran a su derecha de todas las gráficas la información útil de sus ajustes en el análisis de sensibilidad para tener un registro de sus métricas MAE y MAPE.

Tabla 3.1. Tabla de los escenarios del 1 al 5 en el análisis de sensibilidad

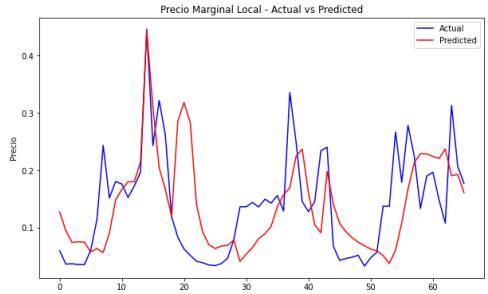
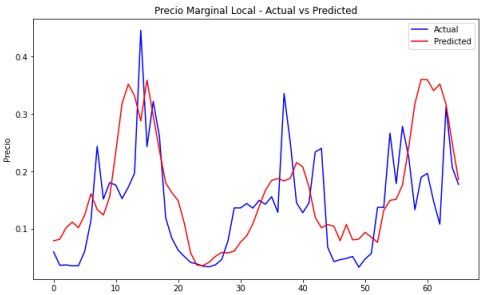
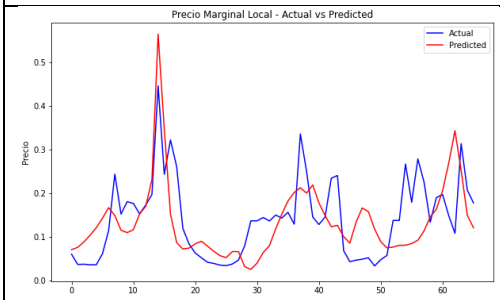
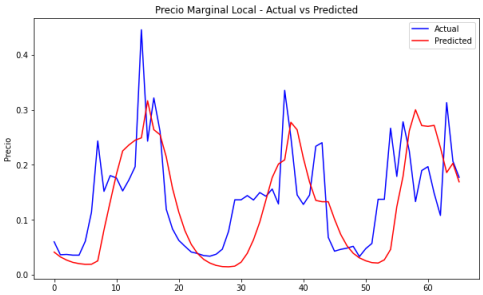
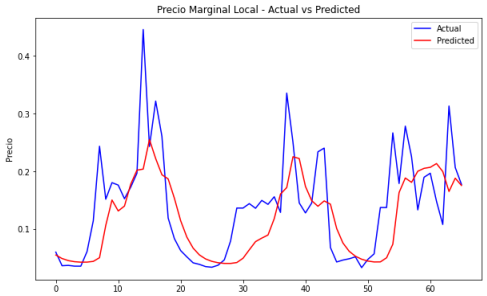
GRÁFICA DEL MODELO:	DESCRIPCIÓN:
	Escenario 1: Red 1+1 con *Adam* 0.01 - *Epochs* 300 y *batch size* 120 MAE: 0.0648 y MAPE: 68.9365
	Escenario 2: Red 1+1 con *Adam* 0.01 - *Epochs* 200 y *batch size* 55 MAE: 0.0672 y MAPE: 61.7969
	Escenario 3: Red 1+1 con *Adam* 0.01 - *Epochs* 500 y *batch size* 250 MAE: 0.0663 y MAPE: 65.3848
	Escenario 4: Red 1+1 con *Adam* 0.01 - *Epochs* 180 y *batch size* 300 MAE: 0.0633 y MAPE: 48.4952
	Escenario 5: Red 1+1 con *Adam* 0.011 - *Epochs* 180 y *batch size* 300 MAE: 0.0512 y MAPE: 38.43

Tabla 3.2. Tabla de los escenarios del 6 al 10 en el análisis de sensibilidad

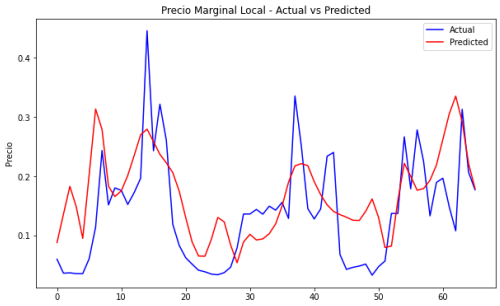
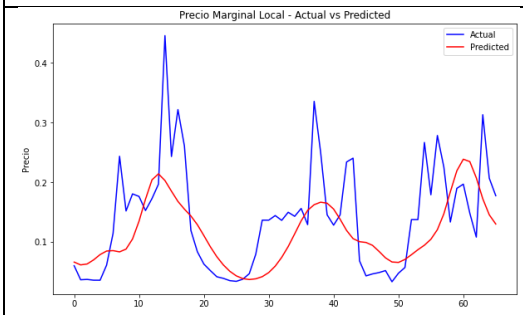
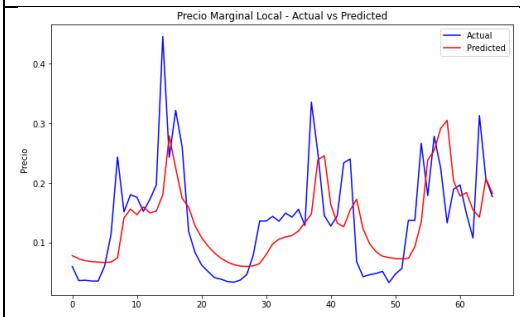
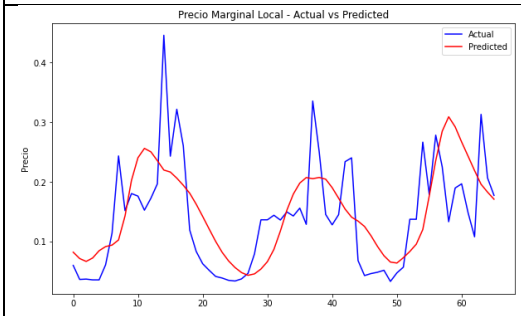
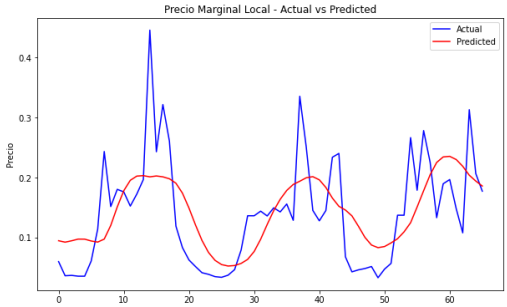
GRÁFICA DEL MODELO:	DESCRIPCIÓN:
	Escenario 6: Red 1+1 con *Adam* 0.011 - *Epochs* 180 y *batch size* 320 MAE: 0.0651 y MAPE: 84.5352
	Escenario 7: Red 1+1 con *Adam* 0.012 - *Epochs* 180 y *batch size* 300 MAE: 0.0587 y MAPE: 46.2075
	Escenario 8: Red 1+1 con *Adam* 0.011 - *Epochs* 80 y *batch size* 300 MAE: 0.0523 y MAPE: 48.4454
	Escenario 9: Red 1+1 con *Adam* 0.011 - *Epochs* 100 y *batch size* 300 MAE: 0.0589 y MAPE: 55.254
	Escenario 10: Red 1+1 con *Adam* 0.011 - *Epochs* 180 y *batch size* 100 MAE: 0.0561 y MAPE: 58.5486

Tabla 3.3. Tabla de los escenarios del 11 al 15 en el análisis de sensibilidad.

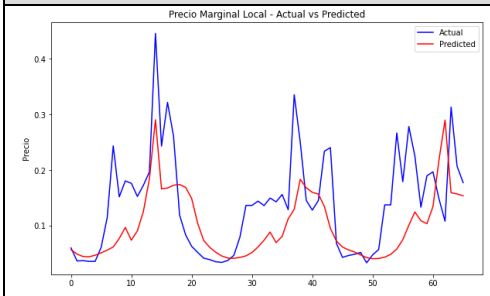
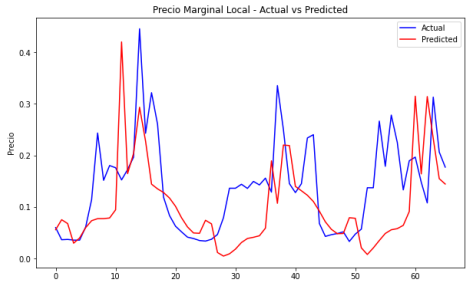
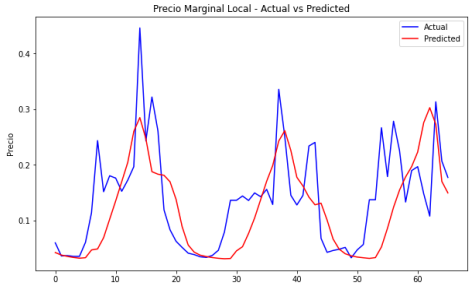
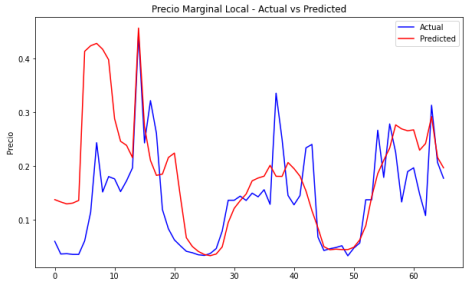
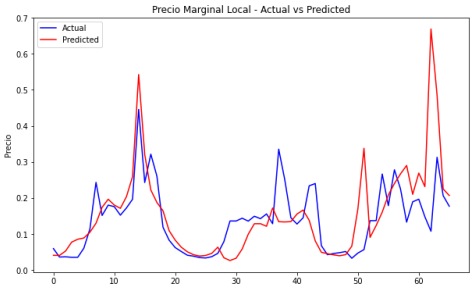
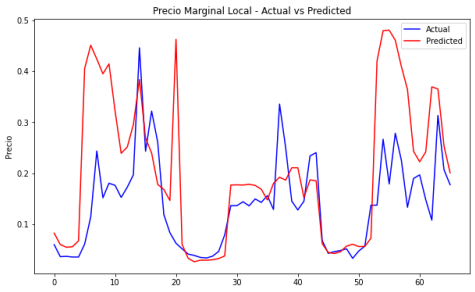
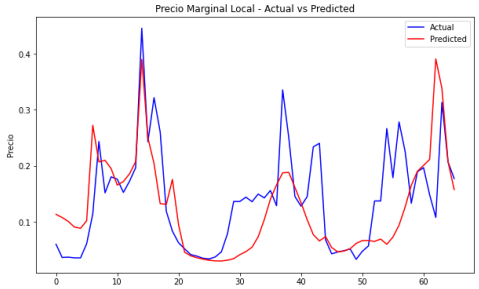
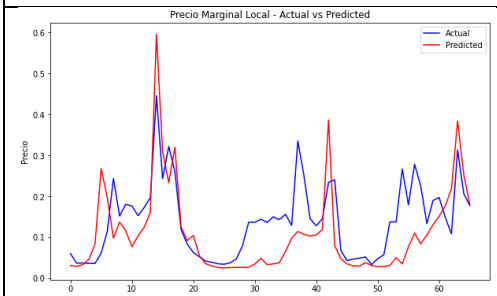
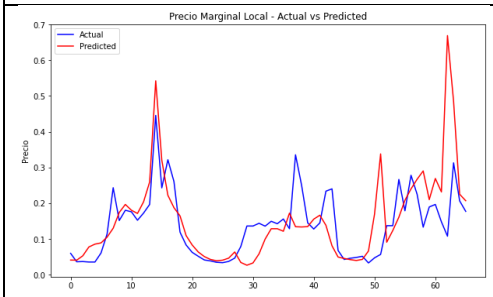
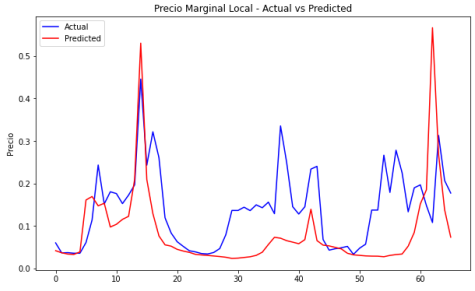
GRÁFICA DEL MODELO:	DESCRIPCIÓN:
	Escenario 11: RED 2+1 con *dropout* 0.4 + *BatchNormalization* (momentum=0.4) - *Adam* 0.011 - *Epochs* 580 y *batch size* 380 MAE: 0.0634 y MAPE: 43.4756
	Escenario 12: RED 2+1 con *dropout* 0.4 + *BatchNormalization* (momentum=0.4) - *Adam* 0.001 - *Epochs* 600 y *batch size* 380 MAE: 0.0743 y MAPE: 54.8543
	Escenario 13: RED 2+1 con *dropout* 0.4 + *BatchNormalization* (momentum=0.4) - *Adam* 0.011 - *Epochs* 800 y *batch size* 400 MAE: 0.0565 y MAPE: 41.22
	Escenario 14: RED 2+1 con *dropout* 0.3 + *BatchNormalization* (momentum=0.3) - *Adam* 0.011 - *Epochs* 800 y *batch size* 400 MAE: 0.0688 y MAPE: 69.3303
	Escenario 15: RED 2+1 con *dropout* 0.3 + *BatchNormalization* (momentum=0.3) - *Adam* 0.011 - *Epochs* 1,000 y *batch size* 400 MAE: 0.0596 y MAPE: 51.8946

Tabla 3.4. Tabla de los escenarios del 16 al 20 en el análisis de sensibilidad.

GRÁFICA DEL MODELO:	DESCRIPCIÓN:
	Escenario 16: RED 2+1 con *dropout* 0.4 + *BatchNormalization* (momentum=0.3) - *Adam* 0.011 - *Epochs* 1,000 y *batch size* 400 MAE: 0.0838 y MAPE: 69.3173
	Escenario 17: RED 2+1 con *dropout* 0.3 + *BatchNormalization* (momentum=0.4) - *Adam* 0.011 - *Epochs* 1,000 y *batch size* 400 MAE: 0.0542 y MAPE: 45.584
	Escenario 18: RED 2+1 con *dropout* 0.3 + *BatchNormalization* (momentum=0.4) - *Adam* 0.011 - *Epochs* 1,100 y *batch size* 420 MAE: 0.0646 y MAPE: 46.6838
	Escenario 19: RED 2+1 con *dropout* 0.3 + *BatchNormalization* (momentum=0.3) - *Adam* 0.011 - *Epochs* 1,000 y *batch size* 400 MAE: 0.0596 y MAPE: 51.8946
	Escenario 20: RED 2+1 con *dropout* 0.3 + *BatchNormalization* (momentum=0.3) - *Adam* 0.011 - *Epochs* 1,200 y *batch size* 450 MAE: 0.0774 y MAPE: 50.441

El resultado del análisis de sensibilidad se contempla en la Figura 3.56 que reúne a los 20 escenarios para verificar las diferencias entre ellas y como se generalizaban con los datos reales representados en color negro en el gráfico. Sin embargo, para una mejor apreciación se mostrarán los escenarios en dos grupos, aquellos pertenecientes a la red “1+1” en la Figura 3.57 y aquellos al grupo de la red “2+1” a la Figura 3.58.

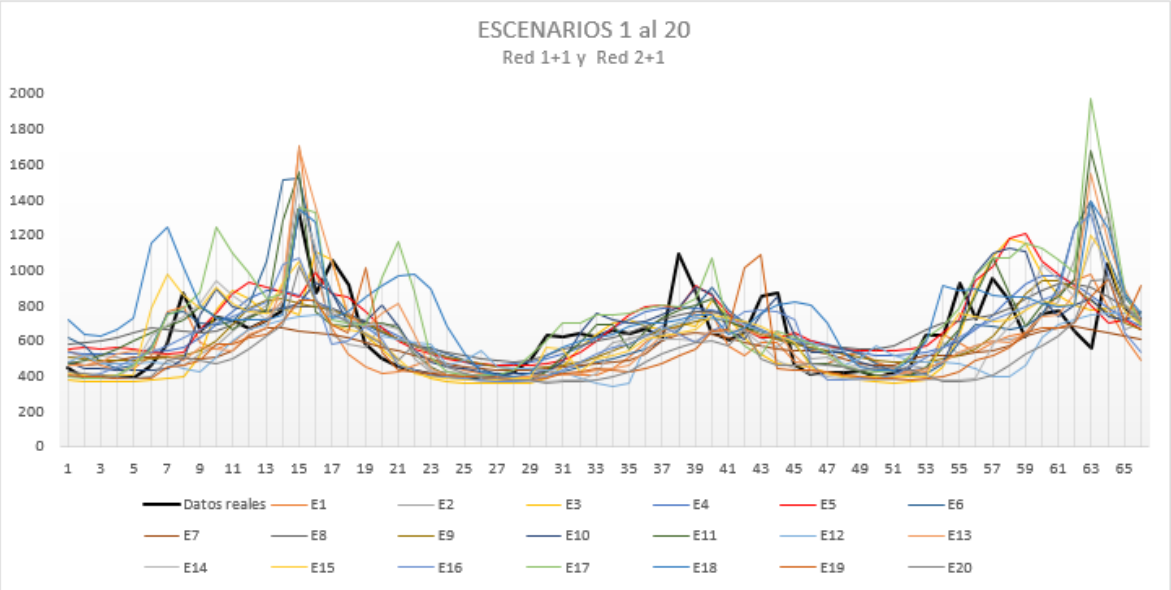


Figura 3.56 Gráfica de todos los escenarios del Análisis de Sensibilidad.

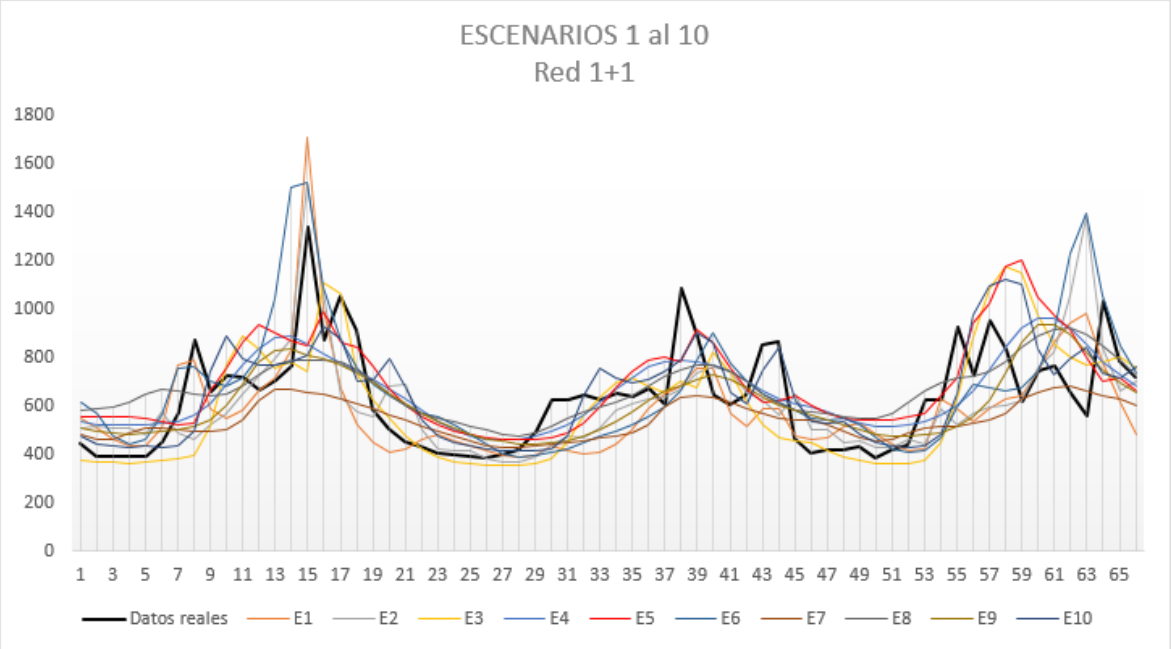


Figura 3.57 Gráfica del Análisis de sensibilidad de la red 1+1.

La Figura 3.57 nos permite ver con más claridad aquellos pronósticos correspondientes a la red “1+1” donde ningún escenario se acerca lo satisfactoriamente a generalizar el comportamiento de los datos, pero cabe reconocer que los escenarios más similares son el 5 y 6 respectivamente. Los valores reales se muestran en color negro, y aunque todos los escenarios presentan cierto grado de similitud con los reales, es importante complementar esta observación general con las métricas detalladas proporcionadas en la tabla de la Figura 3.59

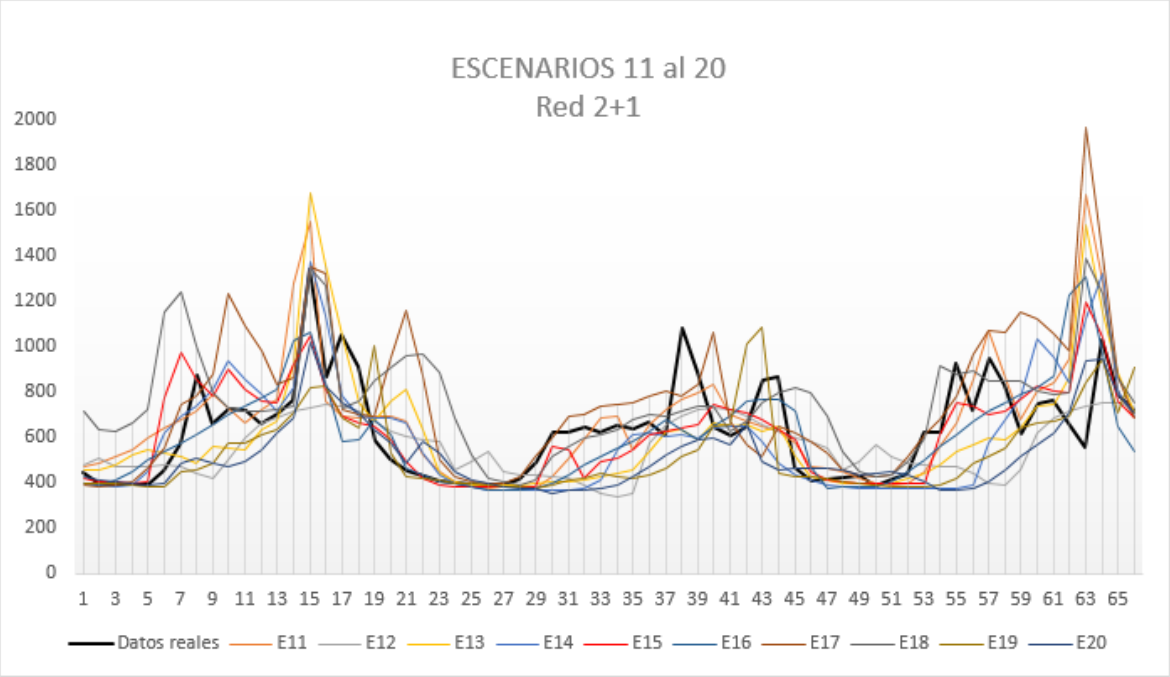


Figura 3.58 Gráfica del Análisis de Sensibilidad de la red 2+1.

Por otro lado, al examinar la red de 2 capas más 1 (2+1), cuya representación gráfica se muestra en la Figura 3.58, se observa que el escenario que más se asemeja a los valores reales (representados en color negro) es el escenario 15 (en color rojo). Sin embargo, este escenario muestra un notable desfase y una eficiencia limitada a simple vista. En comparación con la red de “1+1”, se evidencia que la red “2+1” es aún menos eficiente. El análisis de sensibilidad, por otro lado, aporta principalmente el rango de los hiperparámetros en el que la red se desempeña bien, ya que la extrapolación de cada hiperparámetro a valores fuera de los experimentados en los escenarios solo deteriora la capacidad de la red para replicar el patrón de comportamiento horario del Precio Marginal Local (PML).

A continuación, se reúne las métricas MAE y MAPE en la Figura 3.59 de los resultados del análisis de sensibilidad:

ANÁLISIS DE SENSIBILIDAD

CAPAS	ESCENARIO	MAE	MAPE
1+1	1	0.0648	68.9365
	2	0.0672	61.7969
	3	0.0663	65.3848
	4	0.0633	48.4952
	5	0.0512	38.43
	6	0.0651	84.5352
	7	0.0587	46.2075
	8	0.0523	48.4454
	9	0.0589	55.254
	10	0.0561	58.5486
CAPAS	ESCENARIO	MAE	MAPE
2+1	11	0.0634	43.4756
	12	0.0743	54.8543
	13	0.0565	41.22
	14	0.0688	69.3303
	15	0.0596	51.8946
	16	0.0838	69.3173
	17	0.0542	45.584
	18	0.0646	46.6838
	19	0.0596	51.8946
	20	0.0774	50.441

Figura 3.59 Métricas obtenidas durante el análisis de sensibilidad.

La Figura 3.59 presenta un análisis de sensibilidad en el que se comparan 20 escenarios diferentes de la red LSTM y compararlas en función de dos métricas: el Error Absoluto Medio (MAE) y el Porcentaje de Error Absoluto Medio (MAPE). Aquí hay un resumen basado en los datos obtenidos a 20 escenarios de prueba:

Rangos de Valores:

MAE: El mínimo del MAE es 0.0512 (escenario 5) y el máximo es 0.0838 (escenario 16).

MAPE: El mínimo del MAPE es 38.43 (escenario 5) y el máximo es 84.535 (escenario 6).

Mejor desempeño:

El escenario 5 tiene el MAE más bajo (0.0512) y el MAPE más bajo (38.43), lo que indica el mejor desempeño general en ambas métricas. Su grafico muestra una generalización aceptable de sus pronósticos frente a los datos reales.

Peor desempeño:

El escenario 16 tiene el MAE más alto (0.0838). y el segundo peor es el escenario 6 tiene el MAPE más alto (84.5352), lo que indica el peor desempeño en términos de error porcentual.

En general, la elección de la arquitectura de la red LSTM dependerá si se valora más la precisión absoluta (MAE), entonces se podría preferir modelos como el modelo 5. Si la prioridad es reducir el error porcentual (MAPE), entonces el modelo 5 también es preferible, ya que tiene el MAPE más bajo. La red "1+1", especialmente en el escenario 5 con un MAE de 0.0512 y un MAPE de 38.43, destaca como una opción sólida y con un buen equilibrio entre precisión y generalización. Las configuraciones más complejas de la red "2+1", aunque presentan ciertos puntos de rendimiento destacados, parecen no ofrecer una mejora significativa en comparación con la red "1+1", y en algunos casos, muestran un riesgo de sobreajuste, como se evidencia en el modelo 16 con un MAPE de 69.3173.

Épocas (*Epochs*): Se observa que el rendimiento de la red es sensible a la duración de las épocas. Aunque existe variabilidad, valores en el rango de 180 a 300 épocas parecen proporcionar un buen equilibrio entre rapidez de entrenamiento y capacidad de generalización. El escenario 5, con 180 épocas, destaca como una configuración efectiva.

Tasa de Aprendizaje (*Adam*): La tasa de aprendizaje de 0.011, como se utilizó en el escenario 5, demostró ser adecuada para lograr un rendimiento óptimo. Este valor facilita el proceso de convergencia durante el entrenamiento, resultando en predicciones precisas y generalizadas.

Tamaño del Lote (*Batch Size*): El tamaño del lote también influye en el rendimiento de la red. Se observa que tamaños de lote alrededor de 300 a 400 proporcionan buenos resultados en términos de MAE y MAPE. Nuevamente, el escenario 5, con un tamaño de lote de 300, muestra un rendimiento destacado.

La elección apropiada sería una red tipo "1+1", con épocas de 180 a 300, la tasa *Adam* en 0.011 y el *batch size* en 300, más sin embargo el *batch size* es un valor que hace lento al modelo y esta susceptible a reducirse dada la eficiencia de tiempo en que trabaje el modelo.

3.9.4 Búsqueda en cuadricula:

Una buena práctica de afinación del modelo dentro del machine Learning es también aplicar una *"search grid"* para que el código realice una búsqueda automatizada de los mejores hiperparámetros. Para el modelo LSTM secuencial con Prophet, aplicamos los mejores resultados de la búsqueda iterativa manual y la búsqueda en cuadricula automática se realiza ahora a los hiperparámetros de Prophet a experimentar y comparar; véase Figura 3.60.

```
from sklearn.model_selection import ParameterGrid

# Definir parámetros para la búsqueda en cuadrícula
param_grid = {
    'changepoint_prior_scale': [0.001, 0.01, 0.1, 0.5],
    'seasonality_prior_scale': [0.01, 0.1, 1.0, 10.0],
    'holidays_prior_scale': [0.01, 0.1, 1.0, 10.0]
}

# Crear una cuadrícula de parámetros
grid = ParameterGrid(param_grid)

best_model = None
best_mape = float('inf')  # inicializar con infinito para luego minimizar

# Búsqueda en la cuadrícula
for params in grid:
    temp_model = Prophet(**params, yearly_seasonality=False, daily_seasonality=True)
    temp_model.fit(df)

    future = temp_model.make_future_dataframe(periods=48, freq='H')
    forecast = temp_model.predict(future)

    y_pred = forecast['yhat'].tail(48).values
    mape_temp = np.mean(np.abs((y_real - y_pred) / y_real)) * 100

    if mape_temp < best_mape:
        best_mape = mape_temp
        best_model = temp_model

print(f"Mejor MAPE: {best_mape}")
```

Figura 3.60 Búsqueda cuadricula de hiperparámetros "alta demanda" "festivos" y "estacionales" en LSTM + Prophet.

La Figura 3.60 es un bloque de código que define un conjunto de parámetros con diferentes valores para cada aspecto clave del LSTM secuencial con Prophet, el cual tiene los siguientes comandos:

- *"from sklearn.model_selection import ParameterGrid"*: Importa la función "*ParameterGrid"* que permite crear una cuadrícula de parámetros para realizar la búsqueda. Se definen tres hiperparámetros para optimizar: "*changepoint_prior_scale*, *seasonality_prior_scale* y *holidays_prior_scale"*. Para cada "*hiperparámetro"*, se han dado 4 valores posibles, lo que significa que hay 4×4×4=64 combinaciones posibles.

- *"changepoint_prior_scale"*: Este parámetro controla la flexibilidad del modelo para ajustarse a los cambios en la serie temporal, como las horas de alta demanda. Un valor bajo significa que el modelo será más flexible y se ajustará a los cambios más rápidamente, mientras que un valor alto hará que el modelo sea más suave y se ajuste menos a cambios locales. Los valores proporcionados para "changepoint_prior_scale" son [0.001, 0.01, 0.1, 0.5], representando niveles de importancia a los puntos de cambio.
- *"seasonality_prior_scale"*: Este parámetro controla la fuerza de la estacionalidad en el modelo. La estacionalidad se refiere a patrones recurrentes en la serie temporal, como variaciones diarias o estacionales. Los valores proporcionados para "*seasonality_prior_scale"* son [0.01, 0.1, 1.0, 10.0], lo que indica diferentes niveles de importancia asignados a la estacionalidad.
- *"holidays_prior_scale"*: Este parámetro controla la influencia de las vacaciones/festivos en el modelo. Las vacaciones/festivos pueden tener un impacto significativo en ciertos patrones de series temporales. Los valores proporcionados para *"holidays_prior_scale"* son [0.01, 0.1, 1.0, 10.0], lo que representa diferentes niveles de importancia asignados.
- Se utiliza "*ParameterGrid*" para generar todas las combinaciones posibles de los hiperparámetros definidos en "*param_grid. Grid*" es un generador que producirá cada conjunto de parámetros en sucesión.
- *"best_model"* se inicializa como "*None"* y servirá para almacenar el mejor modelo según el error MAPE. *"best_mape"* se inicializa como infinito para que cualquier valor que encontremos inicialmente sea menor.
- Iteración sobre la Cuadrícula: El bucle *"for params in grid*": itera sobre cada conjunto de hiperparámetros generado por la cuadrícula (*grid*). "*params*" toma un conjunto específico de valores para *"changepoint_prior_scale", "seasonality_prior_scale",* y "*holidays_prior_scale*" en cada iteración.
- Creación de un Modelo Temporal: Se crea un modelo Prophet temporal "*temp_model*" en cada iteración utilizando los hiperparámetros específicos de esa iteración. Además, se desactiva la estacionalidad anual "*yearly_seasonality=False*" y se activa la estacionalidad diaria *"daily_seasonality=True"*. El modelo temporal se ajusta a los datos de series temporales "*df*" mediante "*temp_model.fit(df)*".

- Generación de Datos Futuros para Predicción: Se crea un marco de datos futuro "*future*" que abarca 48 períodos futuros con una frecuencia horaria. Se utiliza el modelo temporal para predecir los valores futuros *"forecast = temp_model.predict(future)"*.
- Cálculo del MAPE: Se calcula el MAPE comparando las predicciones "*y_pred*" con los valores reales "*y_real*". "*mape_temp*" es la puntuación de MAPE para el conjunto actual de hiperparámetros.
- Comparación y actualización del mejor Modelo: Se compara el MAPE actual "*mape_temp*" con el mejor MAPE encontrado hasta el momento "*best_mape*". Si el MAPE actual es menor, se actualiza "*best_mape*" con este valor y "*best_model*" con el modelo temporal actual. Al final de la búsqueda en cuadrícula, se imprime el mejor MAPE encontrado entre todas las combinaciones de hiperparámetros.

3.10 Conclusión

En el capítulo 3, se explicó de manera sistemática y clara el proceso iterativo de la metodología. Desde el preprocesamiento de los datos hasta la afinación del modelo, asegurando de que los modelos pudieran capturar de manera eficiente los patrones de comportamiento presentes en nuestro conjunto de datos. Este paso esencial sentó las bases para desarrollar modelos de pronóstico robustos.

Después se mostraron los códigos de los modelos de pronóstico utilizados, detallando sus constituciones y presentando las métricas de desempeño, como la validación de series temporales y los errores de pronóstico. Este análisis comparativo nos permitió informar sobre las cualidades más importantes de cada modelo.

Finalmente se llegó al análisis del proceso de afinación al modelo LSTM + Prophet. Esto incluyó la implementación de la ingeniería de características, la validación complementaria con walkforward, y un análisis de sensibilidad destinado a identificar los rangos adecuados de ajuste en los hiperparámetros. Se termina este proceso iterativo con una búsqueda en cuadrícula utilizando *'ParameterGrid'* para el modelo LSTM con Prophet. Este enfoque ordenado y estructurado en cada etapa de la metodología ha sentado las bases para construir modelos de pronóstico confiables y efectivos, alineados con los objetivos de nuestra investigación.

CAPÍTULO IV. RESULTADOS

4.1 Introducción

A continuación, se observarán los resultados de una implementación distribuida a través de 28 nodos representativos de la república. El capítulo 4 comienza con los resultados obtenidos mediante la aplicación de árboles de decisión para regresión en cada nodo, lo que conduce a un análisis minucioso de cada uno. Durante este proceso, se logra la identificación de las variables más relevantes para la predicción, respaldando así la acertada elección de utilizar árboles de decisión en el análisis del Modelo de Aprendizaje Automático en este proyecto. Además, se presentan los resultados de las métricas de aprendizaje de la red, registrando las validaciones del modelo en cada nodo; este enfoque proporciona valiosas perspectivas para una implementación más amplia y escalable. Además, se presenta la aportación complementaria del modelo de pronóstico con una aplicación web que facilita la visualización de los PML mediante una interfaz gráfica dinámica para la selección de rangos de datos. Esta herramienta permite al usuario ejecutar los algoritmos de pronósticos mediante el código de Python. Finalmente se entregan los resultados del modelo durante la implementación y concentrando las métricas de los pronósticos para su crítica y análisis.

4.2 Implementación:

La implementación del modelo abarca una fase crucial donde se evalúa su rendimiento en diversos nodos de los sistemas de interconexión en toda la república, por ello, la empresa decidió una selección de 28 nodos que sirvan para esta sección de implementación, véase Figura 4.1.

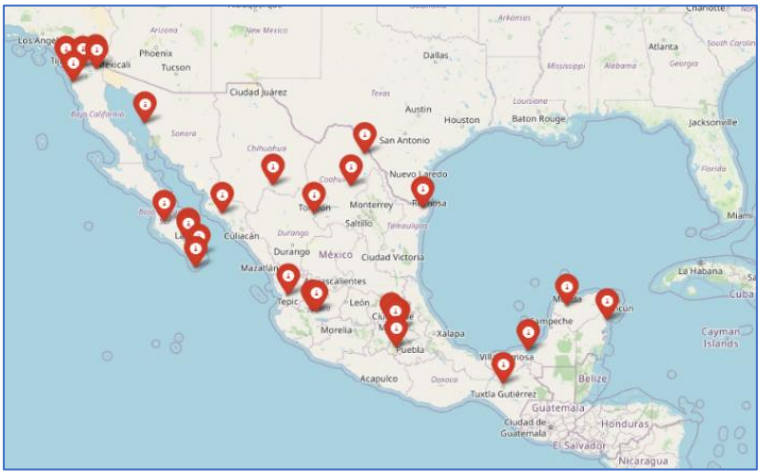


Figura 4.1 Localización geográfica de los nodos para la implementación del modelo.

				NODOP		
	SISTEMA	CENTRO DE CONTROL REGIONAL	ZONA DE CARGA	CLAVE	NOMBRE	NIVEL DE TENSIÓN (kV)
1	SIN	ORIENTAL	TUXTLA	02MMT-400	Manuel Moreno Torres	400
2	SIN	NORESTE	MONCLOVA	06MON-115	Monclova	115
3	SIN	CENTRAL	VDM NORTE	01TTH-230	Teotihuacan	230
4	SIN	PENINSULAR	CAMPECHE	08SLC-230	Santa Lucia	230
5	SIN	OCCIDENTAL	TEPIC VALLARTA	03AGM-400	Aguamilpa	400
6	SIN	PENINSULAR	MERIDA	08PTE-115	Poniente	115
7	BCA	BAJA CALIFORNIA	ENSENADA	07SAF-115	San Felipe	115
8	BCA	BAJA CALIFORNIA	MEXICALI	07CPD-230	Cerro Prieto Dos	230
9	BCA	BAJA CALIFORNIA	TIJUANA	07TCT-69	Tecate	69
10	BCA	BAJA CALIFORNIA	MEXICALI	07VPA-230	Valle de Puebla	230
11	BCS	BAJA CALIFORNIA SUR	CONSTITUCION	07LRO-115	Loreto	115
12	BCS	BAJA CALIFORNIA SUR	LOS CABOS	07CAD-115	Cabo San Lucas Dos	115
13	BCS	BAJA CALIFORNIA SUR	CONSTITUCION	07GAO-115	Central Termica Gral. Agustin Olachea	115
14	BCS	BAJA CALIFORNIA SUR	LA PAZ	07LPZ-115	La Paz	115
15	BCS	BAJA CALIFORNIA SUR	LOS CABOS	07SNT-115	Santiago	115
16	SIN	PENINSULAR	RIVIERA MAYA	08TUM-115	Tulum	115
17	SIN	NORESTE	PIEDRAS NEGRAS	06CBD-400	Carbon Dos	400
18	BCA	BAJA CALIFORNIA	MEXICALI	07RUM-69	Rumorosa	69
19	SIN	OCCIDENTAL	MINAS	03LVT-69	La Venta	69
20	SIN	NORTE	LAGUNA	05MVI-400	Cuadro de Maniobras Villa Nueva	400
21	SIN	NORTE	CAMARGO	05PAR-115	Parral	115
22	SIN	NOROESTE	GUASAVE	04GSV-115	Guasave	115
23	SIN	CENTRAL	CENTRO ORIENTE	01AFM-230	Aeropuerto Internacional Felipe Angeles Maniobras	230
24	SIN	CENTRAL	CENTRO ORIENTE	01ANL-85	Atotonilco	85
25	SIN	NOROESTE	CABORCA	04PLD-230	Puerto Libertad	230
26	SIN	NORESTE	MATAMOROS	06CEN-138	Central	138
27	SIN	OCCIDENTAL	GUADALAJARA	03AMX-69	Almex	69
28	SIN	ORIENTAL	MORELOS	02CBE-400	Ciclo Combinado Centro	400

Figura 4.2 Lista de los 28 nodos para la implementación del modelo.

En la tabla (Figura 4.2) presentada, se detallan los 28 nodos estratégicos seleccionados para la implementación del modelo de pronóstico en la red eléctrica nacional. Los sistemas BCA y BCS, representados por un distintivo color azul claro y azul estándar, respectivamente. Cada nodo está identificado por su sistema, centro de control regional, zona de carga, clave, nombre y nivel de tensión (kV). Esta selección abarca una diversidad geográfica y operativa, permitiendo observar el desempeño del modelo en distintos escenarios y condiciones de la red eléctrica a nivel nacional.

4.2.1 Los árboles de decisión para regresión.

Los árboles de decisión para regresión son modelos predictivos de valores numéricos por lo que el Precio Marginal Local (PML) puede ser tratado como un problema de regresión; cada nodo en el árbol representa una pregunta o una condición sobre las características de los datos, y cada rama representa una posible respuesta a esa pregunta. Las hojas del árbol contienen las predicciones finales.

La construcción del árbol implica dividir el conjunto de datos en subconjuntos más homogéneos en términos de la variable objetivo, seleccionando la característica y el valor de corte que maximizan la homogeneidad de los subconjuntos resultantes. El proceso se repite recursivamente hasta que se alcanza algún criterio de parada, ya sea condicionando la profundidad máxima del árbol o el número mínimo de muestras en una hoja. véase Figura 4.3:

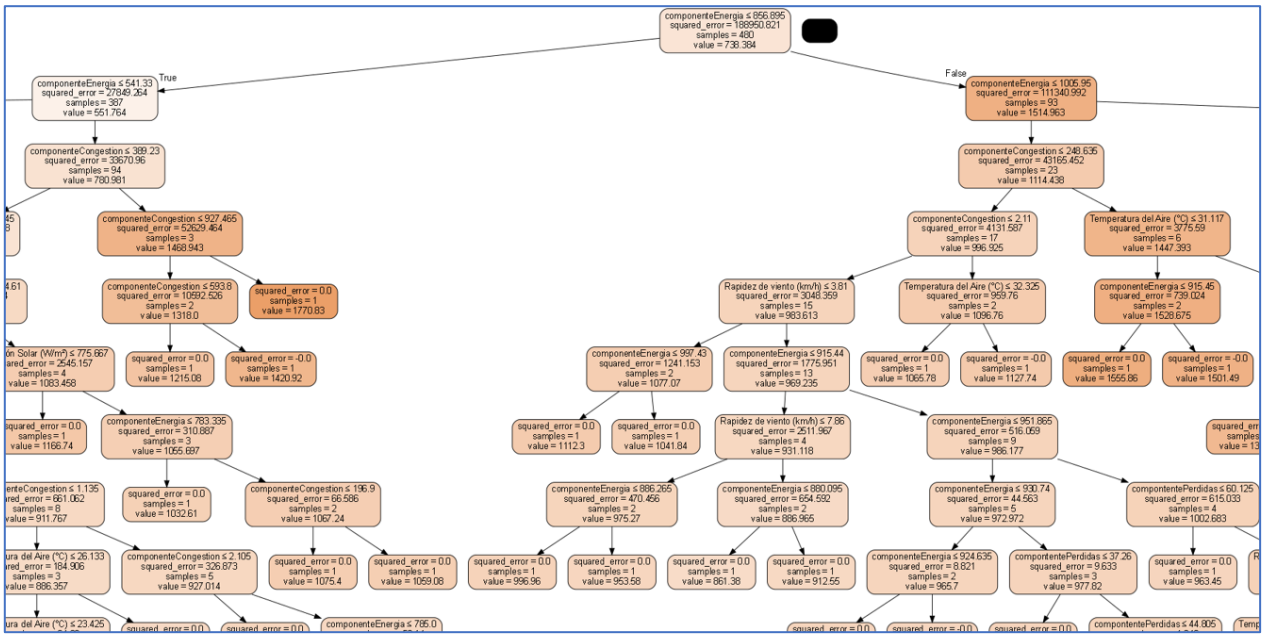


Figura 4.3 Diagrama de árbol del nodo 1 de la implementación del modelo.

En la Figura 4.3 el nodo inicial se conoce como el "nodo raíz" y tiene una profundidad de 0. Los nodos subsiguientes que se dividen a partir del nodo raíz forman el "nivel 1" y así sucesivamente. La profundidad de un nodo en un árbol de decisión se refiere a la distancia desde el nodo raíz hasta ese nodo específico. En un árbol de decisión para regresión, el nodo raíz se elige utilizando la característica (variable independiente) que mejor divide los datos, en términos de reducción del error cuadrático medio (MSE); la elección de la variable más importante se realiza mediante criterios como la ganancia de información; la reducción de Gini (en el caso de árboles de clasificación), o la reducción del MSE (en el caso de árboles de regresión). En cada nodo observamos como el error cuadrático medio (MSE) "*Squared_error*", el número de muestras "*Samples*" y el valor predicho "*Value*" se agregan a la variable a someter una condición. La primera condición, "Componente de energía <= 856.895", es la raíz del árbol. Luego, el árbol se bifurca en "*true*" o "*false*" dependiendo de si la condición es verdadera o falsa para una observación específica.

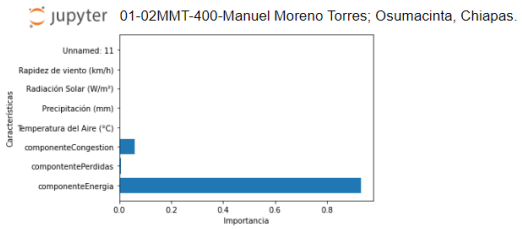
Para el nodo "true" (cuando " Componente de energía <= 856.895" es verdadero), se presenta la siguiente condición: " Componente de energía <= 541.33". Se muestra el error cuadrático medio, el número de muestras y el valor predicho asociado a este nodo. En el nodo "false", la condición para este nodo es " Componente de energía <= 1005.95". Al igual que en el nodo "true", se presentan métricas asociadas, como el error cuadrático medio "*Squared_error*", el número de muestras "*Samples*" y el valor predicho "*Value*".

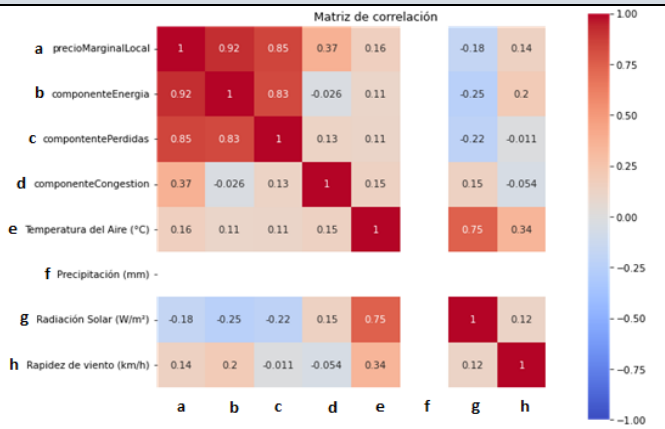
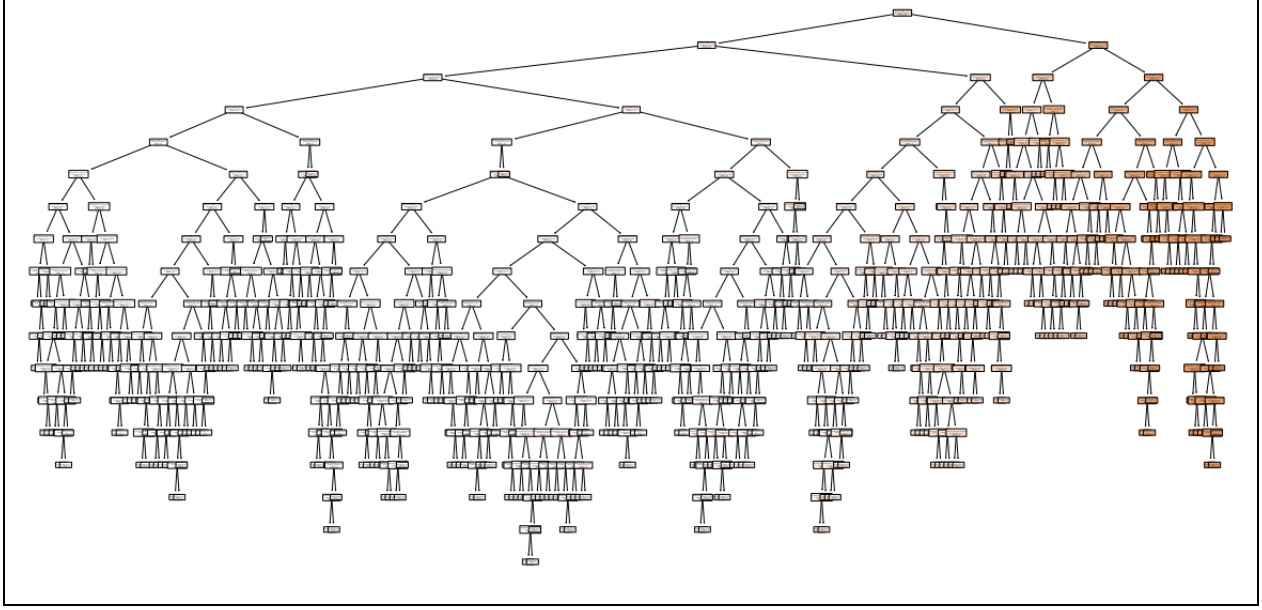
En este contexto, se observa que el nodo identificado como "false" (donde el valor de la variable "Componente de energía" es mayor a 1005.95) exhibe un error cuadrático medio más elevado, alcanzando un valor de 111,340.992. En contraste, el nodo "true" (donde el valor de "Componente de energía" es menor o igual a 541.33) presenta un *"Squared_error"* de 27,849.264. Esta diferencia en los errores cuadráticos medios sugiere que, para las observaciones que cumplen con la condición "Componente de energía > 1,005.95", el modelo no logra predecir con tanta precisión en comparación con las observaciones que cumplen con "Componente de energía <= 541.33". Además, el modelo resalta este nodo con un color distintivo en la Figura 4.3 debido a que alcanza rápidamente el criterio mínimo de error cuadrático medio en menos niveles por este camino.

En cada nivel o profundidad del árbol, se selecciona una variable y se establece un umbral (un valor numérico) para dividir el conjunto de datos en dos grupos, uno que cumple la condición (ramificación verdadera) y otro que no (ramificación falsa). Este proceso se repite en cada nodo, y la profundidad del árbol indica cuántas veces se ha realizado este proceso. Cuando se observa a los "nodos hojas" que las diferencias son bastante pequeñas, y en algunos casos, la métrica utilizada (en este caso, el error cuadrático) podría mostrar valores muy cercanos a cero. Esto indica que el modelo se ajustó muy bien a los datos de entrenamiento específicos en esas ramas. Cabe señalar que los datos pueden capturar ruido y patrones específicos que no se generalizan bien a nuevos datos. Por lo tanto, es importante evaluar el rendimiento del modelo en datos no vistos para obtener una evaluación más precisa de su capacidad predictiva. La solución es que la base de datos se subdivida en 3 conjuntos; entrenamiento, validación y prueba, que es como se recurrió a hacer con un 80%, 10% y 10% respectivamente y además se aplicó una validación con "*timeseriesplit*" donde los datos se fueron entrenando y validando, considerando el orden secuencial de los datos por ser una serie de tiempo.

Incorporar árboles de decisión en el código de pronóstico del Precio Marginal Local representa una contribución significativa a la investigación. Esta elección resulta estratégica y no se había abordado previamente en la literatura especializada. Su implementación ha permitido una comprensión más profunda y precisa del pronóstico, añadiendo una capa adicional de detalle que no se había explorado antes en este contexto. A continuación, veremos los nodos de energía en tablas con su diagrama de árbol, su grafica de variables importantes y su diagrama de correlación.

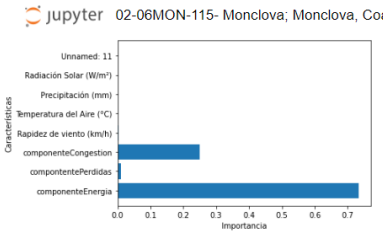
Tabla 4.1. Análisis de variables del nodo 1 (02MMT-400), con árbol de decisión y Matriz de correlación.

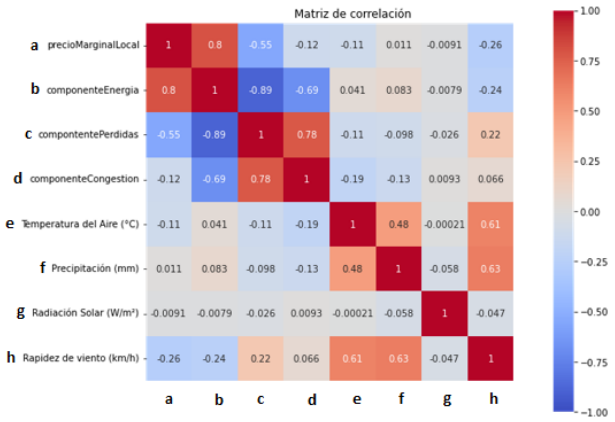
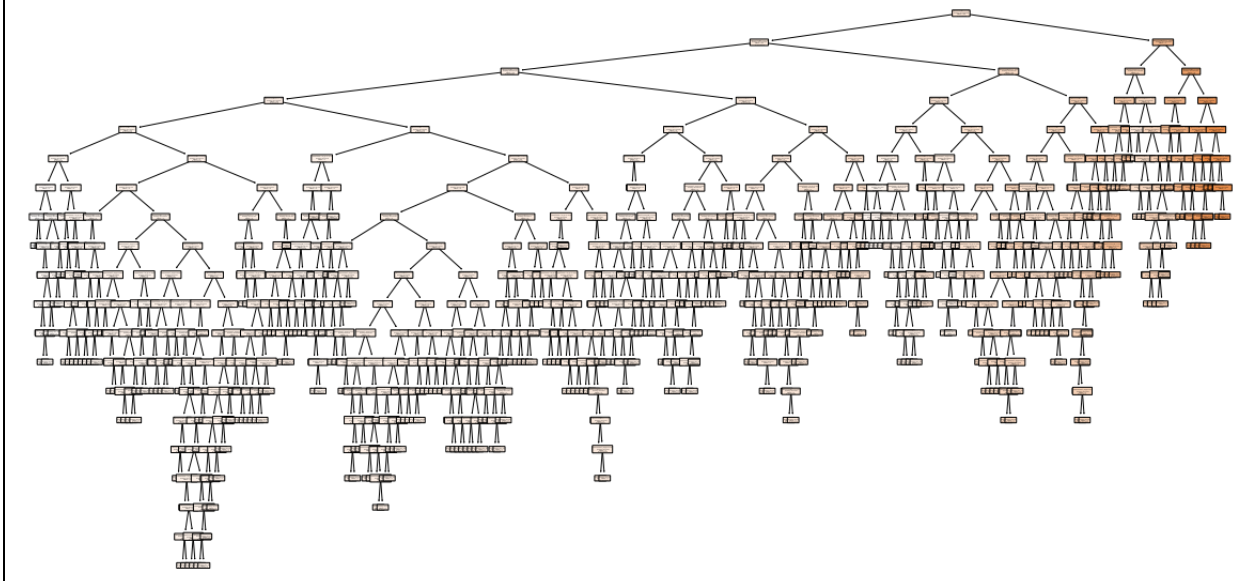
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 1: 02MMT-400 Manuel Moreno Torres; Usumacinta, Chiapas. (SIN)
 

Métricas:
MAE: 0.0770 y MAPE: 53.6317</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

En la Tabla 4.1podemos observar 3 gráficos importantes donde el grafico de barras es una visualización de las variables del nodo 02MMT-240 y el modelo determinó las variables importantes del nodo 02MMT-400; el componente de energía, el componente de congestión y en tercer lugar al componente de perdidas, pero es interesante que en la matriz de correlación observamos que el orden es componente de energía, perdidas y congestión respectivamente.

Tabla 4.2. Análisis de variables del nodo 2 (06MON-115), con árbol de decisión y Matriz de correlación.

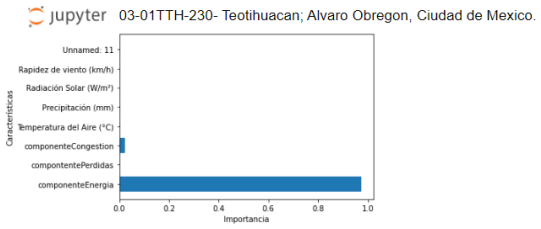
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 2: 06MON-115- Monclova; Monclova, Coahuila de Zaragoza. (SIN)
 

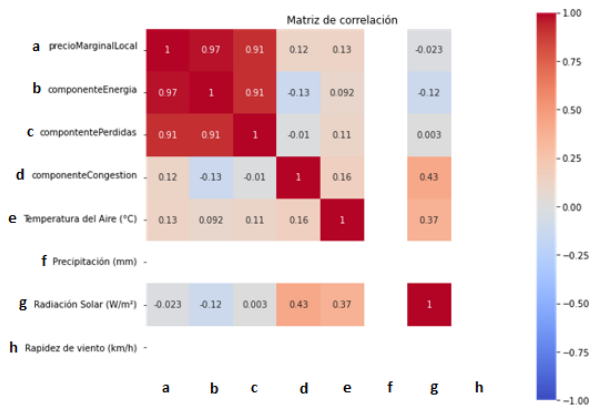
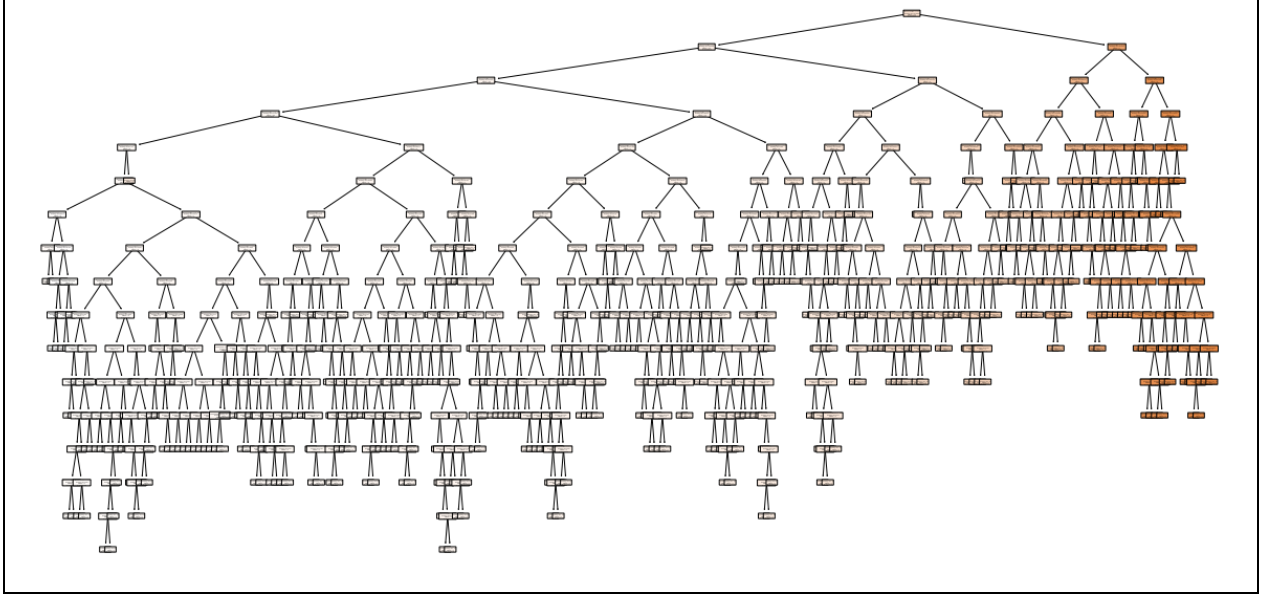
Métricas:
MAE: 0.0037 y MAPE: 76.6232</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.2 nos describe con 3 gráficos al nodo 2: 06MON-115- Monclova; Monclova, Coahuila de Zaragoza, que pertenece también al sistema interconectado nacional (SIN), y el

diagrama de importancia de las variables muestra al; "componente de energía" en 73% y "componente de congestión" en un 25%. Por lo que el componente de energía es el más importante en términos de mejorar el pronóstico del PML dentro del árbol de decisión para regresión y que puede confirmarse en la matriz de correlación también. También se muestran sus métricas de LSTM secuencial con: MAE: 0.0037 y MAPE: 76.6232.

Tabla 4.3. Análisis de variables del nodo 3 (01TTH-230), con árbol de decisión y Matriz de correlación.

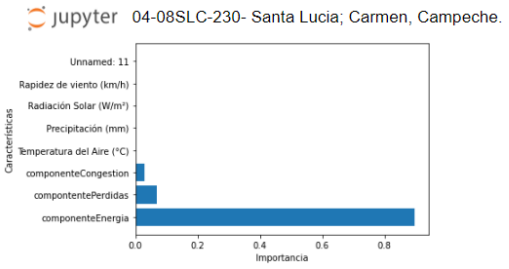
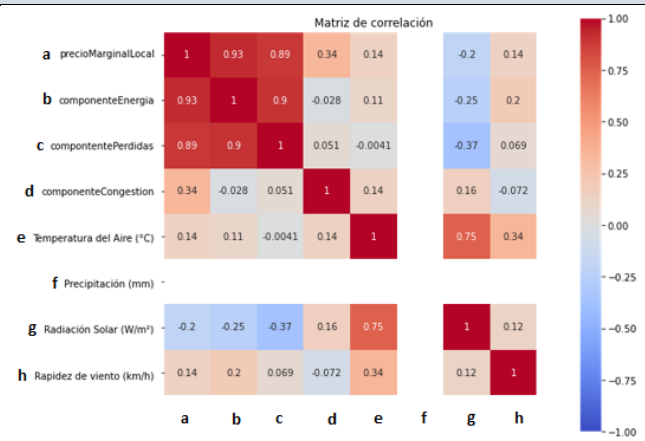
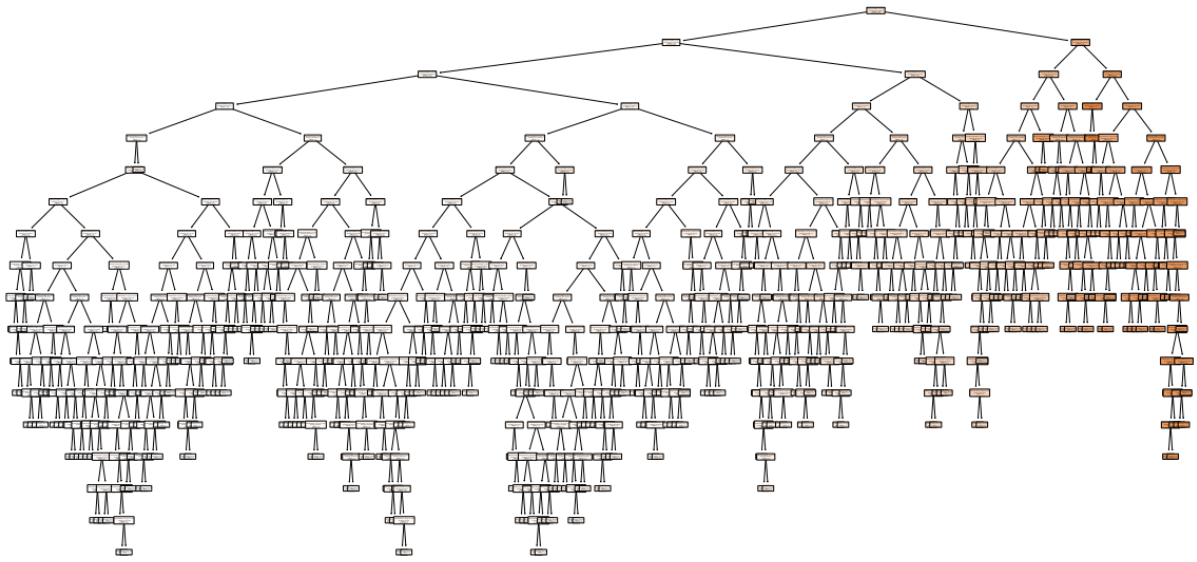
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 3: 01TTH-230- Teotihuacan; Álvaro Obregón, Ciudad de México. (SIN)
 

Métricas:
MAE: 0.0501 y MAPE: 43.6261</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.3 nos describe también con 3 gráficos al nodo 3: 01TTH-230- Teotihuacan;

Álvaro Obregón, Ciudad de México, y que pertenece también al SIN. Similarmente las variables; "componente de energía" en 96% y "componente de congestión" en un 8%, son las de interés para le pronósticos del PML. Las métricas en el modelo de LSTM secuencial tiene: MAE: 0.0501 y MAPE: 43.6261, además en la matriz de correlación resalta que conforme aumenta el componente de energía, aumentará el PML de este nodo, ósea que es directamente proporcional en un coeficiente de correlación de r= 0.97.

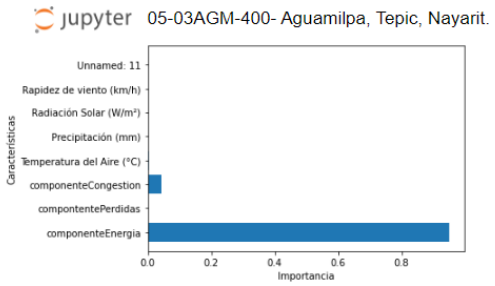
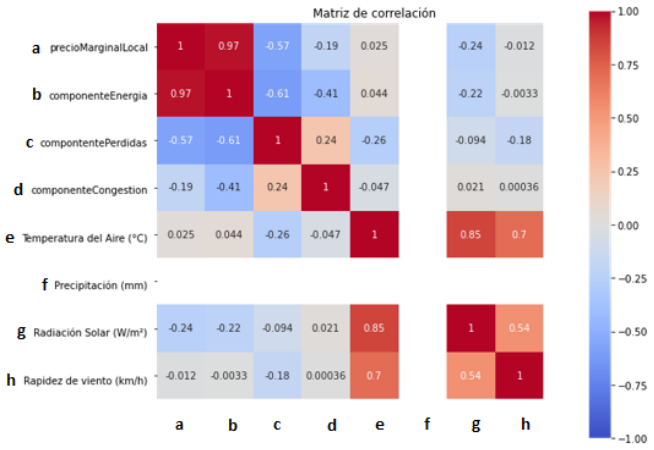
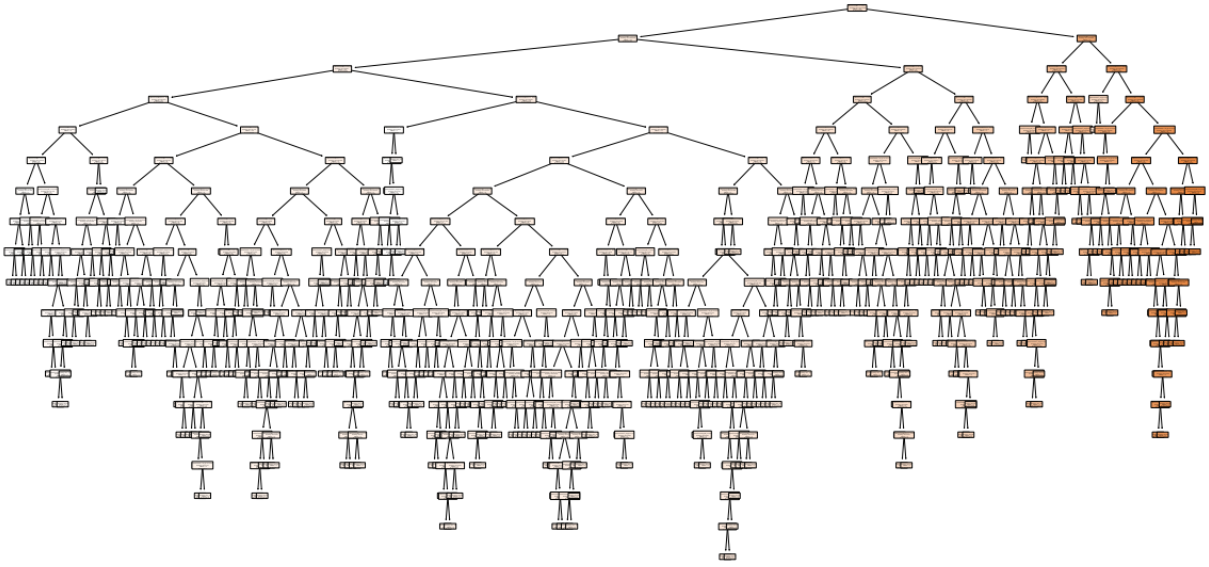
Tabla 4.4. Análisis de variables del nodo 4 (08SLC-230), con árbol de decisión y Matriz de correlación.

IMPORTANCIA DE VARIABLES:	MATRÍZ DE CORRELACIÓN:
NODO 4: 08SLC-230- Santa Lucia; Carmen, Campeche. (SIN)  Métricas: MAE: 0.066 y MAPE: 43.8836	
ÁRBOL DE DECISIÓN PARA REGRESIÓN:	
	

Ahora también en el análisis del nodo 4: 08SLC-230- Santa Lucia; Carmen, Campeche

del SIN su pronóstico del PML se midió con un MAE: 0.066 y MAPE: 43.8836, y que el árbol de decisión determinó que similar a los nodos anteriores tenemos que el componente de energía es el más importante y correlacionado con el PML. Véase Tabla 4.4

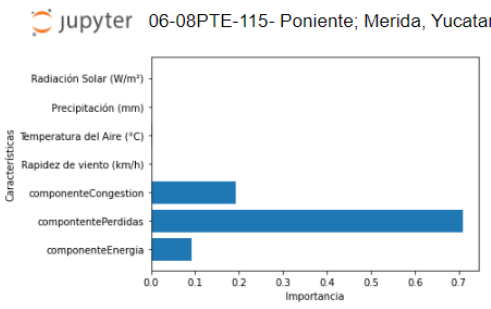
Tabla 4.5. Análisis de variables del nodo 5 (03AGM-400), con árbol de decisión y Matriz de correlación.

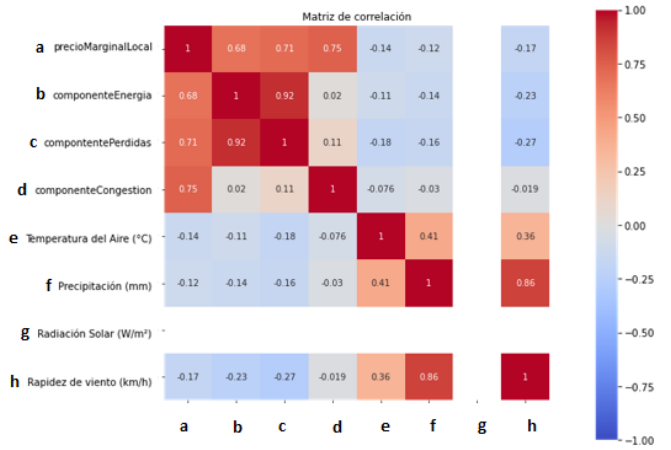
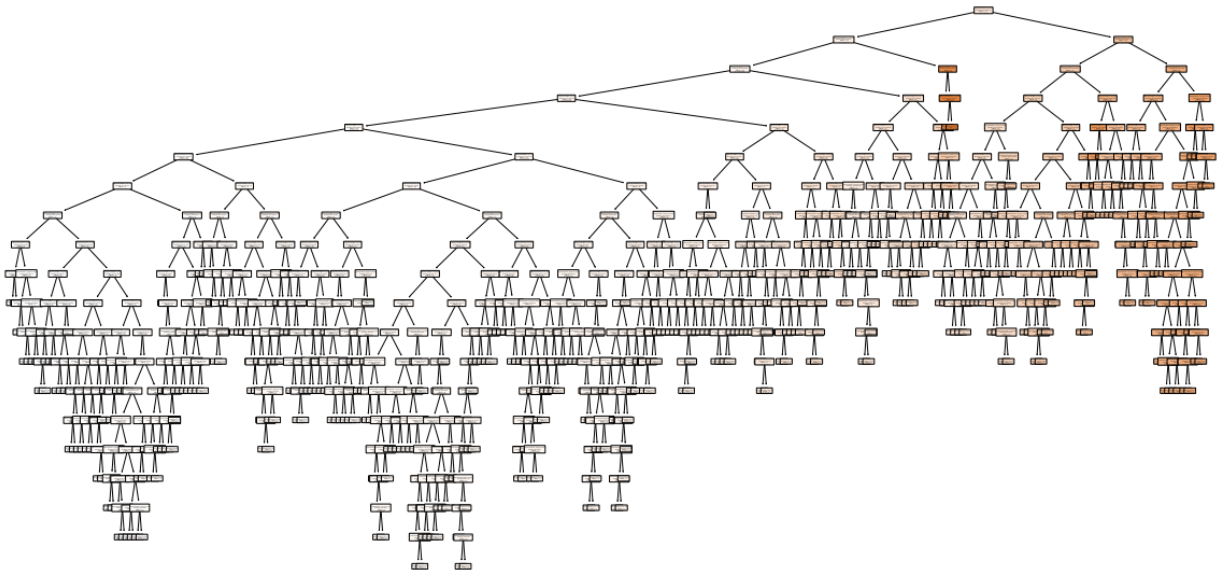
IMPORTANCIA DE VARIABLES:	MATRÍZ DE CORRELACIÓN:
NODO 5: 03AGM-400- Aguamilpa, Tepic, Nayarit. (SIN)  Métricas: MAE: 0.0617 y MAPE: 34.8548	
ÁRBOL DE DECISIÓN PARA REGRESIÓN:	
	

En la Tabla 4.5 observamos el nodo 5 de 03AGM-400- Aguamilpa, Tepic, Nayarit. (SIN) sus componentes importantes son el de energía y el de congestión. Pero vemos que el componente de congestión es inversamente proporcional al PML, lo que su selección sobre el

componente de perdidas tendría que ser comprobada en la práctica, porque puede ser que causen ruido en su integración en la red LSTM + Prophet que se desempeña con un MAE: 0.0617 y MAPE: 34.8548.

Tabla 4.6. Análisis de variables del nodo 6 (08PTE-115), con árbol de decisión y Matriz de correlación.

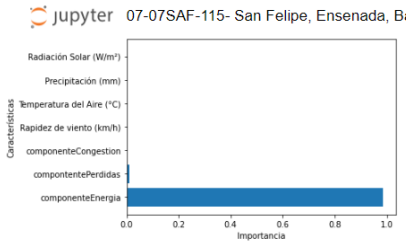
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 6: 08PTE-115- Poniente; Mérida, Yucatán. (SIN)
 

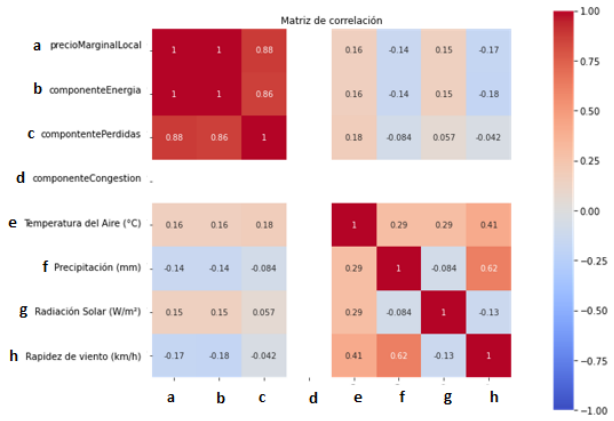
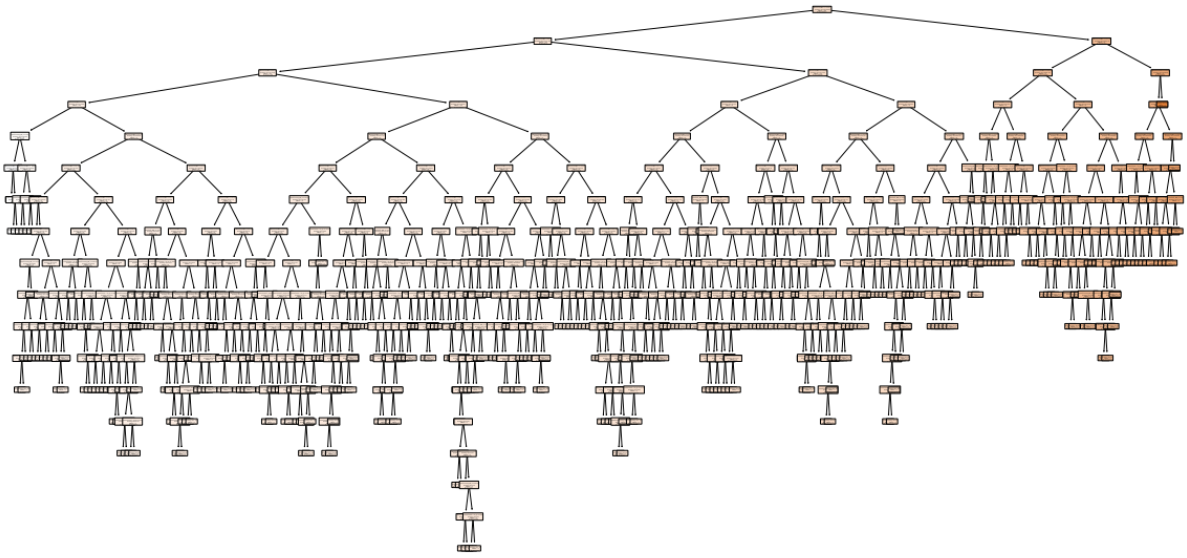
Métricas:
MAE: 0.0607 y MAPE: 103.2024</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.6 representa al nodo 6; 08PTE-115 en Poniente; Mérida, Yucatán, SIN. Observamos que no es igual a los nodos anteriores, aquí el componente de perdidas el más relevante porque ayuda al pronóstico del PML dentro del árbol (en sus nodos de decisión) y

también al componente de congestión con una importancia de 0.2 y al componente de energía en tercer lugar con un 0.09. El nodo es más complejo con el PML (véase la matriz de correlación) y esta podría ser una razón por la que el pronóstico no logra generalizar bien por lo que sus métricas son elevadas con MAE: 0.0607 y MAPE: 103.2024.

Tabla 4.7. Análisis de variables del nodo 7 (07SAF-115), con árbol de decisión y Matriz de correlación.

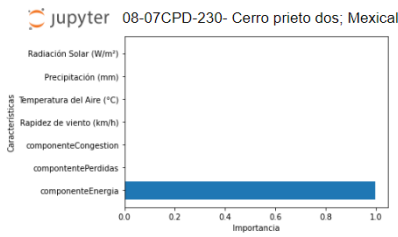
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 7: 07SAF-115- San Felipe, Ensenada, Baja California. (BCA)
 

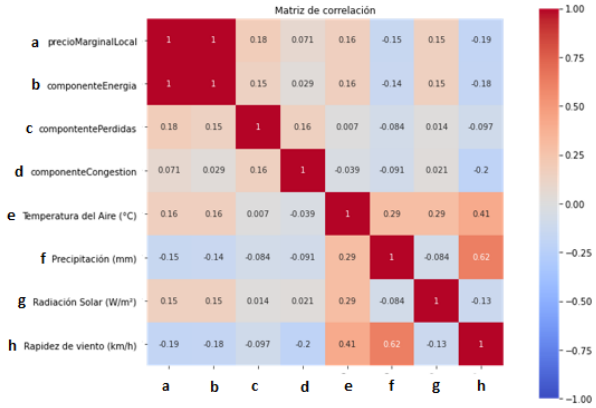
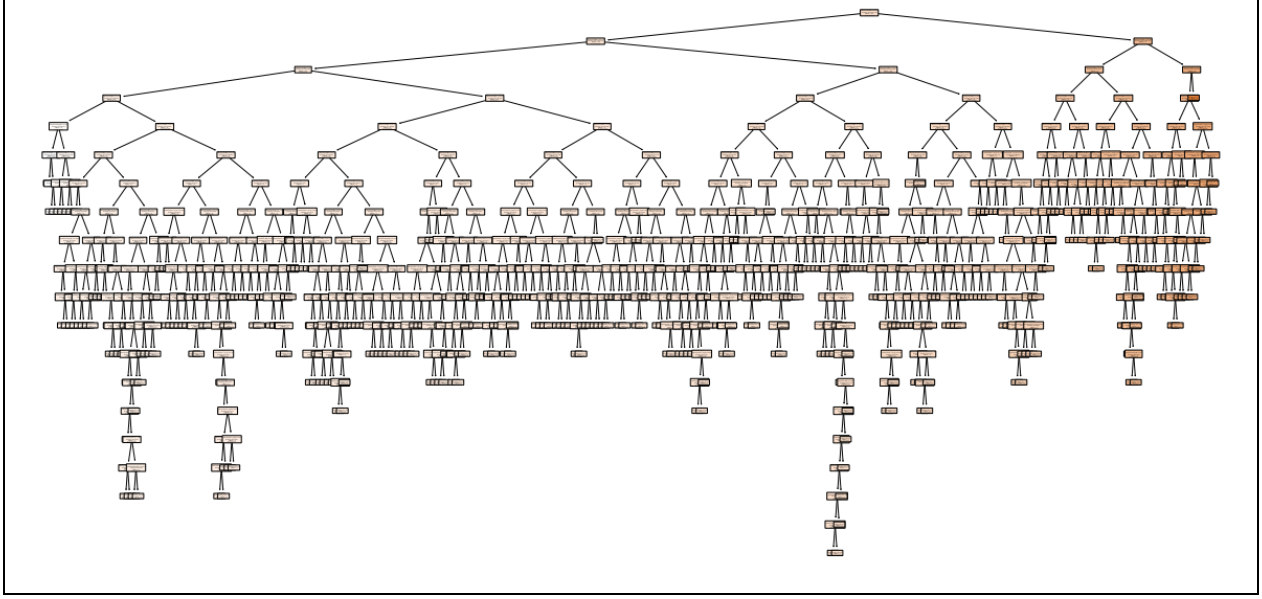
Métricas:
MAE: 0.0357 y MAPE: 101.6683</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.7 es el nodo 7: 07SAF-115- San Felipe, Ensenada, Baja California. (BCA) vemos que el componente de congestión tiene muchos ceros en sus registros lo que en la matriz de correlación la muestra como NaN indicando que su correlación resulta irrelevante y que

incluso se puede descartar. En el árbol determina que sea el componente de energía el importante, y quizás el componente de perdidas, lo que es respaldado con la matriz de correlación también.

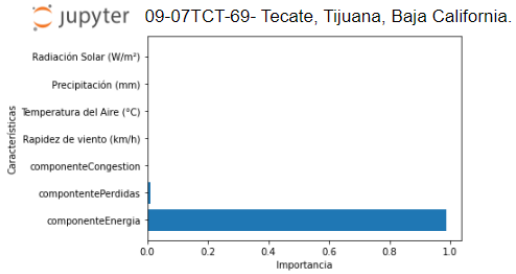
Tabla 4.8. Análisis de variables del nodo 8 (07CPD-230), con árbol de decisión y Matriz de correlación.

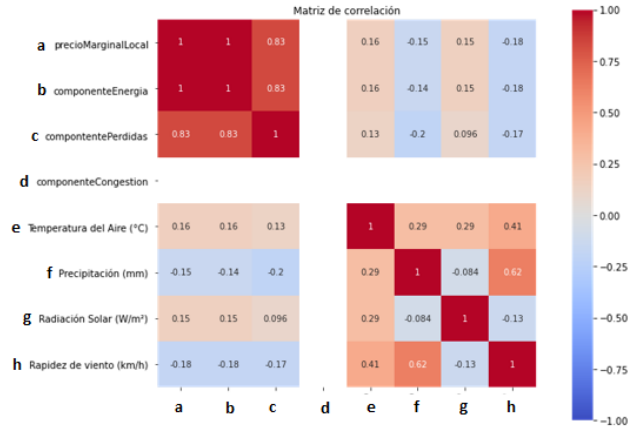
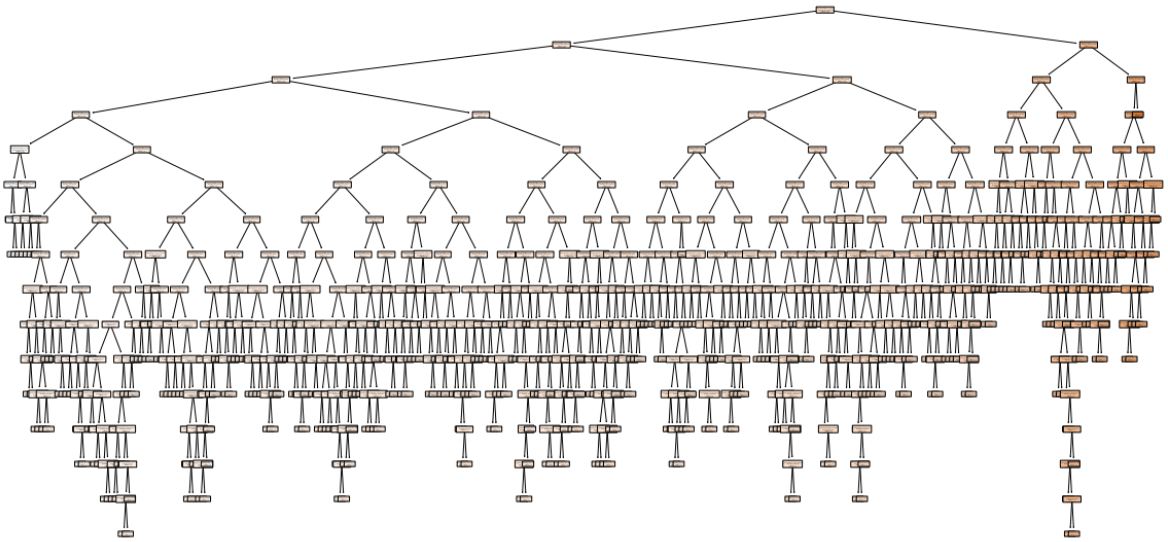
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 8: 07CPD-230- Cerro prieto dos; Mexicali, Baja California. (BCA)
 

Métricas:
MAE: 0.0346 y MAPE: 114.6751</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.8 representa un nodo similar al nodo anterior porque pertenecen al mismo sistema, el componente de energía es tan importante como cerca de un 99% lo cual sugiere que simplemente con esa variable se puede hacer el pronóstico del PML del nodo 8: 07CPD-230-Cerro prieto dos; Mexicali, Baja California, BCA. El diagrama de correlación indica que esa variable explica el comportamiento del PML casi en un 100%.

Tabla 4.9. Análisis de variables del nodo 9 (07TCT-69), con árbol de decisión y Matriz de correlación.

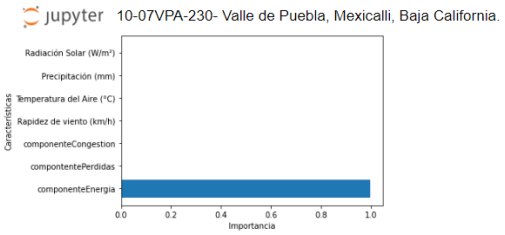
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 9: 07TCT-69- Tecate, Tijuana, Baja California. (BCA)
 

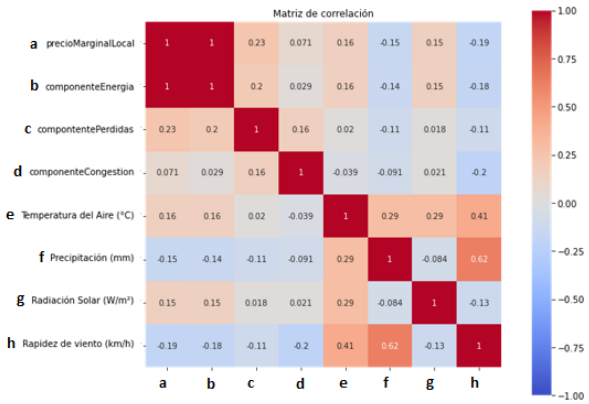
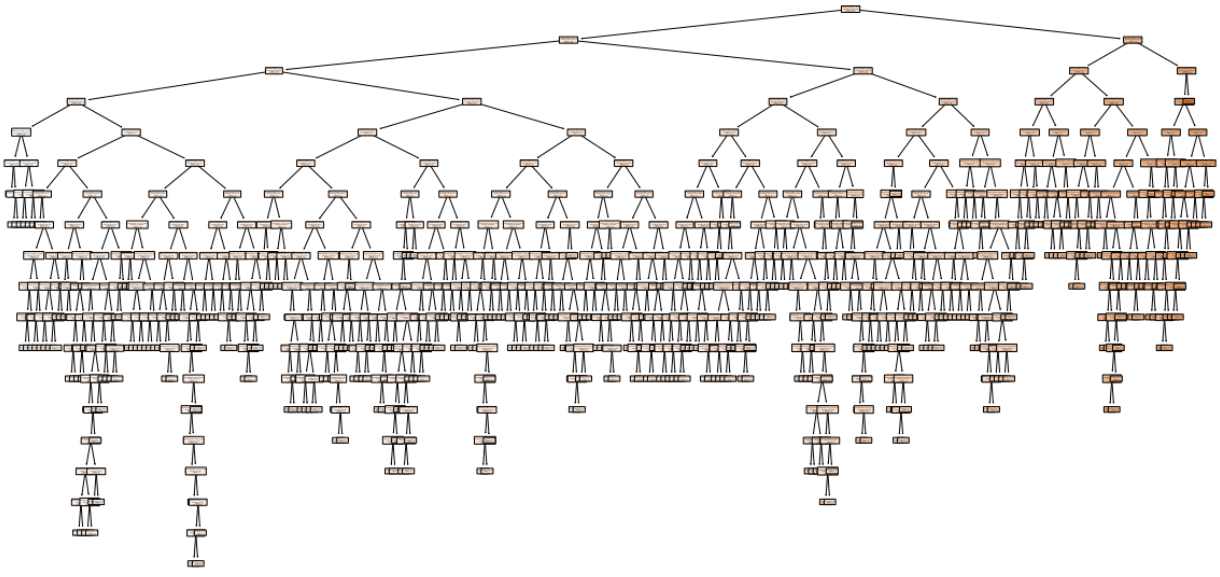
Métricas:
MAE: 0.0364 y MAPE: 103.9716</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.9 es el nodo es muy similar al nodo 7 donde se mencionó que el componente de congestión se podría descartar y que el componente de energía es tan importante con cerca de un 98% lo cual sugiere que simplemente con esa variable se puede hacer el pronóstico del PML del nodo 9: 07TCT-69- Tecate, Tijuana, Baja California, BCA. Podemos destacar que los MAPE´s están muy altos en estos nodos, parece que la variable "componente de energía" tiene una influencia significativa en el rendimiento del modelo, y puede estar contribuyendo de manera dominante a la predicción del Precio Marginal Local (PML). La presencia de otras variables

menos influyentes o incluso irrelevantes podría estar afectando el MAPE, ya que es sensible a los errores relativos.

Tabla 4.10. Análisis de variables del nodo 10 (07VPA-230), con árbol de decisión y Matriz de correlación.

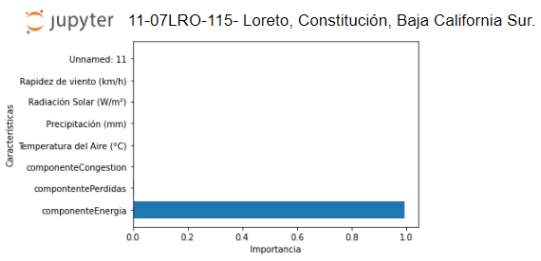
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 10: 07VPA-230- Valle de Puebla, Mexicali, Baja California. (BCA)
 

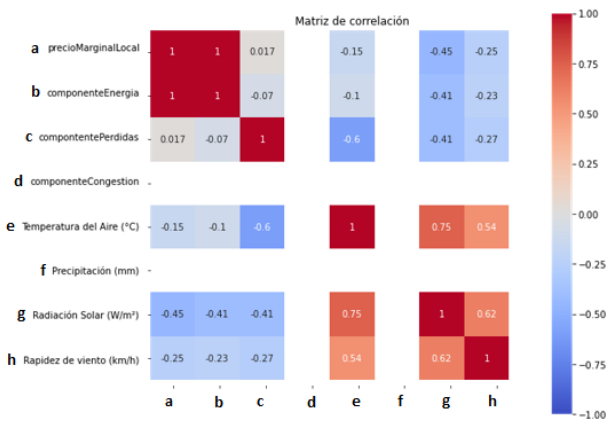
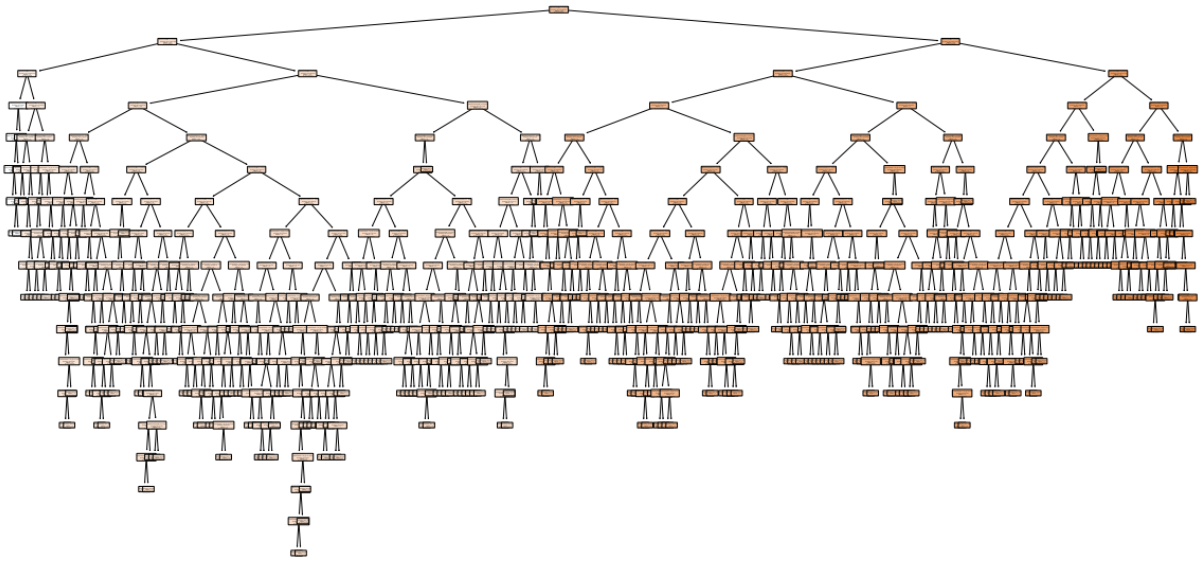
Métricas:
MAE: 0.0359 y MAPE: 106.8991</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.10 nos muestra el nodo 07VPA-230 en Valle de Puebla, Mexicali; también el componente de energía es el que influye en el PML hasta en un 98% o más. Claramente esto no generalizaría a todos los nodos del sistema de baja california por lo que es nuestro deber aplicar a

todos los nodos el mismo análisis.

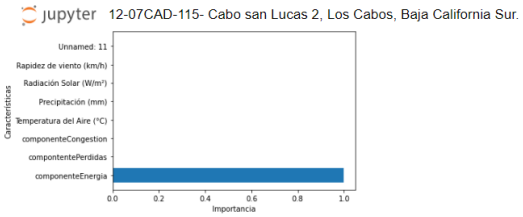
Tabla 4.11. Análisis de variables del nodo 11 (07LRO-115) con árbol de decisión y Matriz de correlación.

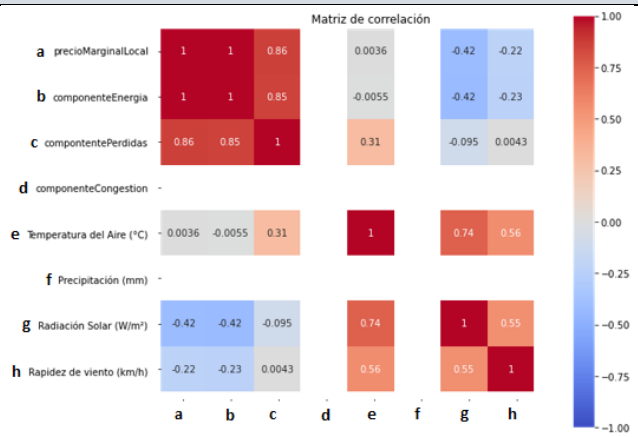
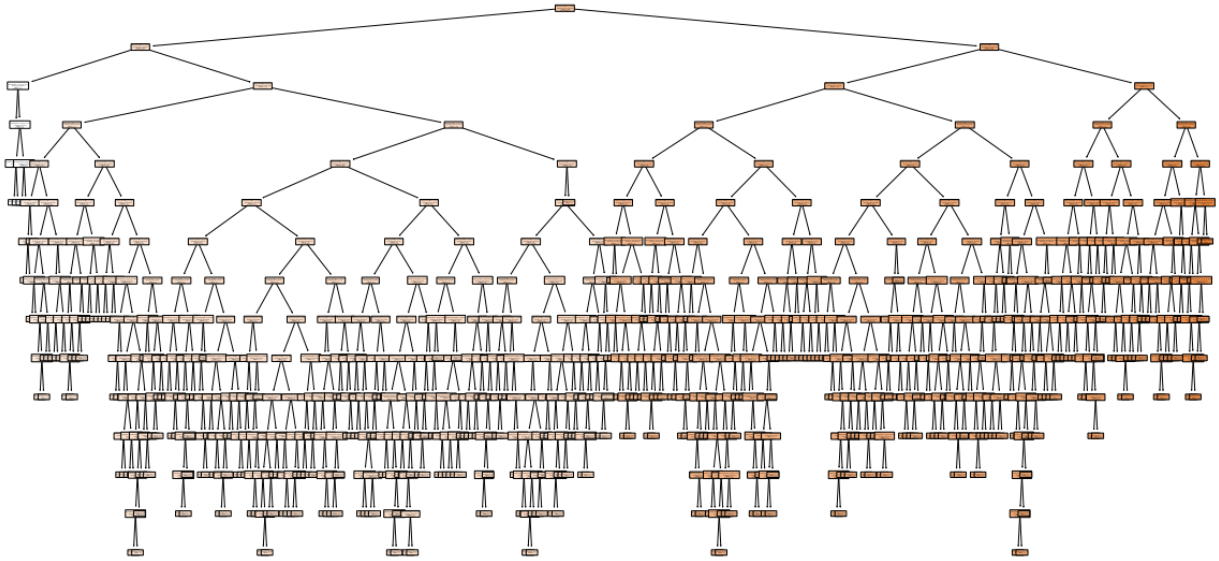
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 11: 07LRO-115- Loreto, Constitución, Baja California Sur. (BCS)
 

Métricas:
MAE: 0.0918 y MAPE: 19.4642</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

En la Tabla 4.11 tenemos un nodo similar en términos de importancia, el árbol determina al componente de energía como se esperaba, pero las correlaciones indican que todos los componentes y variables meteorológicas solo influyen en valores muy pequeños, sea directa o inversamente proporcionales. Esto sugiere algo importante; “discriminar variables sin importancia, de acuerdo con los árboles y matrices de correlación”. Sus métricas de MAE: 0.0918 y MAPE: 19.4642 representan a 07LRO-115- Loreto, Constitución de Baja California Sur (BCS).

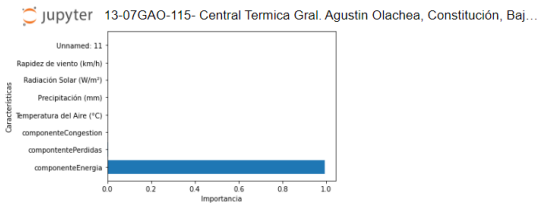
Tabla 4.12. Análisis de variables del nodo 12 (07CAD-115), con árbol de decisión y Matriz de correlación.

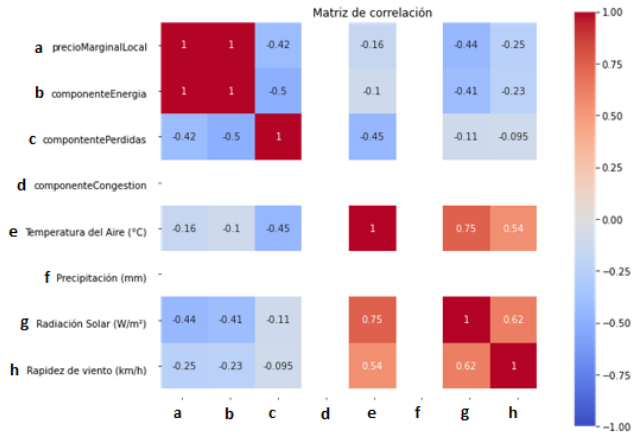
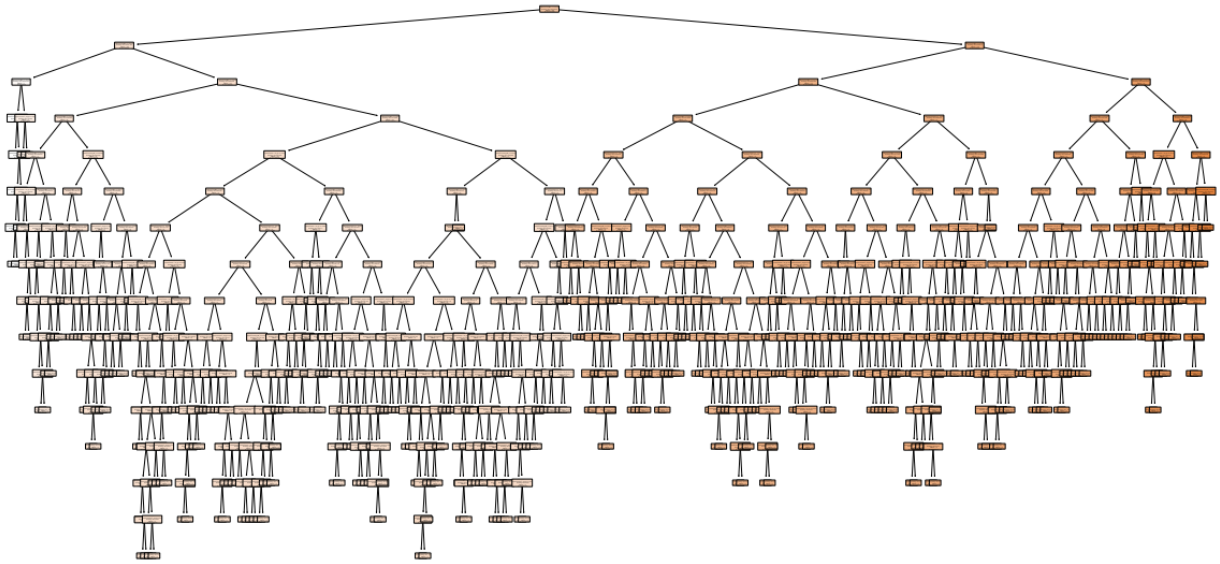
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 12: 07CAD-115- Cabo san Lucas 2, Los Cabos, Baja California Sur. (BCS)
 

Métricas:
MAE: 0.1022 y MAPE: 23.8946</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

En la Tabla 4.12 observamos al nodo 12 que es 07CAD-115 de Cabo San Lucas en Los Cabos (BCS) y es muy similar al nodo 11: 07LRO-115- Loreto, Constitución, Baja California Sur que acabamos de analizar. Comparativamente las métricas suben un poco: MAE: 0.1022 y MAPE: 23.8946 y recordemos que el nodo de Loreto tiene MAE: 0.0918 y MAPE: 19.4642.

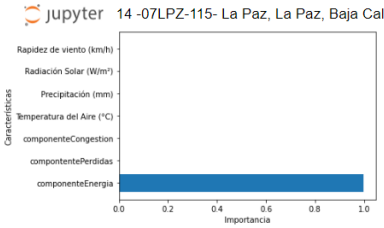
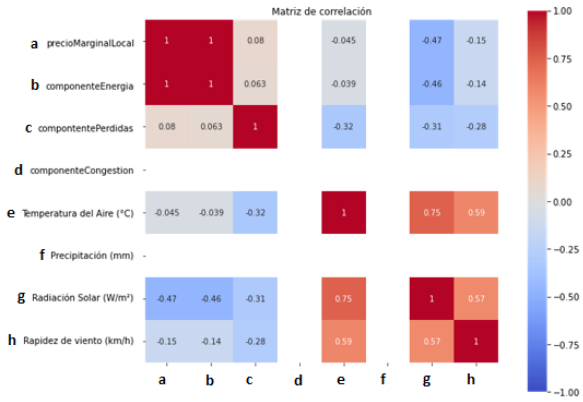
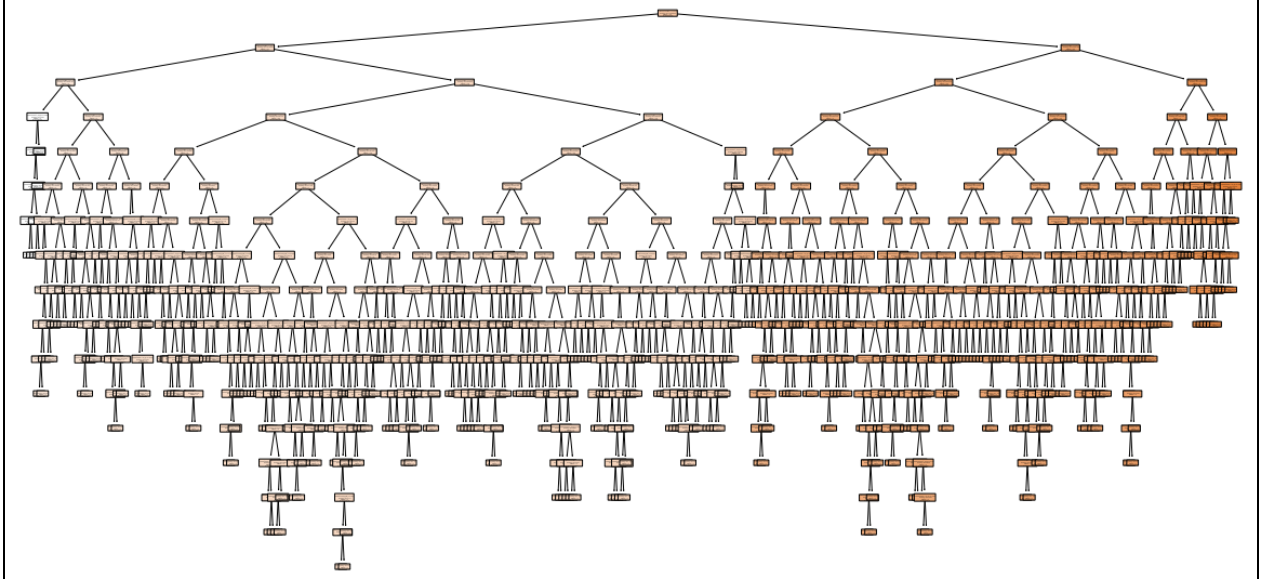
Tabla 4.13. Análisis de variables del nodo 13 (07GAO-115), con árbol de decisión y Matriz de correlación.

<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 13: 07GAO-115- Central Térmica Gral. Agustín Olachea, Constitución, Baja California Sur. (BCS)
 

Métricas:
MAE: 0.0852 y MAPE: 19.0645</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

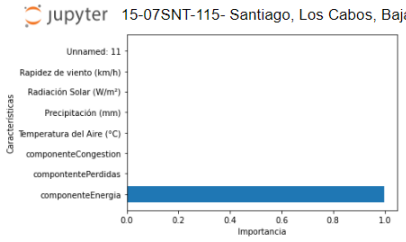
Este Tabla 4.13 es similar al nodo 11, como demuestra al nodo 07GAO-115 de Central Térmica Gral. Agustín Olachea, Constitución, Baja California Sur que pertenece a ese sistema de conexión, y que notamos una similitud tan impresionante al nodo 11 “07LRO-115” de Loreto del mismo estado de la república. En este nodo el pronostico es bueno y las variables obviamente serian el componente de energía por lo que indica el árbol de decision y el análisis de correlación.

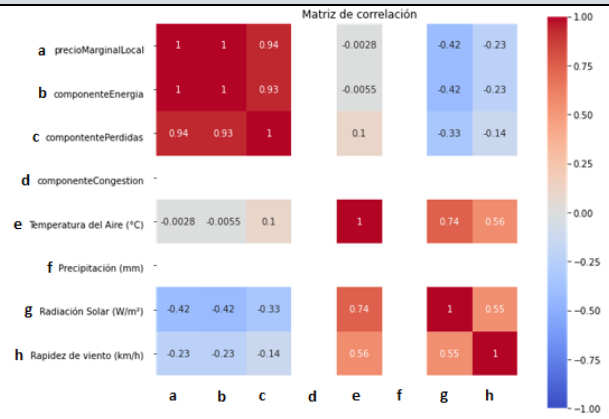
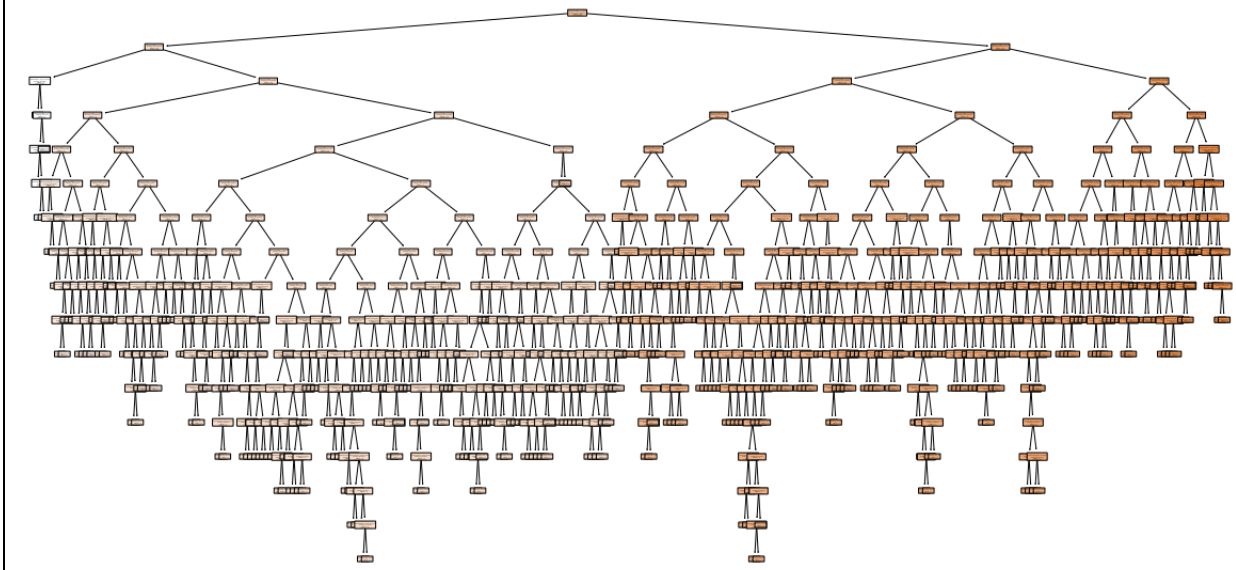
Tabla 4.14. Análisis de variables del nodo 14 (07LPZ-115), con árbol de decisión y Matriz de correlación.

IMPORTANCIA DE VARIABLES:	MATRÍZ DE CORRELACIÓN:
NODO 14: 07LPZ-115- La Paz, La Paz, Baja California Sur. (BCS)  Métricas: MAE: 0.118 y MAPE: 36.8294	
ÁRBOL DE DECISIÓN PARA REGRESIÓN:	
	

La Tabla 4.14 nos muestra el nodo 07LPZ-115 de La Paz, Baja California Sur. Es similar al nodo 13 07GAO-115 y 07LRO-115 y las métricas y valores representados en sus respectivos diagramas y tablas son similares; el componente de energía es el más relevante y significativo para el pronóstico. Sus métricas varían, pero posiblemente se deba a sus volatilidades que disparan los errores de predicción o posiblemente los datos meteorológicos son de menor calidad que los demás.

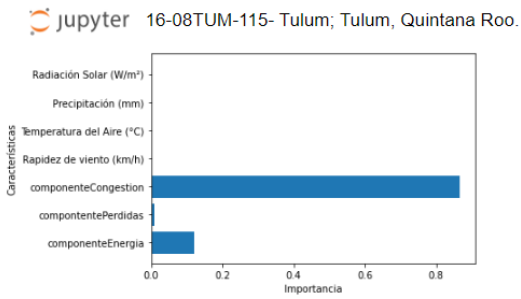
Tabla 4.15. Análisis de variables del nodo 15 (07SNT-115), con árbol de decisión y Matriz de correlación.

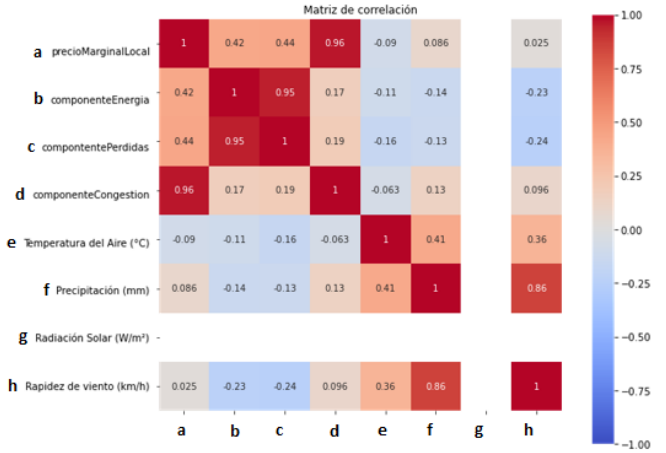
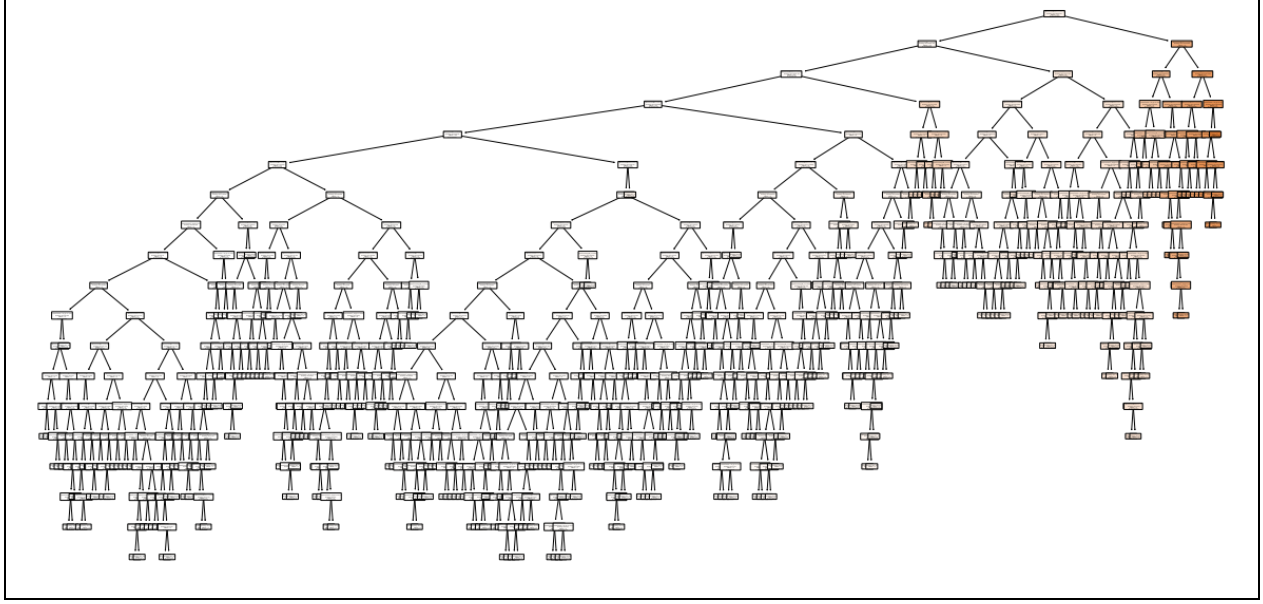
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 15: 07SNT-115- Santiago, Los Cabos, Baja California Sur. (BCS)
 

Métricas:
MAE: 0.0789 y MAPE: 17.0016</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.15 corresponde al nodo 07SNT-115 de Santiago en Los Cabos BCS. Notemos que se repiten patrones en las variables independientes, además de que la variable “componente de energía” tiene una correlación significativa al 100% podremos ver que coincide con el árbol de decisión. Métricas aceptables del modelo por lo que estamos en un sistema que el modelo acierta más veces de las que falla.

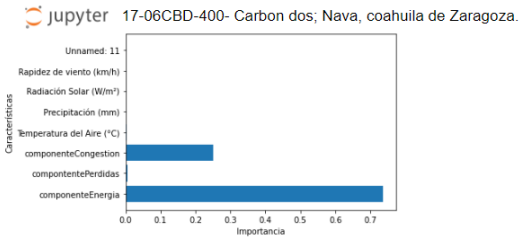
Tabla 4.16. Análisis de variables del nodo 16 (08TUM-115), con árbol de decisión y Matriz de correlación.

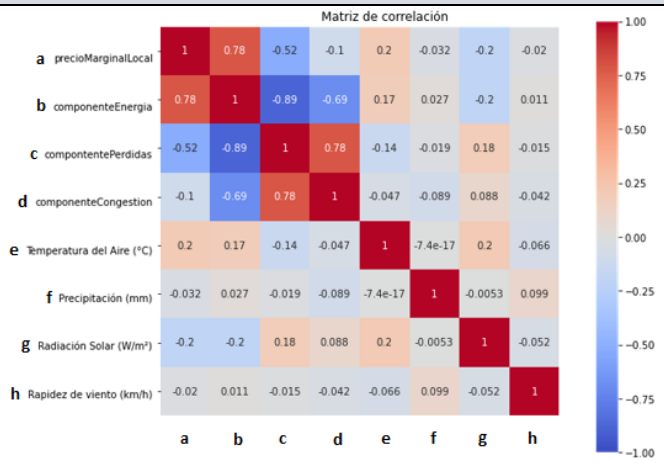
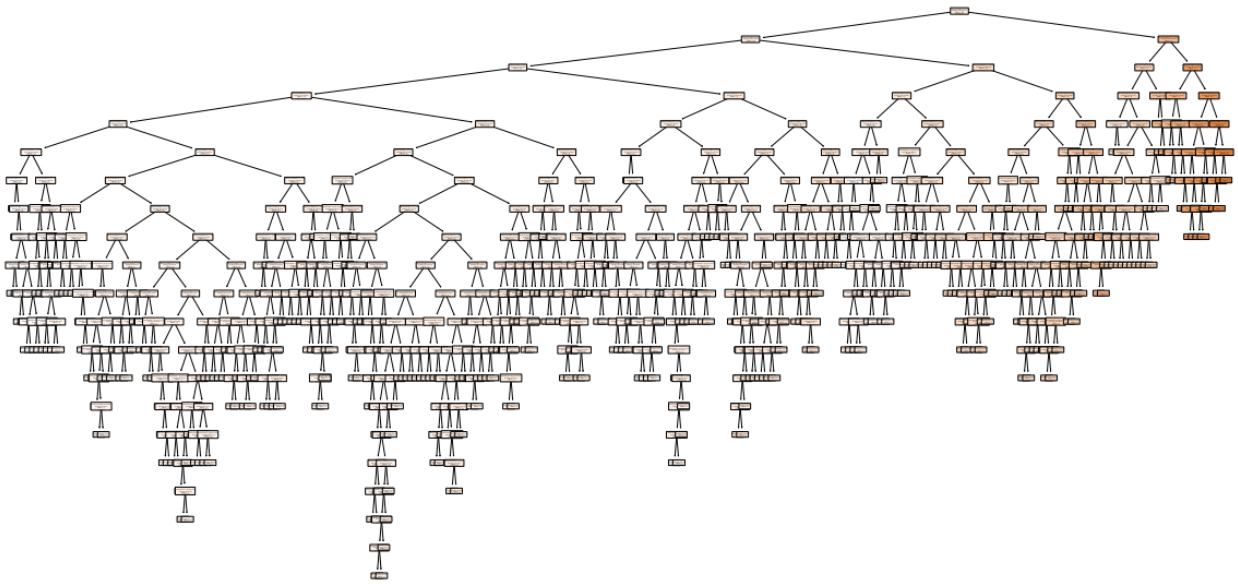
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 16: 08TUM-115- Tulum; Tulum, Quintana Roo. (SIN)
 

Métricas:
MAE: 0.0724 y MAPE: 115.5219</td><td> 
</td></tr>
<tr><td colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:
  </td></tr>
</table>

Un nodo muy diferente en todos los sentidos (véase Tabla 4.16); tenemos más de una variable de importancia y que se muestran correlaciones moderadas, como también el que ya no sea el componente de energía el más relevante, sino que ambos análisis indican que sea el componente de congestión. Los componentes del PML (energía, congestión y de perdidas) tienen los tres una influencia significativa para el árbol de decisión para regresión.

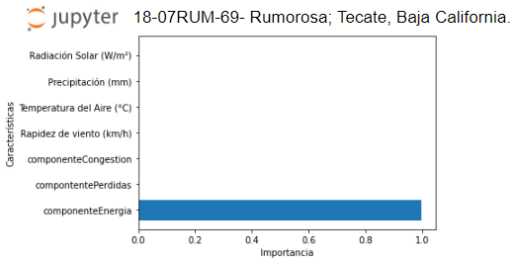
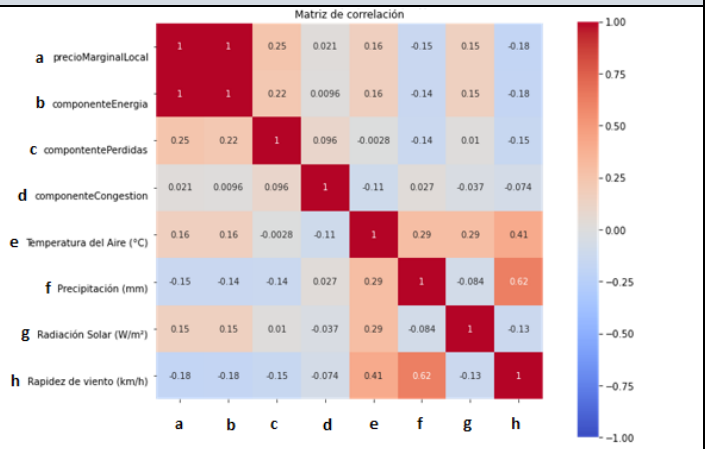
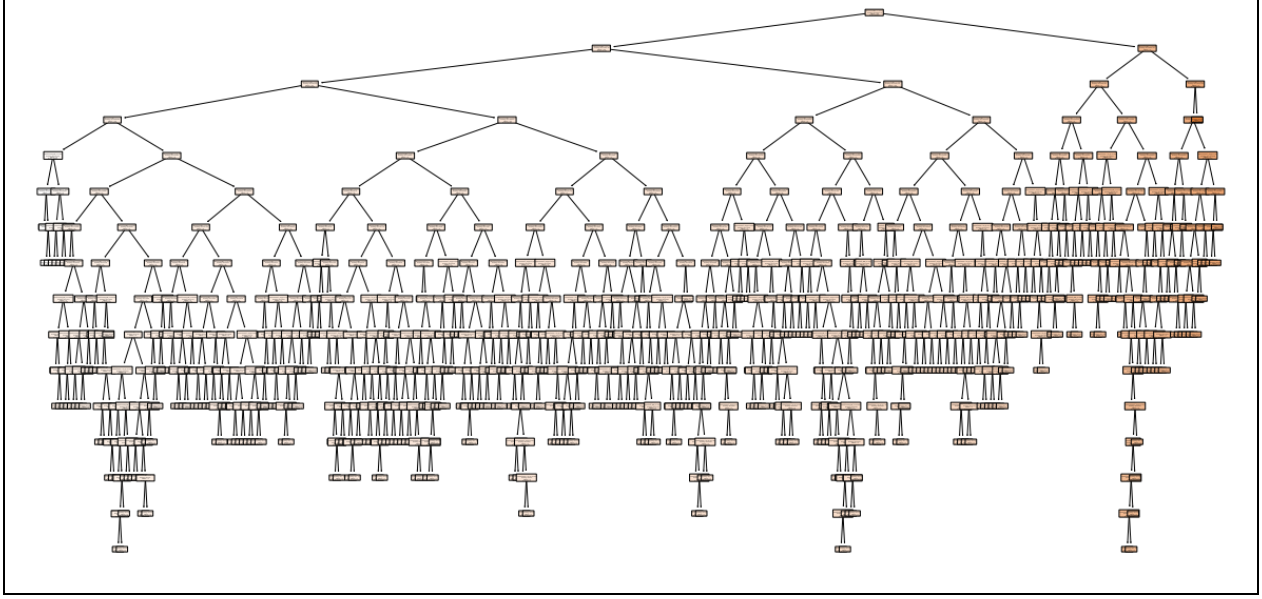
Tabla 4.17. Análisis de variables del nodo 17 (06CBD-400), con árbol de decisión y Matriz de correlación.

<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 17: 06CBD-400- Carbón dos; Nava, Coahuila de Zaragoza. (SIN)
 

Métricas:
MAE: 0.0492 y MAPE: 34.6115</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

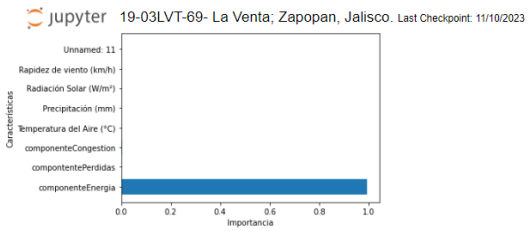
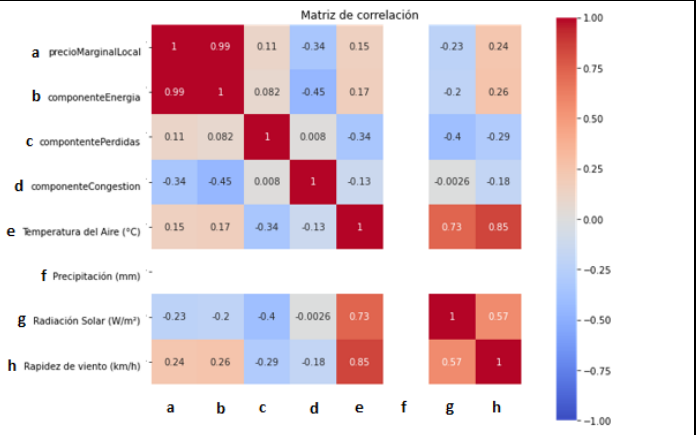
La Tabla 4.17 es el análisis del nodo 06CBD-400 en Carbón dos, en Nava Coahuila de Zaragoza perteneciente al Sistema Interconectado Nacional y este nodo presenta características similares a sus nodos de misma región, pero particularmente este nodo muestra un pronóstico mejor en contraste con el nodo de Tulum y podemos ver que son dos variables las que el árbol decide tomar como relevantes para el pronóstico.

Tabla 4.18. Análisis de variables del nodo 18 (07rum-69), con árbol de decisión y Matriz de correlación.

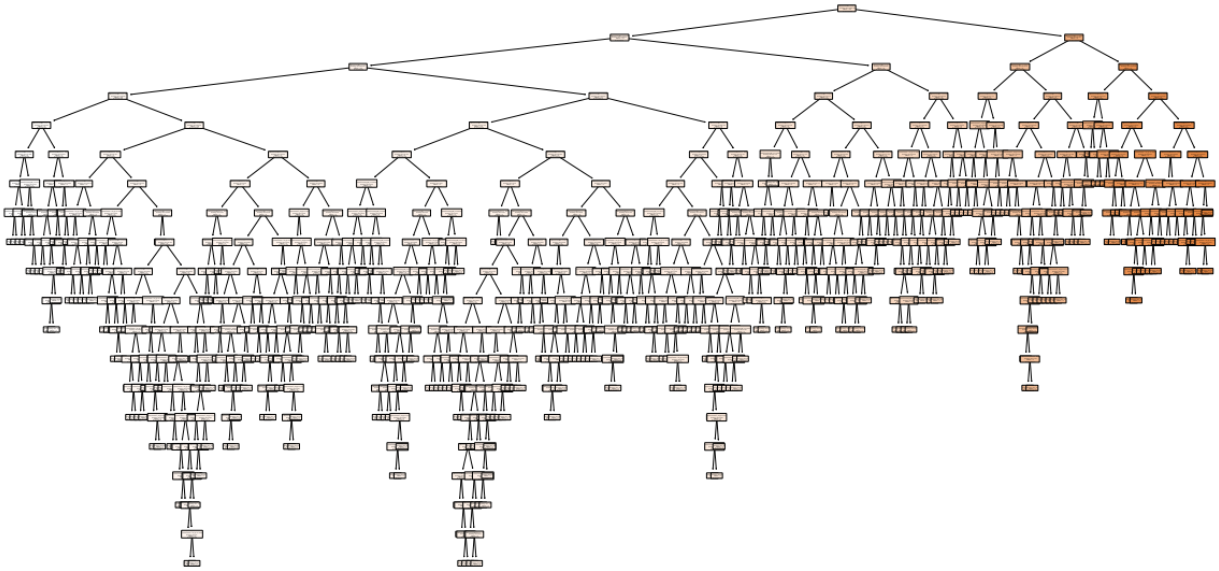
IMPORTANCIA DE VARIABLES:	MATRÍZ DE CORRELACIÓN:
NODO 18: 07RUM-69- Rumorosa; Tecate, Baja California. (BCA)  Métricas: MAE: 0.0392 y MAPE: 107.1246	
ÁRBOL DE DECISIÓN PARA REGRESIÓN:	
	

La Tabla 4.18 es otro nodo de BCA, es el nodo 07RUM-69, es una zona fronteriza del país, pero que sus métricas son similares a otros nodos de BCA como 07VPA-230 de Mexicali. Ambos análisis convergen en que el componente de energía es el relevante, podemos suponer que las demás variables que también influyen; véase que lo hace la radiación solar en un 15%, la temperatura del aire en 16%, también influyen la precipitación y el viento, pero inversamente proporcional. Sus métricas de errores se elevan también, lo que supone que hay nodos que presentaron *outliers* en el análisis, que desafortunadamente dispararon los MAPE´s porque el modelo no fue capaz de predecirlos.

Tabla 4.19. Análisis de variables del nodo 19 (03LVT-69), con árbol de decisión y Matriz de correlación.

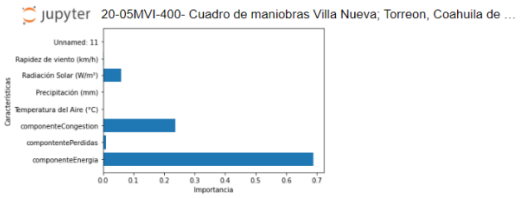
IMPORTANCIA DE VARIABLES:	MATRÍZ DE CORRELACIÓN:
NODO 19: 03LVT-69- La Venta; Zapopan, Jalisco. (SIN)  Métricas: MAE: 0.0566 y MAPE: 47.6623	

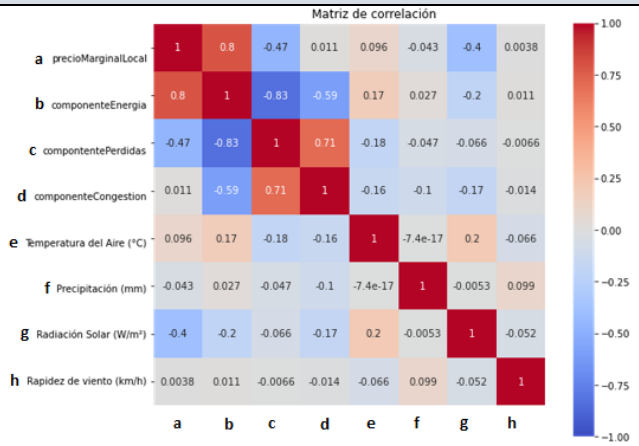
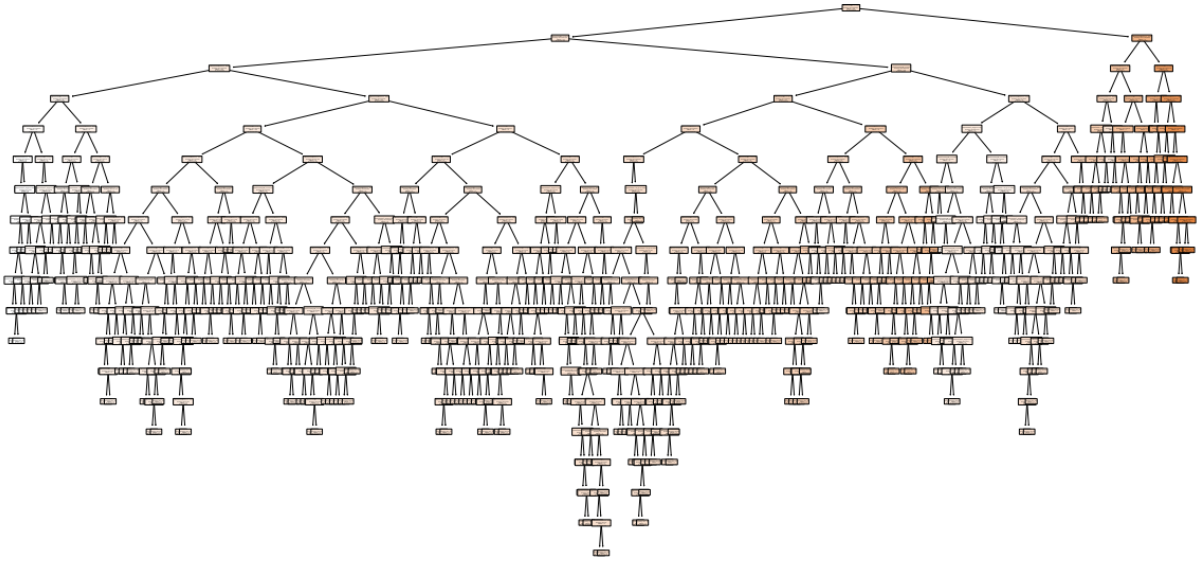
ÁRBOL DE DECISIÓN PARA REGRESIÓN:



La Tabla 4.19 es el nodo 03LVT-69 en La Venta, Zapopan, Jalisco. (SIN); vemos patrones similares pero métricas menores de MAE: 0.0566 y MAPE: 47.6623 en comparación con MAE: 0.0392 y MAPE: 107.1246 del nodo de la Rumorosa, Baja california. Nótese que las variables tienen más correlación que en el nodo del norte, pero hay una variable que no tiene importancia y es la “precipitación”, sugiriendo que la calidad de datos en las variables puede facilitar o mejorar el pronóstico y reducir las métricas del modelo LSTM secuencial con Prophet.

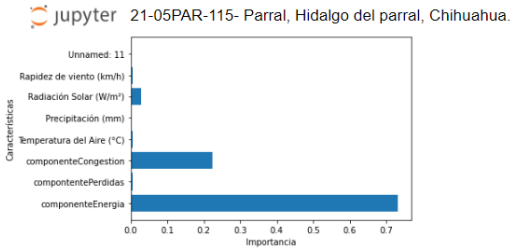
Tabla 4.20. Análisis de variables del nodo 20 (05MVI-400), con árbol de decisión y Matriz de correlación.

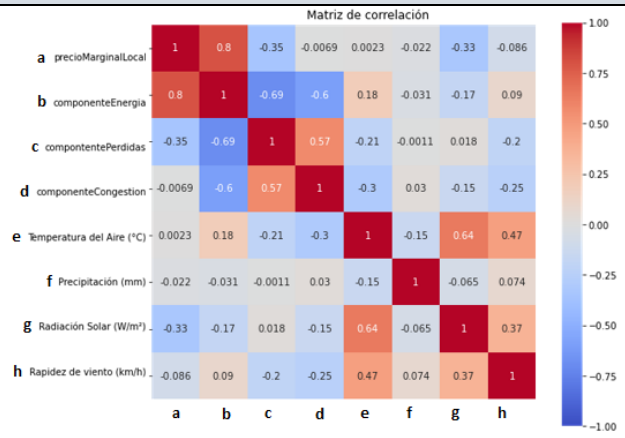
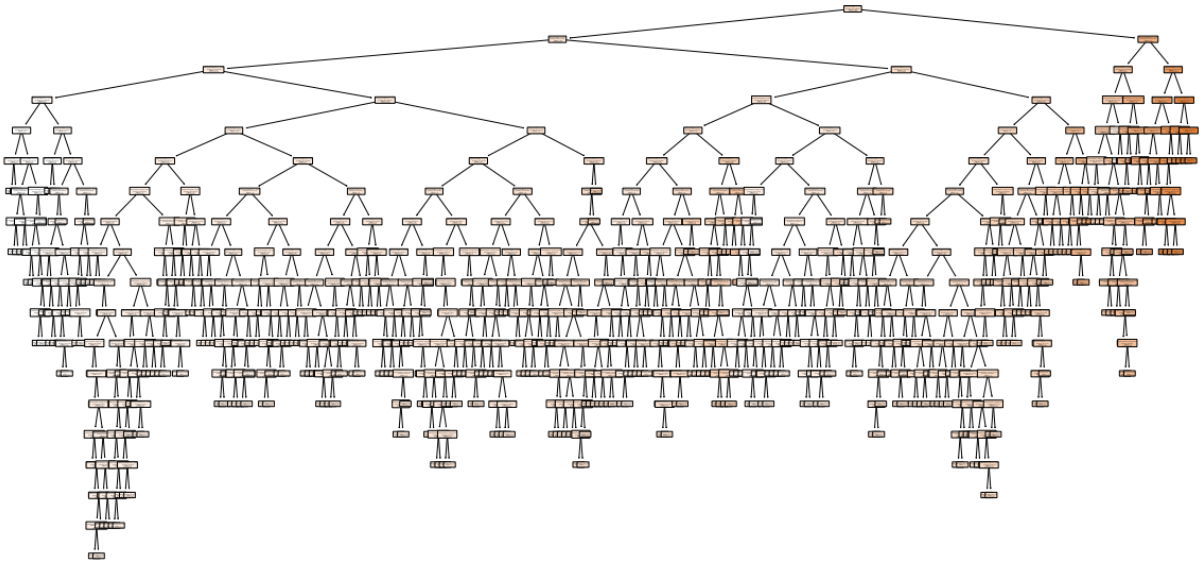
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 20: 05MVI-400- Cuadro de maniobras Villa Nueva; Torreón, Coahuila de Zaragoza.
(SIN)
 

Métricas:
MAE: 0.0494 y MAPE: 27.6973</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

Veamos la Tabla 4.20 correspondiente al nodo 05MVI-400 localizado en Cuadro de maniobras Villa Nueva, Torreón, Coahuila de Zaragoza, SIN. Aquí estamos ante un caso que pone en jaque la idea de que una sola variable importante puede reducir el error en los pronósticos, ya que este nodo y sus resultados sugieren que es bueno tener más de una variable de importancia; véase que las variables importantes que considero el árbol de decisión para regresión que son 3 y sus métricas del modelo LSTM con Prophet fueron aceptables.

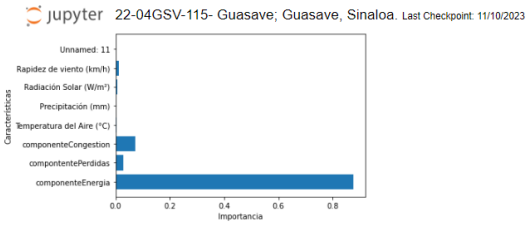
Tabla 4.21. Análisis de variables del nodo 21 (05PAR-115), con árbol de decisión y Matriz de correlación.

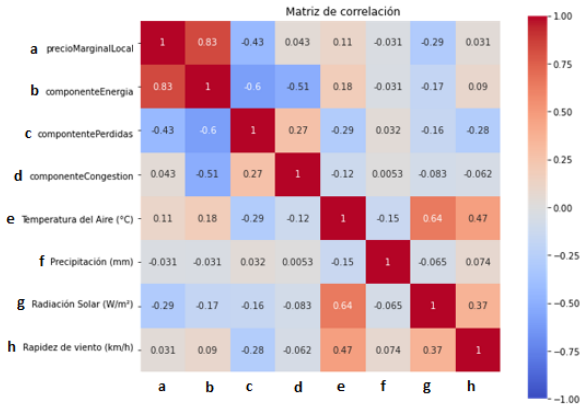
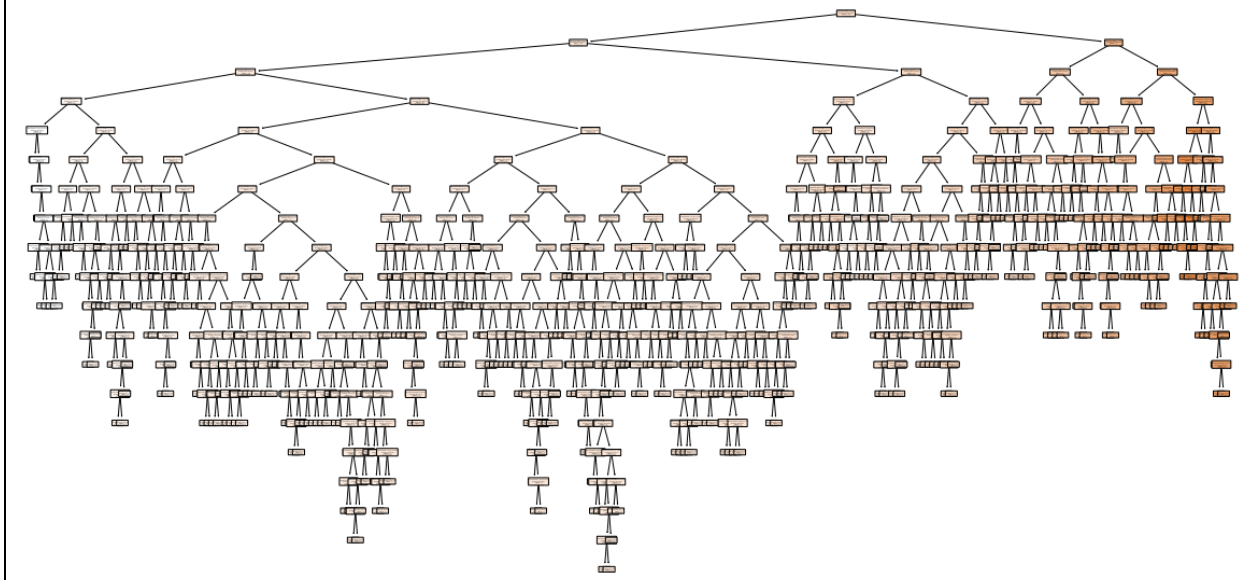
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 21: 05PAR-115- Parral, Hidalgo del parral, Chihuahua. (SIN)
 

Métricas:
MAE: 0.0482 y MAPE: 29.3315</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.21 del nodo 05PAR-115 en Parral, Hidalgo del parral, Chihuahua, SIN, es similar al nodo de Torreón y en este caso dice que sean el componente de energía, el componente de congestión y la variable de radiación solar las importantes para el pronóstico en este nodo en particular, véase grafica de barras de la importancia de las variables y sugiere que variables con calidad de datos podrán aportar al pronóstico y ser detectados por el árbol de decision, pero además de que el modelo soporta bien una base de datos completa y sugiere que el modelo es robusto y confiable.

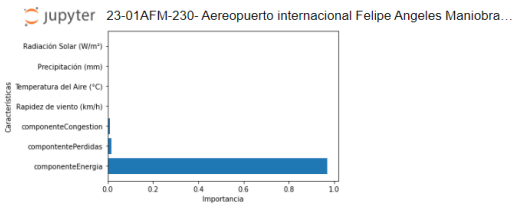
Tabla 4.22. Análisis de variables del nodo 22 (04GSV-115), con árbol de decisión y Matriz de correlación.

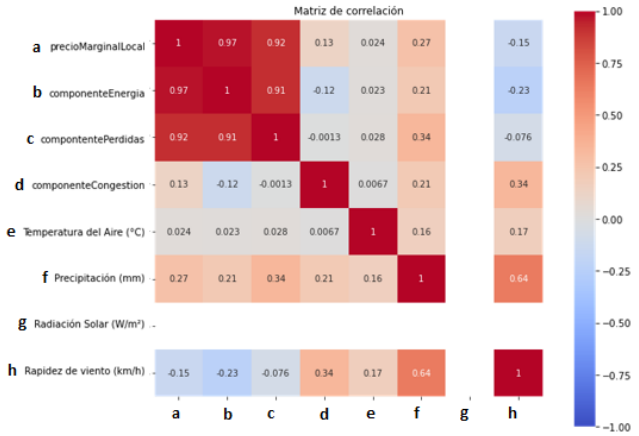
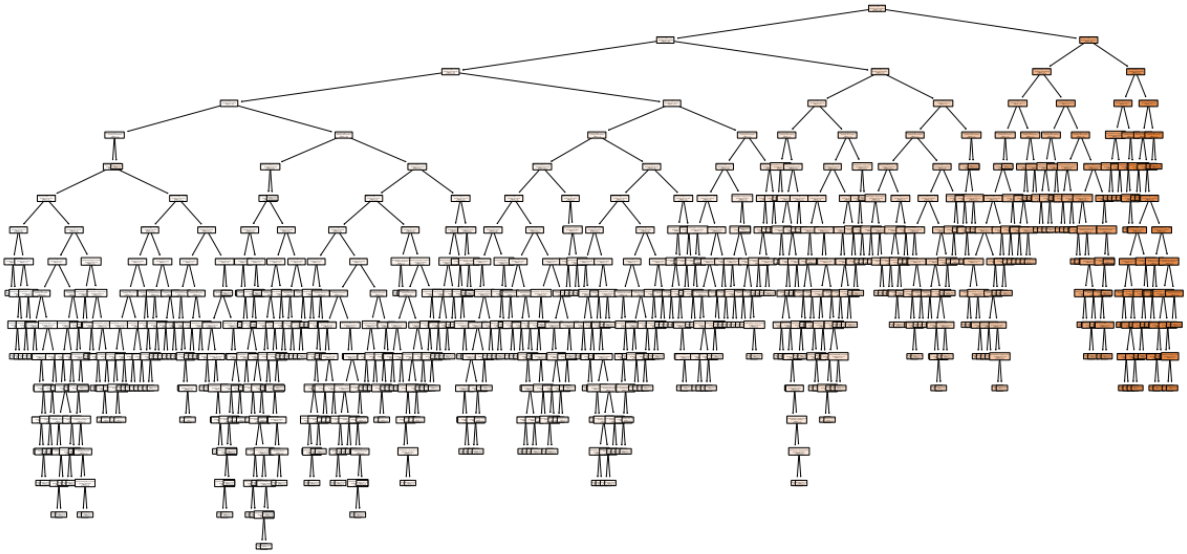
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 22: 04GSV-115- Guasave; Guasave, Sinaloa. (SIN)
 

Métricas:
MAE: 0.0509 y MAPE: 18.7698</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

Observar que en la Tabla 4.22 se ve a otro nodo del SIN; es el 04GSV-115 Guasave, Guasave, Sinaloa, SIN. Tiene a sus tres componentes de energía con importancia relevante en el pronóstico, pero también una diminuta importancia las variables de rapidez del viento y radiación solar. Entonces la más importante es el componente de energía. El diagrama de correlación es similar al nodo anterior dado que los datos meteorológicos fueron compartidos por disponer solo esos datos meteorológicos para esa región.

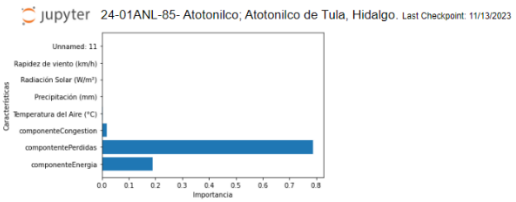
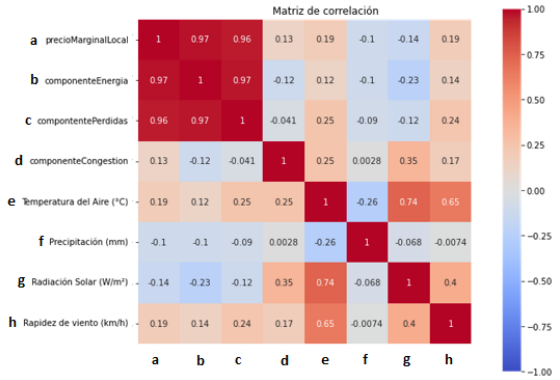
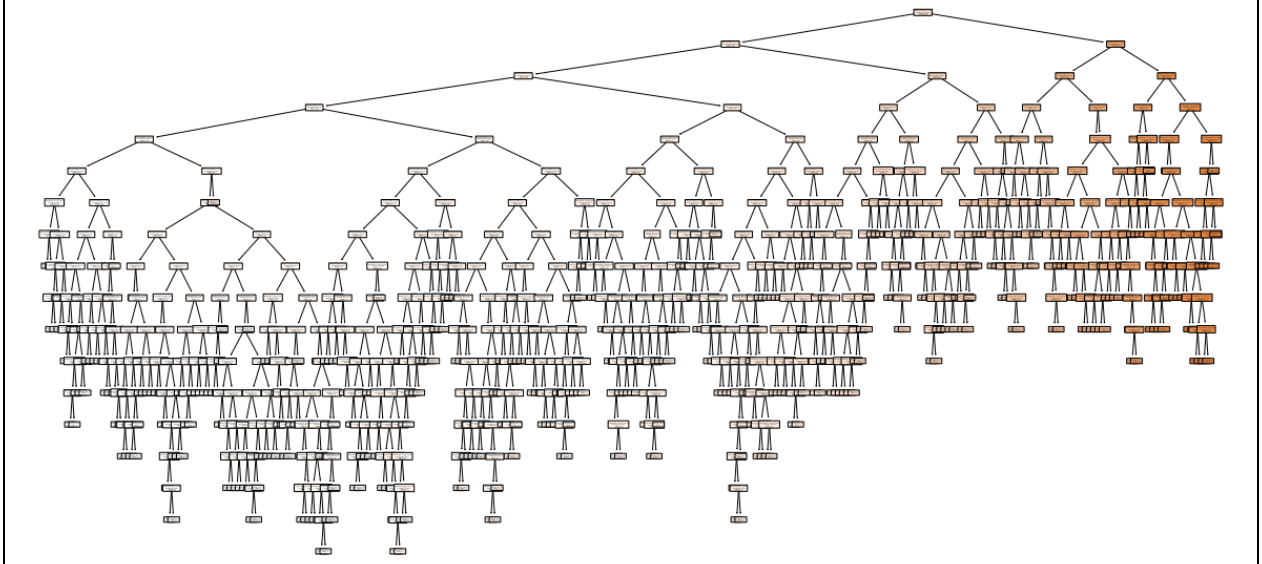
Tabla 4.23. Análisis de variables del nodo 23 (01AFM-230), con árbol de decisión y Matriz de correlación.

<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 23: 01AFM-230- Aeropuerto internacional Felipe Ángeles Maniobra; Zumpango, México. (SIN)
 

Métricas:
MAE: 0.0603 y MAPE: 61.1565</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

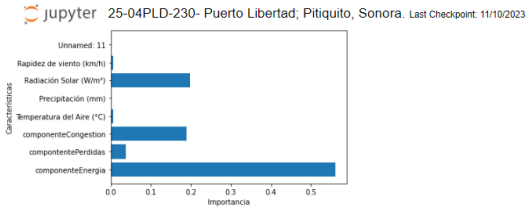
La Tabla 4.23 muestra los gráficos y resultados del nodo 01AFM-230 en el Aeropuerto internacional Felipe Ángeles Maniobra, Zumpango, Estado de México, SIN. El componente de energía es el más importante y las otras variables casi no los aprecia el algoritmo de árbol de decisión para regresión. También en la matriz de correlación se detecta una variable que no correlaciona en ningún sentido al nodo de Zumpango (la radiación solar) por lo que la calidad de datos podría ser de puros ceros que no ayudan al modelo a predecir.

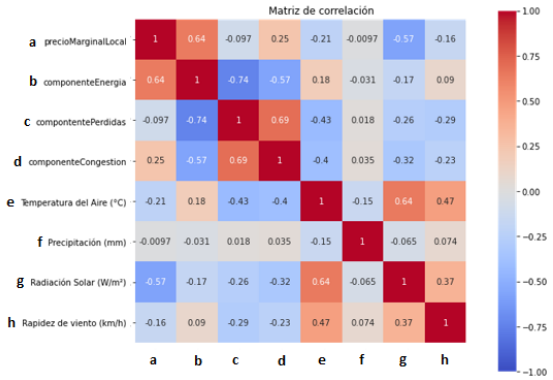
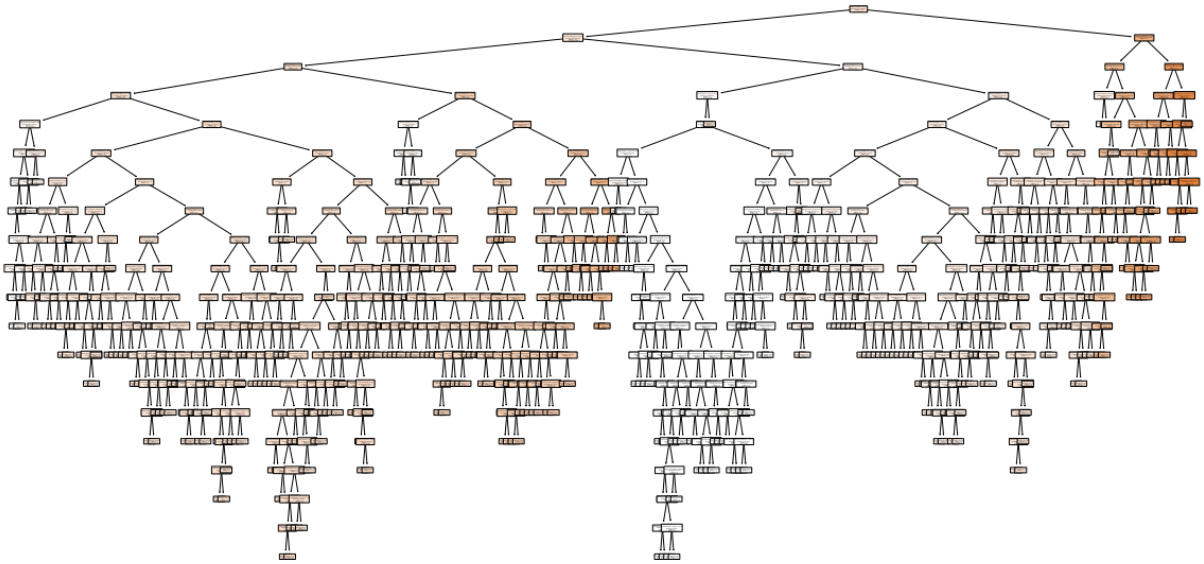
Tabla 4.24. Análisis de variables del nodo 24 (01ANL-85), con árbol de decisión y Matriz de correlación.

IMPORTANCIA DE VARIABLES:	MATRÍZ DE CORRELACIÓN:
NODO 24: 01ANL-85- Atotonilco; Atotonilco de Tula, Hidalgo. (SIN)  Métricas: MAE: 0.0538 y MAPE: 52.9369	
ÁRBOL DE DECISIÓN PARA REGRESIÓN:	
	

La Tabla 4.24 corresponde al nodo 01ANL-85 en Atotonilco de Tula, Hidalgo, SIN. Es un nodo que peculiarmente el componente de perdidas es la variable más significativa para el pronóstico con un 0.78 nivel de importancia, dejando al componente de energía en cerca de 0.2 de importancia. Observando el árbol y sus variables vemos comportamientos similares pero el diagrama de correlación es significativamente diferente a todos; la correlación de los componentes en valores por encima del 90% de correlación, y lo más interesante es que también tienen correlación las variables meteorológicas con el PML en este nodo, observando que todas influye en el precio de energía en el nodo de Atotonilco, Hidalgo, pero el modelo LSTM secuencial se eleva mucho su MAPE: 52.93% y se puede deber a una alta volatilidad.

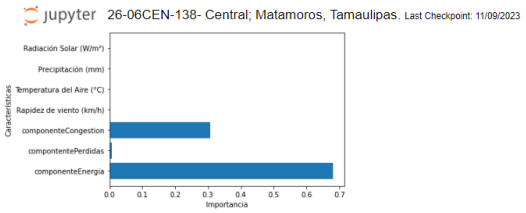
Tabla 4.25. Análisis de variables del nodo 25 (04PLD-230), con árbol de decisión y Matriz de correlación.

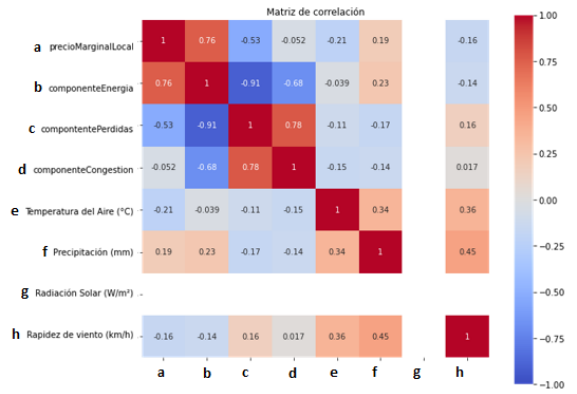
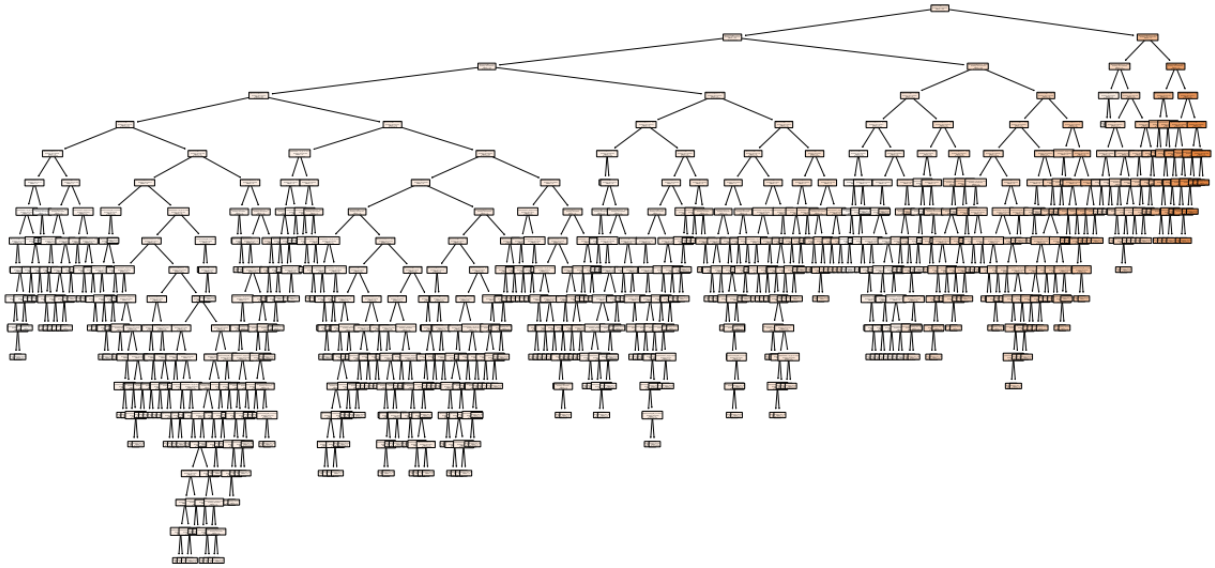
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 25: 04PLD-230- Puerto Libertad; Pitiquito, Sonora. (SIN)
 

Métricas:
MAE: 0.0579 y MAPE: 62.241</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.25 nos muestra el nodo 04PLD-230 de Puerto Libertad, Pitiquito, Sonora del SIN. Dicho nodo es complejo porque el árbol y diagrama de correlación nos demuestran que las variables son significativas por lo que quizás el modelo puede tener dificultades para generalizar un pronóstico que se asimile mucho a los PML reales. También caben otras posibles causas del porque el MAE: 0.0579 y MAPE: 62.241como ya se han mencionado antes que puede deberse a la alta volatilidad característica de los nodos del SIN.

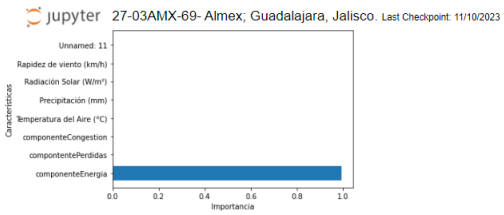
Tabla 4.26. Análisis de variables del nodo 26 (06CEN-138), con árbol de decisión y Matriz de correlación.

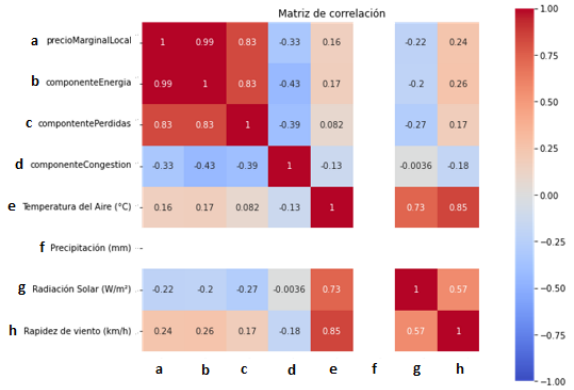
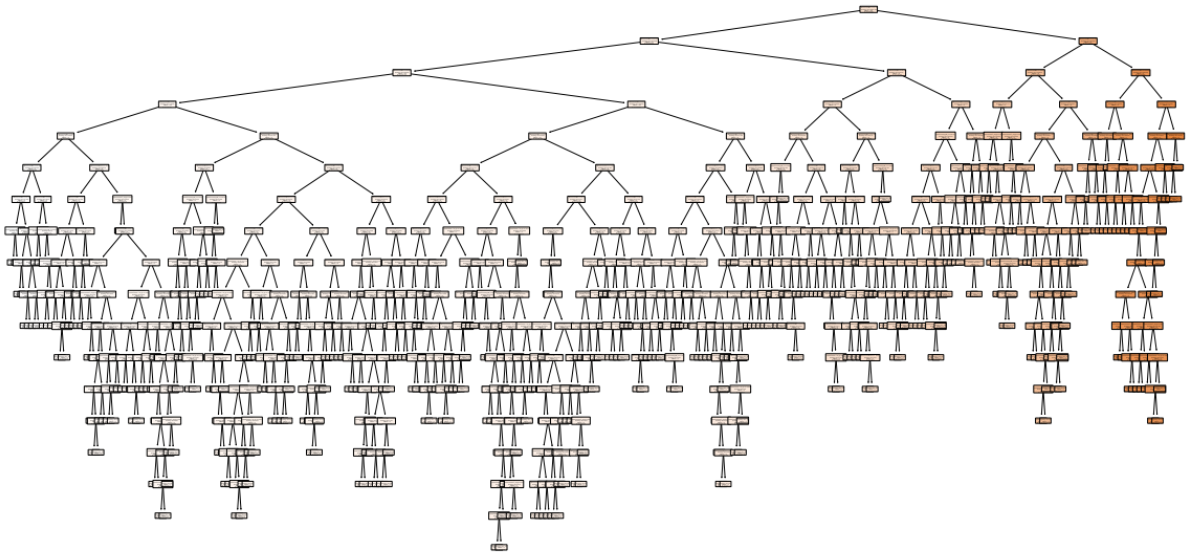
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 26: 06CEN-138- Central; Matamoros, Tamaulipas. (SIN)
 

Métricas:
MAE: 0.0536 y MAPE: 35.56</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

El nodo 26 es el nodo 06CEN-138 localizado en la central de matamoros, Tamaulipas perteneciente también al SIN (véase Tabla 4.26). Tenemos un pronóstico mejor que el nodo 04PLD-230 o el nodo 01ANL-85 recientemente explicados. Este nodo tiene de MAE: 0.0536 y MAPE: 35.56 lo cual es aceptable. Aquí el modelo predice así de bien a pesar de que aquí la variable radiación solar tiene valores en cero todo el mes, y observar que el componente de congestión tiene poca correlación a pesar de que es importante en el árbol, pero no el más importante ya que gana el componente de energía con un 0.67 nivel de importancia.

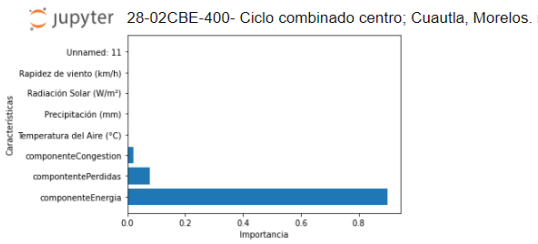
Tabla 4.27. Análisis de variables del nodo 27 (03AMX-69), con árbol de decisión y Matriz de correlación.

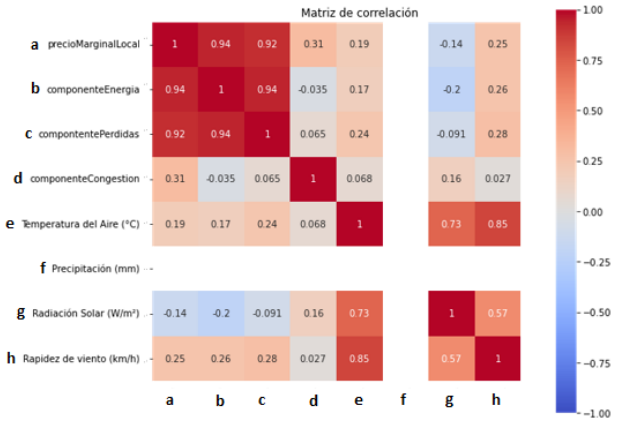
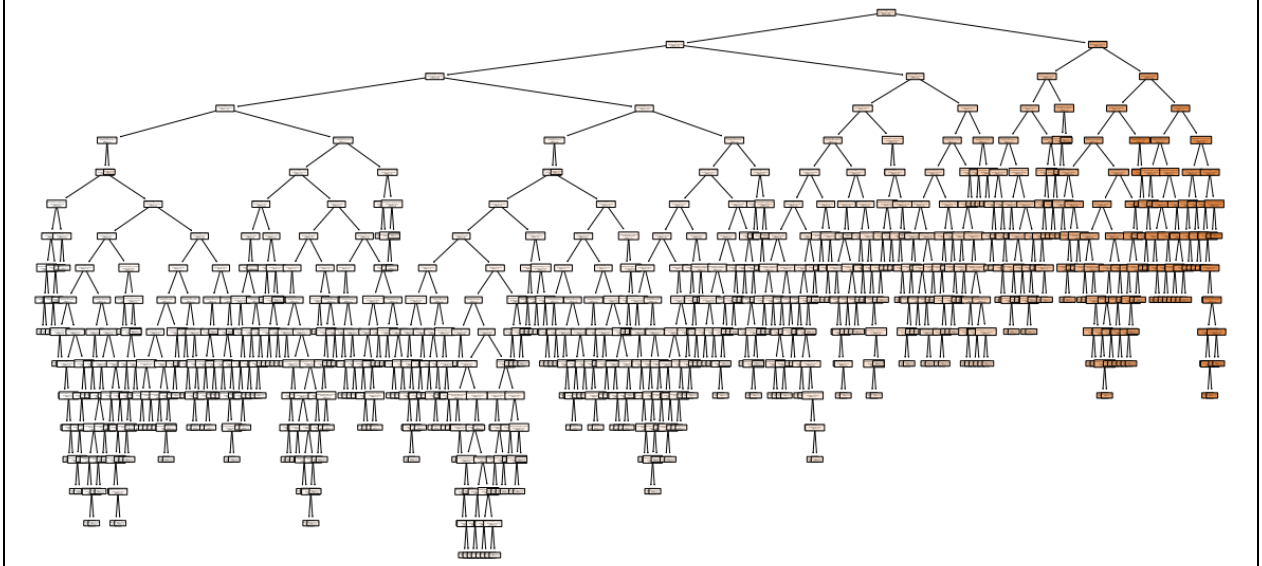
<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 27: 03AMX-69- Almex; Guadalajara, Jalisco. (SIN)
 

Métricas:
MAE: 0.0539 y MAPE: 45.3761</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

La Tabla 4.27 del nodo 03AMX-69 tiene la respuesta a la inquietud anterior, nótese que elige el árbol de decisión para regresión una sola variable (componente de energía), y que las demás variables independientes tienen poca relevancia el modelo LSTM con Prophet en comparación con el nodo anterior (06CEN-138 de MAE: 0.0536 y MAPE: 35.56) falla relativamente más con MAE: 0.0539 y MAPE: 45.376. La diferencia principal es la base de datos meteorológicos que usa este nodo ya que debe tener una calidad más útil para el pronóstico.

Tabla 4.28. Análisis de variables del nodo 28 (02CBD-400) con árbol de decisión y Matriz de correlación.

<table>
<tr><th>IMPORTANCIA DE VARIABLES:</th><th>MATRÍZ DE CORRELACIÓN:</th></tr>
<tr><td>NODO 28: 02CBE-400- Ciclo combinado centro; Cuautla, Morelos. (SIN)
 

Métricas:
MAE: 0.0571 y MAPE: 46.9182</td><td> 
</td></tr>
<tr><th colspan="2">ÁRBOL DE DECISIÓN PARA REGRESIÓN:</th></tr>
<tr><td colspan="2">  </td></tr>
</table>

Finalmente, el nodo 28 en la Tabla 4.28: es el nodo 02CBE-400 Cuautla, Morelos, SIN. Es similar a los últimos descritos y muy similar al nodo de 03AMX-69 (Guadalajara) pero vemos que tiene a los otros dos componentes de energía considerados por el árbol y observando la matriz de correlación también desaparece la variable de la precipitación. MAE: 0.0571 y MAPE: 46.9182 son similares y podemos considerar a los nodos del SIN como nodos con dificultad del modelo para acertar los PML, y puede deber a que son nodos muy inestables; con picos y valles de los PML que no entran dentro de lo normalmente aprendido por los modelos.

En resumen, a los 28 nodos podemos observar similitudes en sus datos y gráficos para aquellos que son de BCS y BCA, aunque los de BCS son mejores para predecir en el modelo LSTM se puede identificar que sus datos meteorológicos no son altamente correlacionados con el PML o importantes para el árbol, además solo es el componente de energía el que cumple ambos reconocimientos por las dos técnicas (árbol de decisión para regresión y el análisis de correlación). Por el contrario, los nodos del SIN son nodos complejos, considerablemente más aquellos del suroeste como Campeche y Tulum, por lo que esos nodos tienen por ejemplo los tres componentes de energía en valores elevados como en el nodo de Atotonilco Hidalgo, que mostro los máximos niveles de correlación en sus variables y donde en todos esos nodos el LSTM tuvo serios problemas para predecir el PML.

Lo correspondiente es aplicar lo aprendido con este resultado y adaptar estas ideas en el futuro para mejorar sus pronósticos y mejorar sus métricas. Cabe mencionar que el modelo ya está ajustado en hiperparámetros, validado con "*timeseriesplit"* como en *"walkforward validation"* y además de usar debidamente la subdivisión de conjuntos en entrenamiento, validación y prueba, se revisaron las gráficas de perdida y se recurrió tanto a un análisis de sensibilidad como una búsqueda cuadricula, por lo que el modelo y su estructura (ósea el código) es sólido en sus etapas y estructura.

La Tabla 4.29 es un resultado estadístico de las correlaciones de sus componentes de la energía nivel de coeficiente "**r**", entonces; hay componentes con correlación fuerte en la energía en el mayor de los casos, pero no siempre, pues también existen algunos nodos con correlación directamente proporcional con el componente de perdida y hasta de congestión, esto quiere decir en que algunos nodos el problema de la perdida de energía es fuerte y en otras que la saturación de la red por limitaciones técnicas propias del consumo y generación provocan ser el componente congestión como el más importante para establecer el PML. Por ejemplo, el nodo 07LPZ-115 (La Paz) su componente de energía es la variable más importante en un modelo de correlación múltiple. véase Tabla 4.29:

Tabla 4.29. Resultado de los nodos Interconectados y ordenados de acuerdo con el nivel de correlación.

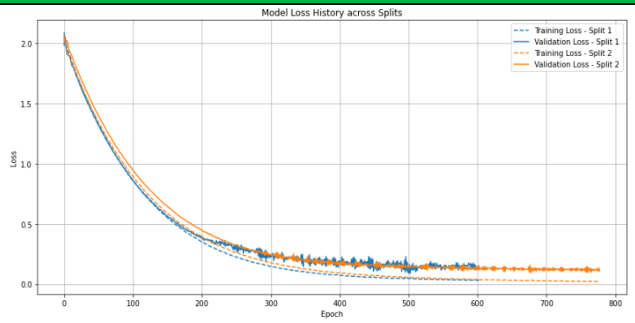
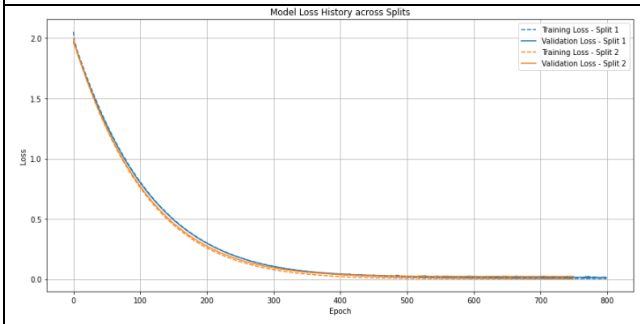
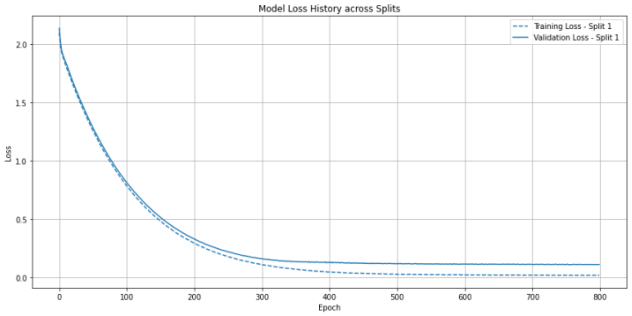
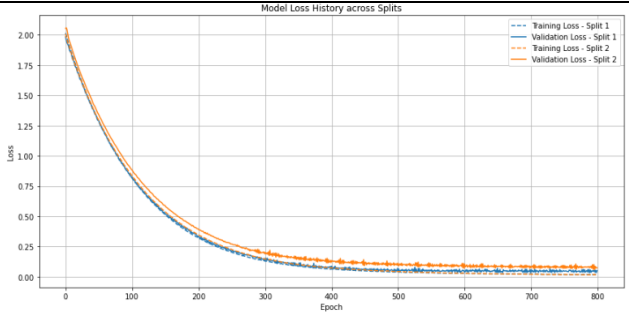
	NODOP					PRODESEN		
	CLAVE	NOMBRE	NIVEL DE TENSIÓN (kV)	SISTEMA	ZONA DE CARGA	REGION DE TRANSMISION	COMPONENTE	Coeficiente de correlación
15	07SNT-115	Santiago	115	BCS	LOS CABOS	LOS CABOS	energía	0.9999
9	07TCT-69	Tecate	69	BCA	TIJUANA	TIJUANA	energía	0.9998
12	07CAD-115	Cabo San Lucas Dos	115	BCS	LOS CABOS	LOS CABOS	energía	0.9996
14	07LPZ-115	La Paz	115	BCS	LA PAZ	LA PAZ	energía	0.9996
7	07SAF-115	San Felipe	115	BCA	ENSENADA	ENSENADA	energía	0.9991
18	07RUM-69	Rumorosa	69	BCA	MEXICALI	MEXICALI	energía	0.9955
8	07CPD-230	Cerro Prieto Dos	230	BCA	MEXICALI	MEXICALI	energía	0.9938
27	03AMX-69	Almex	69	SIN	GUADALAJARA	GUADALAJARA	energía	0.9923
13	07GAO-115	Central Termica Gral. Agustin Olachea	115	BCS	CONSTITUCION	CONSTITUCION	energía	0.9916
19	03LVT-69	La Venta	69	SIN	MINAS	GUADALAJARA	energía	0.9911
11	07LRO-115	Loreto	115	BCS	CONSTITUCION	CONSTITUCION	energía	0.9902
10	07VPA-230	Valle de Puebla	230	BCA	MEXICALI	MEXICALI	energía	0.9828
5	03AGM-400	Aguamilpa	400	SIN	TEPIC VALLARTA	TEPIC	energía	0.9797
24	01ANL-85	Atotonilco	85	SIN	CENTRO ORIENTE	CENTRAL	energía	0.9544
23	01AFM-230	Aeropuerto Internacional Felipe Angel	230	SIN	CENTRO ORIENTE	CENTRAL	energía	0.9521
3	01TTH-230	Teotihuacan	230	SIN	VDM NORTE	CENTRAL	energía	0.9505
1	02MMT-400	Manuel Moreno Torres	400	SIN	TUXTLA	GRIJALVA	energía	0.8667
20	05MVI-400	Cuadro de Maniobras Villa Nueva	400	SIN	LAGUNA	LAGUNA	energía	0.8527
21	05PAR-115	Parral	115	SIN	CAMARGO	CHIHUAHUA	energía	0.8486
2	06MON-115	Monclova	115	SIN	MONCLOVA	MONTERREY	energía	0.8449
17	06CBD-400	Carbon Dos	400	SIN	PIEDRAS NEGRAS	RIO ESCONDIDO	energía	0.8447
26	06CEN-138	Central	138	SIN	MATAMOROS	MATAMOROS	energía	0.8087
22	04GSV-115	Guasave	115	SIN	GUASAVE	LOS MOCHIS	energía	0.5565
25	04PLD-230	Puerto Libertad	230	SIN	CABORCA	HERMOSILLO	energía	0.5442
16	08TUM-115	Tulum	115	SIN	RIVIERA MAYA	CANCUN	congestión	0.8995
6	08PTE-115	Poniente	115	SIN	MERIDA	MERIDA	congestión	0.8391
4	08SLC-230	Santa Lucia	230	SIN	CAMPECHE	LERMA	congestión	0.7398
28	02CBE-400	Ciclo Combinado Centro	400	SIN	MORELOS	PUEBLA	pérdidas	0.7883

La Tabla 4.29 muestra los nodos para la implementación donde resaltan en colores azules los nodos de los sistemas BCA y BCS, mientras que los que no tienen colores son del sistema SIN. Lo más destacable de la tabla es que resume en orden las correlaciones de las variables que se identificaron de relevancia en el proyecto principalmente con árboles de decisión para regresión, pero en el caso de la Tabla 4.29 están ordenadas de acuerdo con el coeficiente de correlación "**r**" que representan la influencia medida de las variables con el PML, en sentido e intensidad.

4.2.2 Validación durante la implementación:

Durante la implementación se obtuvieron graficas con "*walkforward validation*" que avalarían si el modelo podía mejorar su desempeño para predecir.

Tabla 4.30. Validación *walk-forward* a los modelos entrenados para los nodos 1 al 4.

GRÁFICA DE DESEMPEÑO:	DESCRIPCIÓN:
	**NODO 1:** 02MMT-400 Manuel Moreno Torres; Usumacinta, Chiapas. MAE: 0.0770 y MAPE: 53.6317
	**NODO 2:** 02-06MON-115-Monclova; Monclova, Coahuila de Zaragoza. MAE: 0.0037 y MAPE: 76.6232
	**NODO 3:** 01TTH-230- Teotihuacan; Álvaro Obregón, Ciudad de México. MAE: 0.0501 y MAPE: 43.6261
	**NODO 4:** 08SLC-230- Santa Lucia; Carmen, Campeche. MAE: 0.066 y MAPE: 43.8836

En la Tabla 4.30: En el nodo 1, (02MMT-400) observamos métricas clave de desempeño en el proceso de entrenamiento cercanos a cero en cada *fold* o *split*. Se muestran el Error Absoluto Medio (MAE) de 0.0770, mientras que el Error Porcentual Absoluto Medio (MAPE) alcanza el 53.6317 en el pronóstico de ese nodo. La gráfica del entrenamiento revela dos series de validación, identificadas por dos divisiones (splits). Ambas series muestran pérdida tanto en el conjunto de entrenamiento como en el de validación. Es notable que las curvas convergen hacia valores cercanos a cero, aproximadamente 0.2. Sin embargo, una observación interesante es la presencia de irregularidades en las líneas, creando un patrón que se asemeja a las líneas de interferencia. En el contexto del *machine learning*, tales patrones pueden sugerir fluctuaciones inesperadas en la optimización del modelo durante el entrenamiento. Estas irregularidades podrían indicar problemas como *overfitting*, *underfitting* o la influencia de *outliers* en los datos.

Nodo 1: 02MMT-400 Manuel Moreno Torres; Usumacinta, Chiapas.

- MAE: 0.0770, MAPE: 53.6317. Irregularidades en las líneas sugieren posibles problemas de ajuste o influencia de datos atípicos.

Nodo 2: 02-06MON-115- Monclova; Monclova, Coahuila de Zaragoza.

- MAE: 0.0037 y MAPE: 76.6232. Notablemente bajo MAE, pero MAPE elevado. La discrepancia entre MAE y MAPE podría indicar la presencia de valores atípicos o datos extremos.

Nodo 3: 01TTH-230- Teotihuacan; Álvaro Obregón, Ciudad de México.

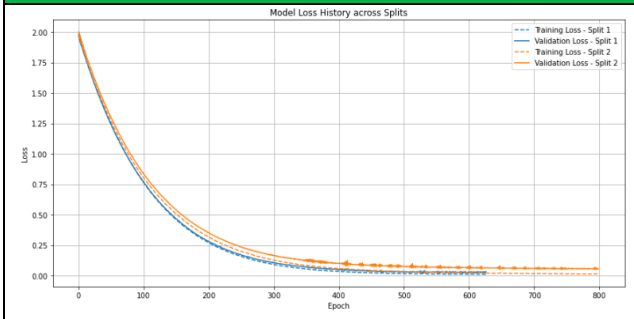
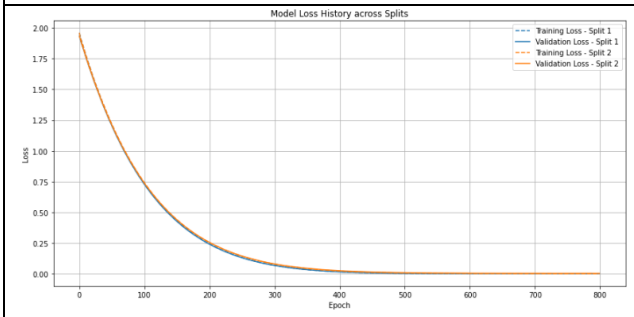
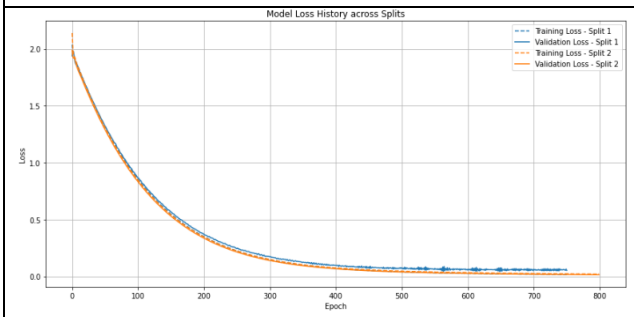
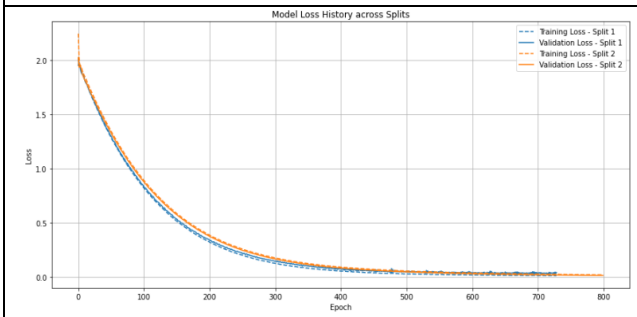
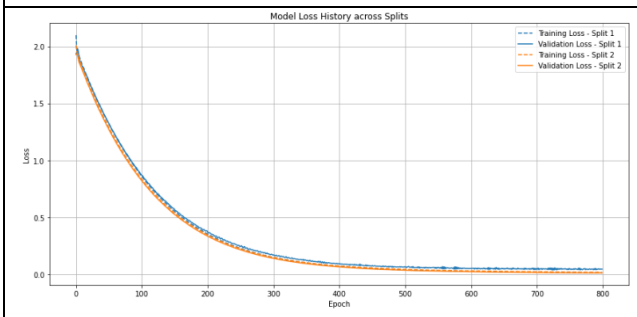
- MAE: 0.0501, MAPE: 43.6261. Pérdida moderada en entrenamiento y validación. MAPE más bajo que en el Nodo 1, indicando un mejor ajuste relativo.

Nodo 4: 08SLC-230- Santa Lucia; Carmen, Campeche.

- MAE: 0.066, MAPE: 43.8836. Similar al Nodo 3 en términos de MAPE. Patrón de pérdida en entrenamiento y validación, aunque ligeramente más alto que en el Nodo 3.

El Nodo 2 destaca por un MAE sorprendentemente bajo, pero el MAPE sugiere cierta variabilidad que requiere atención. Los Nodos 3 y 4 muestran pérdidas moderadas, con MAPE y MAE relativamente similares, indicando una mejor estabilidad en comparación con el Nodo 1.

Tabla 4.31. Validación *walk-forward* a los modelos entrenados para los nodos 5 al 9.

GRÁFICA DE DESEMPEÑO:	DESCRIPCIÓN:
	**NODO 5:** 03AGM-400- Aguamilpa, Tepic, Nayarit. MAE: 0.0617 y MAPE: 34.8548
	**NODO 6:** 08PTE-115- Poniente; Mérida, Yucatán. MAE: 0.0607 y MAPE: 103.2024
	**NODO 7:** 07SAF-115- San Felipe, Ensenada, Baja California. MAE: 0.0357 y MAPE: 101.6683
	**NODO 8:** 07CPD-230- Cerro prieto dos; Mexicali, Baja California. MAE: 0.0346 y MAPE: 114.6751
	**NODO 9:** 07TCT-69- Tecate, Tijuana, Baja California. MAE: 0.0364 y MAPE: 103.9716

En la Tabla 4.31: tenemos el Nodo 5, (03AGM-400) con MAE de 0.0617 y un MAPE de 34.8548. donde su representación gráfica muestra un comportamiento intrigante en las líneas de validación. En la época 630, estas líneas dejan de proyectarse, sugiriendo que la pérdida se ha reducido significativamente, posiblemente alcanzando valores cercanos a cero. Este fenómeno podría deberse a un ajuste exitoso del modelo, logrando minimizar la diferencia entre las predicciones y los valores reales.

Es destacable que las líneas azules de validación logran alcanzar cero, lo cual sugiere un ajuste aún más preciso para este conjunto de datos específico. La interrupción en la proyección de las líneas de validación alrededor de la época 630 podría estar asociada a diversos factores, como la convergencia del modelo o la consecución de un nivel óptimo de pérdida, indicando una convergencia efectiva del proceso de entrenamiento.

Nodo 5: 03AGM-400- Aguamilpa, Tepic, Nayarit.

- MAE: 0.0617, MAPE: 34.8548. Líneas de validación dejan de proyectarse alrededor de la época 630. Irregularidades moderadas en el descenso uniforme con líneas azules (*validation*) alcanzando cero.

Nodo 6: 08PTE-115- Poniente; Mérida, Yucatán.

- MAE: 0.0607, MAPE: 103.2024. Pérdida similar al Nodo 5, pero MAPE notablemente más alto. Necesidad de abordar la variabilidad representada por un alto MAPE.

Nodo 7: 07SAF-115- San Felipe, Ensenada, Baja California.

- MAE: 0.0357, MAPE: 101.6683. MAE más bajo que en los nodos anteriores, pero MAPE elevado. Similar al Nodo 6, sugiriendo la presencia de variabilidad no capturada.

Nodo 8: 07CPD-230- Cerro prieto dos; Mexicali, Baja California.

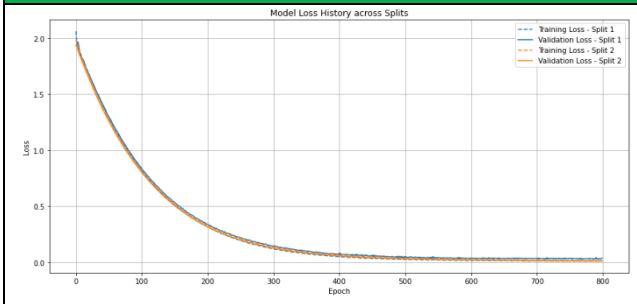
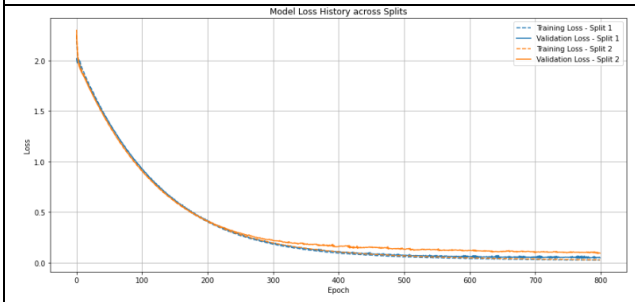
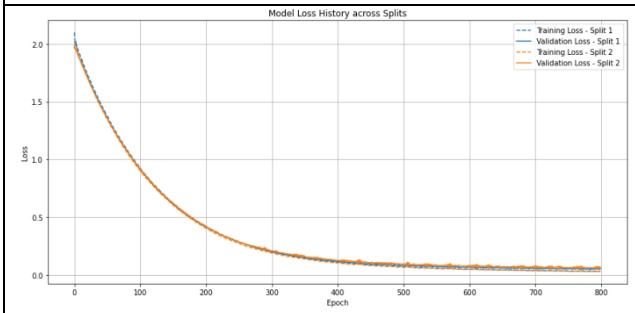
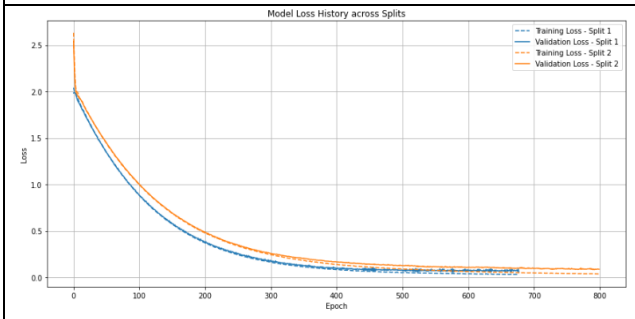
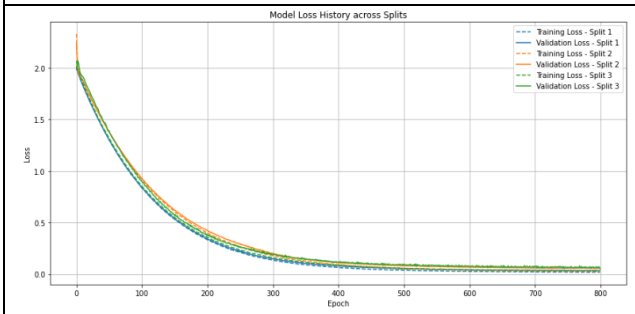
- MAE: 0.0346, MAPE: 114.6751. Bajo MAE, pero MAPE elevado. Enfoque necesario en la mejora de la estabilidad del modelo.

Nodo 9: 07TCT-69- Tecate, Tijuana, Baja California.

- MAE: 0.0364, MAPE: 103.9716. Similar al Nodo 6 y 7 en términos de MAPE elevado. Se sugiere una atención especial a la variabilidad del modelo.

Los Nodos comparten características similares con MAPE alto en sus pronósticos reales, y se observa una tendencia común de pérdidas moderadas convergentes sin mayor problema a ceros.

Tabla 4.32. Validación *walk-forward* a los modelos entrenados para los nodos 10 al 14.

GRÁFICA DE DESEMPEÑO:	DESCRIPCIÓN:
	**NODO 10:** 07VPA-230- Valle de Puebla, Mexicali, Baja California. MAE: 0.0359 y MAPE: 106.8991
	**NODO 11:** 07LRO-115- Loreto, Constitución, Baja California Sur. MAE: 0.0918 y MAPE: 19.4642
	**NODO 12:** 07CAD-115- Cabo san Lucas 2, Los Cabos, Baja California Sur. MAE: 0.1022 y MAPE: 23.8946
	**NODO 13:** 07GAO-115- Central Térmica Gral. Agustín Olachea, Constitución, Baja California Sur. MAE: 0.0852 y MAPE: 19.0645
	**NODO 14:** 07LPZ-115- La Paz, La Paz, Baja California Sur. MAE: 0.1192 y MAPE: 100.5854

Correspondientemente a la Tabla 4.32: el nodo 10, (07VPA-230) las métricas de pronóstico del modelo revelan un MAE de 0.0359 y un MAPE de 106.8991. La representación gráfica exhibe menos sinuosidad en comparación con algunos nodos previos, indicando una progresión más suave durante el proceso de entrenamiento. Todas las líneas se acercan gradualmente a cero, lo cual sugiere que el modelo está encontrando un ajuste efectivo y logrando una buena generalización. Es notable que, incluso después de alcanzar las 800 épocas, el límite establecido para el modelo, las líneas aún no han llegado completamente a cero, aunque están muy cerca. Esta situación puede sugerir que el modelo podría beneficiarse de un mayor número de épocas o ajustes finos en los hiperparámetros para lograr una convergencia más precisa.

Nodo 11: 07LRO-115- Loreto, Constitución, Baja California Sur.

- MAE: 0.0918, MAPE: 19.4642. MAE más alto que en el Nodo 10, pero MAPE considerablemente más bajo. La discrepancia entre MAE y MAPE sugiere una variabilidad baja en el modelo.

Nodo 12: 07CAD-115- Cabo san Lucas 2, Los Cabos, Baja California Sur.

- MAE: 0.1022, MAPE: 23.8946. Similar al Nodo 11, pero con MAPE ligeramente más alto. Indica una necesidad de atención a la variabilidad en los datos.

Nodo 13: 07GAO-115- Central Térmica Gral. Agustín Olachea, Constitución, B. C. Sur.

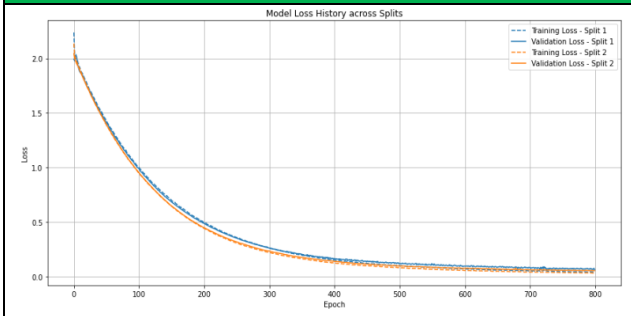
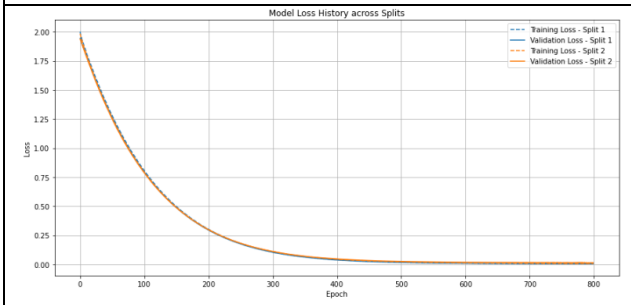
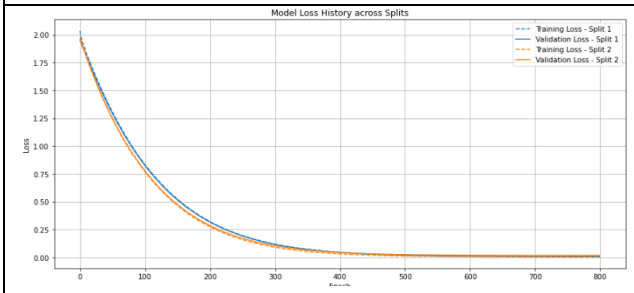
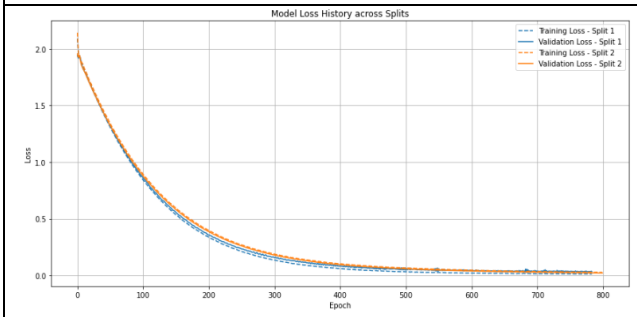
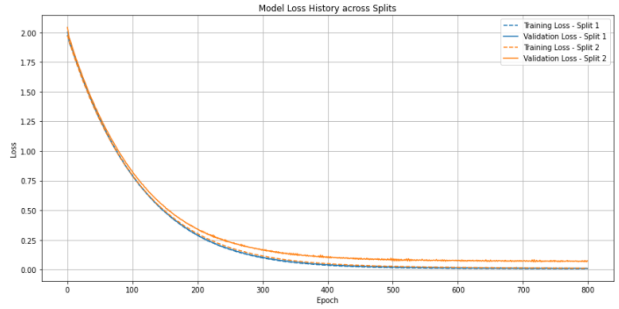
- MAE: 0.0852, MAPE: 19.0645. MAE más bajo que en los nodos 11 y 12, pero MAPE similar al Nodo 11. Sugerencia de una mejor generalización, pero aún con cierta variabilidad.

Nodo 14: 07LPZ-115- La Paz, La Paz, Baja California Sur.

- MAE: 0.1192, MAPE: 100.5854. MAE más alto y MAPE significativamente elevado. Indicación de problemas sustanciales en la generalización del modelo.

En resumen, la comparación sugiere que el Nodo 10, a pesar de no llegar a cero, tiene un rendimiento mejor en términos de generalización en comparación con los nodos 12 y 14. Los nodos 11 y 13 destacan como ejemplos de mayor estabilidad y baja variabilidad.

Tabla 4.33. Validación *walk-forward* a los modelos entrenados para los nodos 15 al 19.

GRÁFICA DE DESEMPEÑO:	DESCRIPCIÓN:
	**NODO 15:** 07SNT-115- Santiago, Los Cabos, Baja California Sur. MAE: 0.0789 y MAPE: 17.0016
	**NODO 16:** 08TUM-115- Tulum; Tulum, Quintana Roo. MAE: 0.0724 y MAPE: 115.5219
	**NODO 17:** 06CBD-400- Carbón dos; Nava, Coahuila de Zaragoza. MAE: 0.0492 y MAPE: 34.6115
	**NODO 18:** 07RUM-69- Rumorosa; Tecate, Baja California. MAE: 0.0392 y MAPE: 107.1246
	**NODO 19:** 03LVT-69- La Venta; Zapopan, Jalisco. MAE: 0.0566 y MAPE: 47.6623

Para la Tabla 4.33: inicia con el nodo 15, (07SNT-115), las métricas de rendimiento exhiben un MAE de 0.0789 y un MAPE de 17.0016. La validación gráfica de la pérdida muestra una convergencia hacia valores cercanos a cero, aproximadamente 0.1. Notablemente, a partir de la época 400, se observa un cambio en el rendimiento del modelo, donde la disminución de la pérdida ocurre a un ritmo menos rápido. Esta transición sugiere que el modelo ha aprendido eficientemente la base de datos, encontrando una generalización efectiva en las primeras épocas. La desaceleración en el descenso de la pérdida después de la época 400 podría indicar que el modelo ha alcanzado un límite de capacidad, y sugiere una posible limitación en la mejora continua, pero el rendimiento alcanzado hasta ese punto parece aceptable.

Nodo 16: 08TUM-115- Tulum; Tulum, Quintana Roo.

- MAE: 0.0724, MAPE: 115.5219. Similar al Nodo 15 en términos de MAE, pero MAPE significativamente más alto. Se sugiere una atención especial a la variabilidad en los datos.

Nodo 17: 06CBD-400- Carbón dos; Nava, Coahuila de Zaragoza.

- MAE: 0.0492, MAPE: 34.6115. MAE más bajo y MAPE más bajo que en el Nodo 15. Indicación de una mejor generalización y menor variabilidad.

Nodo 18: 07RUM-69- Rumorosa; Tecate, Baja California.

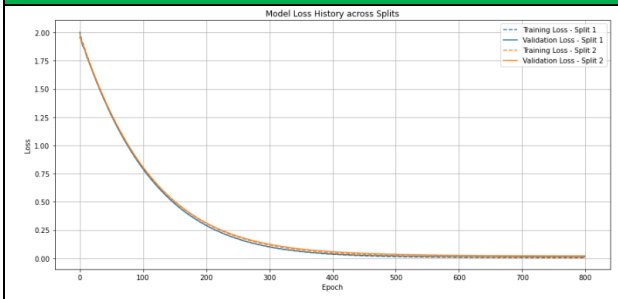
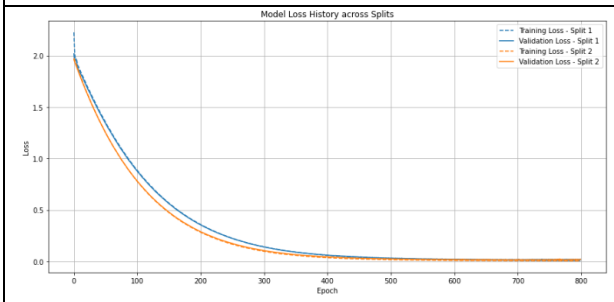
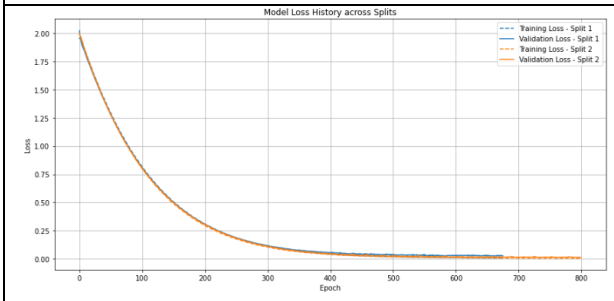
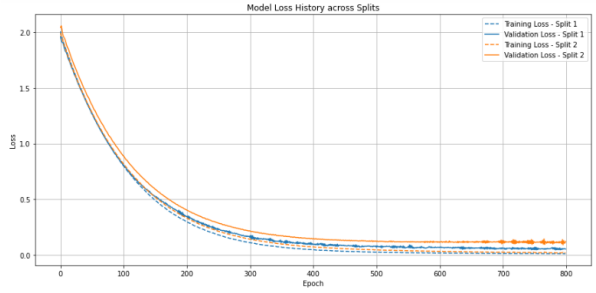
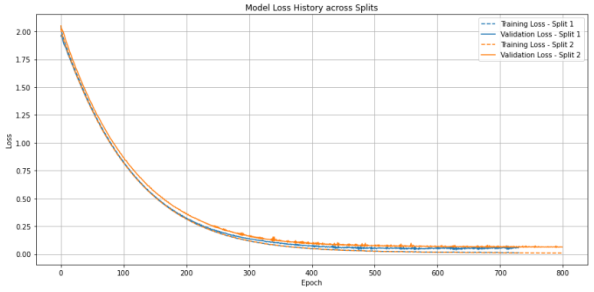
- MAE: 0.0392, MAPE: 107.1246. MAE más bajo que en el Nodo 15, pero MAPE significativamente más alto. Necesidad de abordar la variabilidad para mejorar la generalización.

Nodo 19: 03LVT-69- La Venta; Zapopan, Jalisco.

- MAE: 0.0566, MAPE: 47.6623. Similar al Nodo 15 en términos de MAE, con MAPE más bajo. Se sugiere una atención especial a la variabilidad en los datos.

La comparación sugiere que el nodo 15 y el nodo 17 muestran un rendimiento prometedor en términos de generalización y menor variabilidad, mientras que el nodo 16 y el nodo 18 podrían beneficiarse de ajustes adicionales para mejorar la estabilidad del modelo.

Tabla 4.34. Validación *walk-forward* a los modelos entrenados para los nodos 20 al 24.

GRÁFICA DE DESEMPEÑO:	DESCRIPCIÓN:
	**NODO 20:** 05MVI-400- Cuadro de maniobras Villa Nueva; Torreón, Coahuila de Zaragoza. MAE: 0.0494 y MAPE: 27.6973
	**NODO 21:** 05PAR-115- Parral, Hidalgo del parral, Chihuahua. MAE: 0.0482 y MAPE: 29.3315
	**NODO 22:** 04GSV-115- Guasave; Guasave, Sinaloa. MAE: 0.0509 y MAPE: 18.7698
	**NODO 23:** 01AFM-230- Aeropuerto internacional Felipe Ángeles Maniobra; Zumpango, México. MAE: 0.0603 y MAPE: 61.1565
	**NODO 24:** 01ANL-85- Atotonilco; Atotonilco de Tula, Hidalgo. MAE: 0.0538 y MAPE: 52.9369

Ahora en la Tabla 4.34: comenzamos con el nodo 20, (05MVI-400) las métricas de desempeño presentan un MAE de 0.0494 y un MAPE de 27.6973. La representación gráfica de la pérdida exhibe un aprendizaje convergente y tranquilo, similar al nodo 15. Alrededor de la época 400, se observa una desaceleración en la pérdida durante los dos *splits*, indicando una transición hacia un rendimiento más estable. Esta desaceleración se refleja en las dos rectas de entrenamiento y validación. Este comportamiento sugiere que, alrededor de la época 400, el modelo ha aprendido de manera efectiva y ha alcanzado un estado de estabilidad en el proceso de entrenamiento. Aunque la pérdida aún no ha llegado a cero, la desaceleración indica que el modelo ha logrado una convergencia satisfactoria.

Nodo 21: 05PAR-115- Parral, Hidalgo del Parral, Chihuahua.

- MAE: 0.0482, MAPE: 29.3315. Similar al Nodo 20 en términos de MAE y MAPE. Indicación de un rendimiento estable y convergente.

Nodo 22: 04GSV-115- Guasave; Guasave, Sinaloa.

- MAE: 0.0509, MAPE: 18.7698. MAE ligeramente más alto que en el Nodo 20, pero MAPE más bajo. Indicación de una menor variabilidad en los datos.

Nodo 23: 01AFM-230- Aeropuerto internacional Felipe Ángeles Maniobra; Zumpango, México.

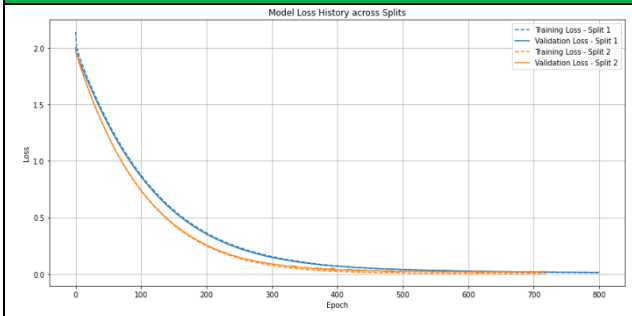
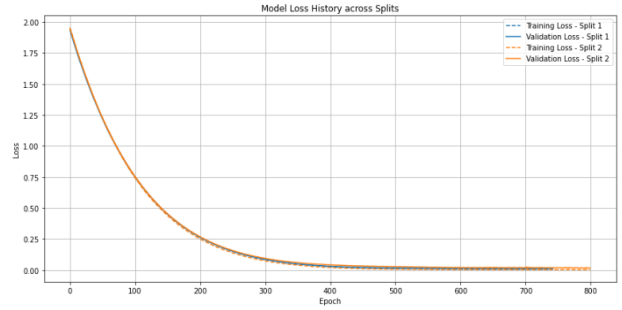
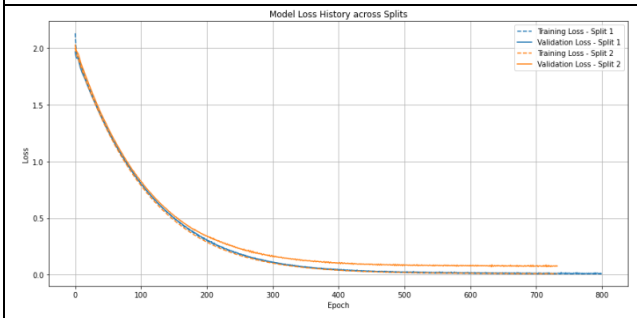
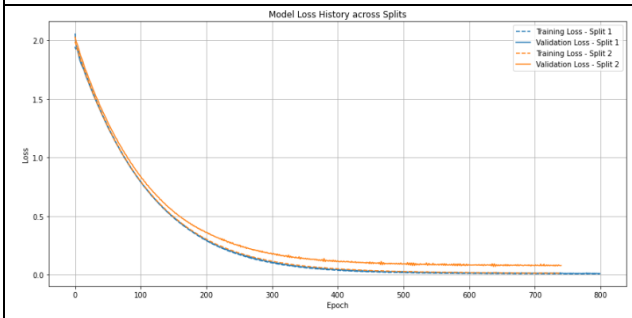
- MAE: 0.0603, MAPE: 61.1565. MAE más alto y MAPE significativamente más alto que en el Nodo 20. Se sugiere una atención especial a la variabilidad en los datos.

Nodo 24: 01ANL-85- Atotonilco; Atotonilco de Tula, Hidalgo.

- MAE: 0.0538, MAPE: 52.9369. MAE ligeramente más alto que en el Nodo 20, con un MAPE más alto. Necesidad de abordar la variabilidad para mejorar la generalización.

La comparación sugiere que los Nodos 20 y 21 tienen un rendimiento similar y estable, mientras que el Nodo 22 muestra menor variabilidad. Los Nodos 23 y 24 requieren una atención especial para abordar la variabilidad en los datos.

Tabla 4.35. Validación *walk-forward* a los modelos entrenados para los nodos 25 al 28.

GRÁFICA DE DESEMPEÑO:	DESCRIPCIÓN:
	**NODO 25:** 04PLD-230- Puerto Libertad; Pitiquito, Sonora. MAE: 0.0579 y MAPE: 62.241
	**NODO 26:** 06CEN-138- Central; Matamoros, Tamaulipas. MAE: 0.0536 y MAPE: 35.56
	**NODO 27:** 03AMX-69- Almex; Guadalajara, Jalisco. MAE: 0.0539 y MAPE: 45.3761
	**NODO 28:** 02CBE-400- Ciclo combinado centro; Cuautla, Morelos. MAE: 0.0571 y MAPE: 46.9182

Finalmente, en la Tabla 4.35: en el nodo 25, (04PLD-230), las métricas de rendimiento presentan un MAE de 0.0579 y un MAPE de 62.241. La validación gráfica revela una dinámica interesante con dos *splits*; Se observa que el *split* 2 (naranja) experimenta una mayor aceleración para converger a cero en comparación con el *split* 1 (azul). Aunque ninguno de los *splits* alcanza cero en las 800 épocas, ambos parecen estabilizarse alrededor de la época 400.

Nodo 26: 06CEN-138- Central; Matamoros, Tamaulipas.

- MAE: 0.0536, MAPE: 35.56. MAE y MAPE similares al Nodo 25. Sugerencia de estabilidad y convergencia en ambos nodos.

Nodo 27: 03AMX-69- Almex; Guadalajara, Jalisco.

- MAE: 0.0539, MAPE: 45.3761. MAE y MAPE similares al Nodo 25. Indicación de estabilidad y convergencia en ambos nodos.

Nodo 28: 02CBE-400- Ciclo combinado centro; Cuautla, Morelos.

- MAE: 0.0571, MAPE: 46.9182. MAE y MAPE similares al Nodo 25. Sugerencia de estabilidad y convergencia en ambos nodos.

La convergencia alrededor de la época 400 en el Nodo 25 es un patrón compartido con otros nodos, sugiriendo una estabilización efectiva en el rendimiento. La comparación indica que estos nodos tienen comportamientos de pérdida similares, lo que sugiere estabilidad y convergencia en sus modelos.

Podemos resumir la validación de la implementación como un despliegue del modelo a 28 nodos para verificar su desempeño mediante árboles de decisión y métricas específicas en sus pronósticos y desde su entrenamiento. Esto proporciona una evaluación detallada del modelo en diferentes contextos y escenarios representados por esos nodos específicos.

La implementación en múltiples nodos permite una validación más robusta y específica para cada ubicación, considerando las características únicas de cada nodo. Además, la elección de árboles de decisión como parte de la presentación de resultados ofrece una herramienta adicional para comprender la importancia de las variables y su impacto en las predicciones.

En este proceso, al comparar las métricas de los árboles y analizar su desempeño en diferentes nodos, se obtiene una visión más completa y detallada de la capacidad del modelo para generalizar y adaptarse a diferentes condiciones. Este enfoque detallado y específico aporta confianza y entendimiento adicional sobre la aplicabilidad y robustez del modelo en un despliegue más amplio.

4.3 Exploración y mejoras:

4.3.1 Interfaz Gráfica de Usuario (GUI):

La idea de incorporar una interfaz de usuario para que los analistas puedan ajustar manualmente la visualización gráfica y dinámica de los datos del PML es una buena herramienta. Esto proporciona flexibilidad y experiencia humana en el proceso de toma de decisiones; la razón es que, al existir diversos nodos interconectados de energía en el país (que pudimos reconocer durante la implementación), estos no siempre tienen los mismos comportamientos y características. Dado que un factor importante es el disponer de buenas bases de datos, en calidad y cantidad propicia para el buen análisis dentro del Machine Learning, se mejoró el modelo integrando un menú inicial (GUI) que demuestre la cantidad y disposición de datos antes de correr el modelo, véase Figura 4.4:

Figura 4.4 Menú principal de la Interfaz Gráfica de Usuario "*PML Explorer*".

La Figura 4.4 es la GUI que nos permite acceder a la base de datos del PML; observamos en la imagen que la interfaz se titula "Explorador de datos del PML" ("*PML Explorer*" en inglés), donde el primer paso es seleccionar el archivo que los contiene y se hace con el botón que resalta con marcador rojo donde el usuario puede dar clic en él y abrirá una ventana emergente de ubicaciones de archivos y entonces buscar el documento *"csv"* que lo cargará la ventana de selección de la interfaz, entonces procedería el usuario a explorar los datos presionando el botón correspondiente de la interfaz (menú principal) y podrá ver que fechas tienen sus datos, véase la Figura 4.5:

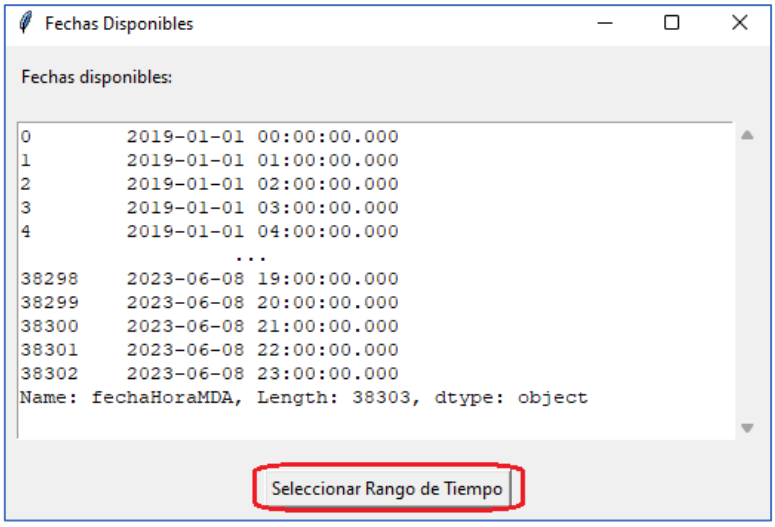


Figura 4.5 Ventana para visualizar las fechas de los datos del PML

Podemos observar que disponemos de más de 38 mil registros y que la interfaz puede reconocer la variable (columna) temporal y mostrarla en esta ventana emergente, entonces el usuario puede reconocer la fecha de inicio y fin de los datos disponibles y que podrá seleccionar el rango de visualización a criterio personal, disponiendo del botón remarcado en color rojo para ingresar manualmente las fechas a visualizar el comportamiento del PML en el periodo de tiempo de su interés, Figura 4.6:

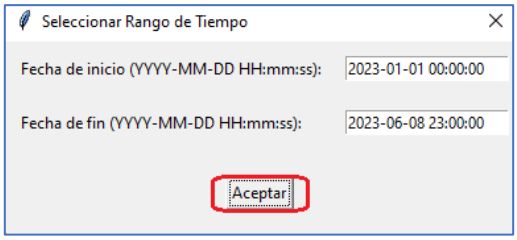


Figura 4.6 Ventana de selección del rango de tiempo para visualizar en la GUI.

En la Figura 4.6 podemos ver que se decidió poner una ventana temporal de todo lo que va del año 2023 y es importante ingresar las fechas en el formato mostrado en la ilustración para que la interfaz pueda tomar esos registros y trabajar con ellos. El usuario puede escoger todos los datos si así lo decide nada más poniendo las fechas correspondientes al primer y último registro y la interfaz los graficará también, y eso será decisión del analista que disponga de esta "*app*".

Después de presionar el botón de aceptar de la ventana de dialogo "seleccionar rango de tiempo" (Figura 4.6) nos permite visualizar los datos como se aprecia en la Figura 4.7:

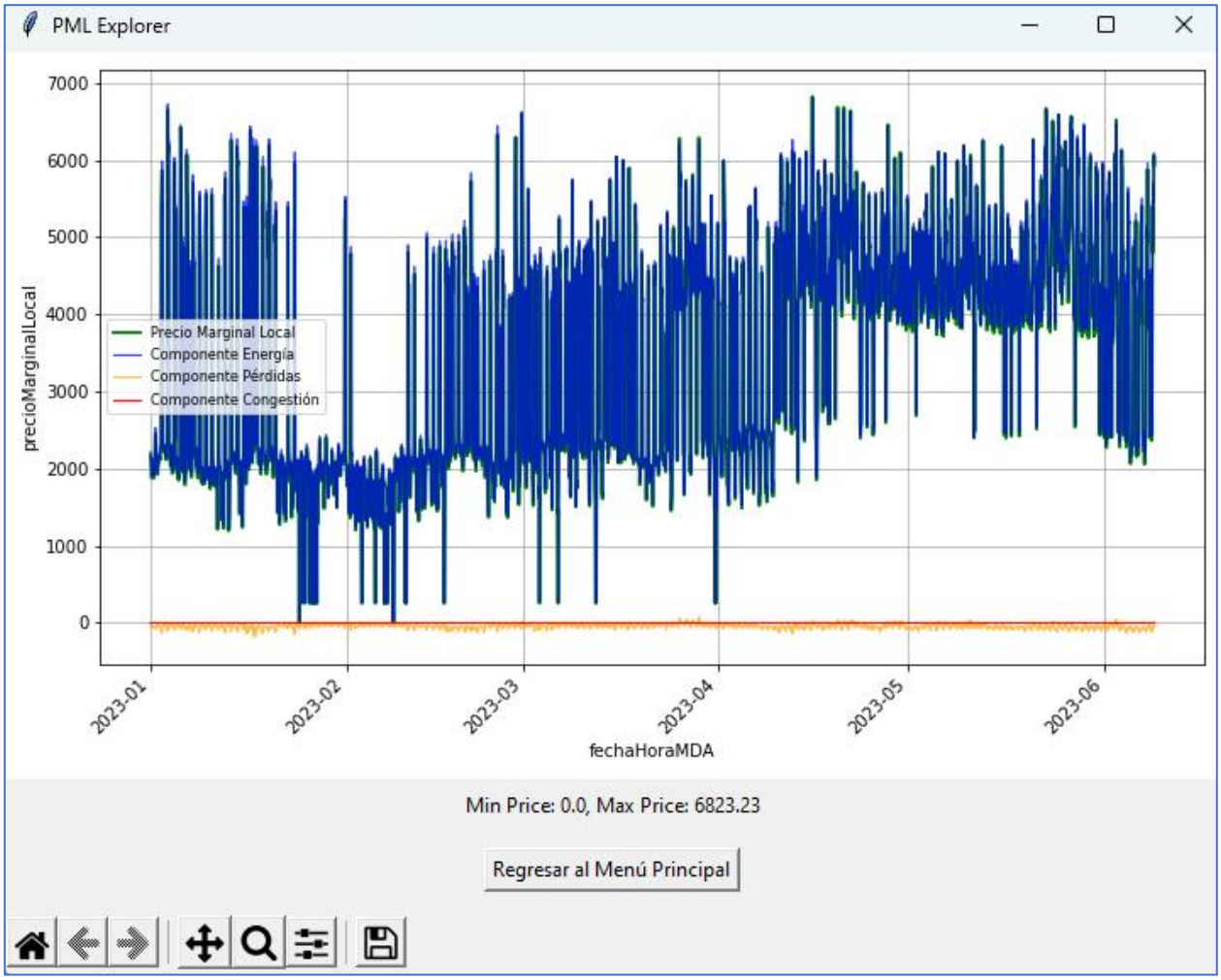


Figura 4.7 Ventana del visualizador de datos del PML de la Interfaz Gráfica de Usuario (GUI).

La Figura 4.7 muestra el rango de tiempo desde el 2023-01-01 00:00:00 hasta 2023-06-08 00:00:00 donde los precios marginales del nodo 07LPZ-115 se comportaron diferente el mes de enero al mes de febrero, notando que en el mes de febrero, el precio se mantenía en los valores más bajos (menores a los $2,400.^{00}$ pesos) pero en marzo regreso a sus valores típicos o normales (de $2,000.^{00}$ a $6,000.^{00}$ pesos) incluyendo a abril, y de mayo a julio el precio ahora no es tan barato (entre los $4,000.^{00}$ y $6,500.^{00}$ pesos el MWatt hora) y finalmente en la única semana disponible del mes de agosto el precio volvió a los precios digamos "normales" entre los $2,000.^{00}$ y $6,000.^{00}$ pesos el MWatt hora; esto indica que cada mes es diferente más que los meses son similares. Entonces el usuario ahora tiene un panorama general del comportamiento del nodo en lo que va del año, pero además puede usar la herramienta de zoom (el icono de lupa) para poder visualizar alguna sección del grafico e incluso guardar estas visualizaciones en imágenes; por ejemplo, si se selecciona las últimas dos semanas con esta herramienta tenemos la siguiente Figura 4.8:

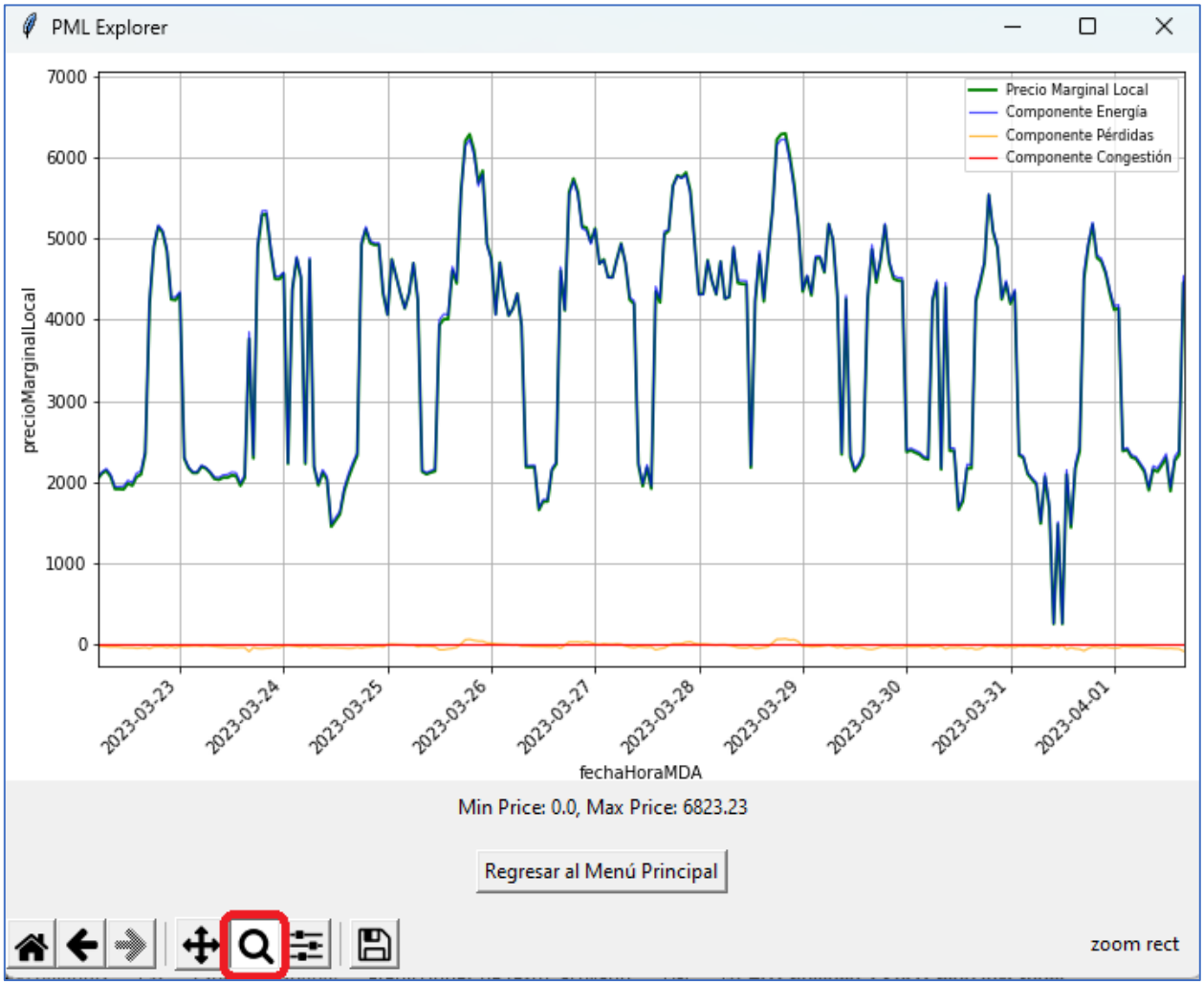


Figura 4.8 Ventana emergente del visualizador al usar la herramienta zoom de la Interfaz de Usuario.

Ahora disponemos de una gráfica más fácil de leer el PML en cuanto a sus registros y en qué momento fueron capturados, y notemos que siempre nos permite visualizar el precio máximo y mínimo por lo que la selección de tiempo que desee el usuario tiene un impacto en la percepción temporal y dinámica con la que evoluciona el PML de forma gráfica y flexible.

Cuando el usuario ya guardo las gráficas de su interés como imágenes en su computadora ahora puede dar clic en el botón “regresar al menú principal” para poder dar clic en el botón “ejecutar el código de análisis y pronostico del PML” que está en el menú principal de la GUI, véase la Figura 4.9:

Figura 4.9 Interfaz de Usuario y botón de redireccionamiento al código de Análisis y pronósticos del PML.

Esto hará que redireccione al usuario al notebook de "*jupyter*" donde está el código de análisis y pronóstico permitiéndole integrar los archivos del PML y los datos climatológicos como es la Figura 4.10:

```
[1]: # Módulo para el análisis de datos tabulares
     import pandas as pd

     def cargar_archivo_csv(archivo, fecha_column, encoding=None):
         try:
             if encoding:
                 df = pd.read_csv(archivo, parse_dates=[fecha_column], encoding=encoding)
             else:
                 df = pd.read_csv(archivo, parse_dates=[fecha_column])
             print(f"Archivo {archivo} cargado exitosamente")
             return df
         except pd.errors.EmptyDataError:
             print(f"El archivo {archivo} está vacío.")
         except pd.errors.ParserError:
             print(f"Error al cargar el archivo {archivo}. Reintentando con otro formato.")

     # Bloque 1 del pretratamiento de datos:
     archivo_csv = "07LPZ-115.csv"
     archivo_csv_latin = "LA PAZ_2023_03.csv"
     df1 = cargar_archivo_csv(archivo_csv, fecha_column='fechaHoraMDA')
     df2 = cargar_archivo_csv(archivo_csv_latin, fecha_column='Fecha Local', encoding='latin1')
```

Figura 4.10 Lectura de archivos para visualizar datos al inicio del código.

Esta función llamada "*cargar_archivo_csv*" toma tres parámetros: archivo (nombre del archivo "csv"), "*fecha_column*" (nombre de la columna que contiene fechas), y "*encoding*" (codificación del archivo, opcional). La función intenta cargar el archivo *"csv"* usando pandas, se proporciona una codificación, que sirve para leer el archivo. Si la carga es exitosa, imprime un mensaje de éxito y devuelve el "*DataFrame*" (*df*) resultante. En caso de que el archivo esté vacío,

captura la excepción "*EmptyDataError*" e imprime un mensaje. Si hay un error de análisis, captura la excepción "*ParserError*" e imprime un mensaje. El usuario puede ingresar la letra "s" si está de acuerdo con los archivos para fusionarlos, o puede ingresar la letra "n" si no está de acuerdo. El resultado es que se imprimen los mensajes en esbozos de fechas de los archivos y las decisiones tomadas en este bloque inicial del código, véase Figura 4.11:

```
Archivo 07RUM-69.csv cargado exitosamente
Archivo TECATE_2023_03.xlsx cargado exitosamente
Rango de fechas propuesto en df1: de 2019-01-01 00:00:00 a 2023-12-05 23:00:00
Rango de fechas propuesto en df2: de 2023-03-01 00:00:00 a 2023-03-31 23:00:00

¿Deseas continuar con este rango de fechas? (s/n): s
```

Figura 4.11 Widgets de dialogo para aceptar los archivos y proceder a compilarlos en el código.

Esto es útil y ahora se dispone de datos en coincidencia de fechas podemos trabajar con ambas bases de datos. Permite leer los archivos de CONAGUA, así como saber si tiene "*encoding*" (formatos) de *".csv"*, o incluso si están en formato Excel, con este "widgets" que realiza la fusión de los archivos a un solo "*dataset*" para el modelo. Ver Figura 4.12:

```
Archivo 07LPZ-115.csv cargado exitosamente
Archivo LA PAZ_2023_03.csv cargado exitosamente
Rango de fechas propuesto en df1: de 2019-01-01 00:00:00 a 2023-06-08 23:00:00
Rango de fechas propuesto en df2: de 2023-03-01 00:00:00 a 2023-03-31 23:00:00
¿Deseas continuar con este rango de fechas? (s/n): s

Fechas en el primer archivo después de ajustes:
0       2019-01-01 00:00:00
1       2019-01-01 01:00:00
2       2019-01-01 02:00:00
3       2019-01-01 03:00:00
4       2019-01-01 04:00:00
              ...
38298   2023-06-08 19:00:00
38299   2023-06-08 20:00:00
38300   2023-06-08 21:00:00
38301   2023-06-08 22:00:00
38302   2023-06-08 23:00:00
Name: fechaHoraMDA, Length: 38303, dtype: datetime64[ns]

Fechas en el segundo archivo después de ajustes:
0     2023-03-01 00:00:00
1     2023-03-01 01:00:00
2     2023-03-01 02:00:00
3     2023-03-01 03:00:00
4     2023-03-01 04:00:00
            ...
739   2023-03-31 19:00:00
740   2023-03-31 20:00:00
741   2023-03-31 21:00:00
742   2023-03-31 22:00:00
743   2023-03-31 23:00:00
Name: Fecha Local, Length: 744, dtype: datetime64[ns]
¿Las fechas en ambos archivos siguen una secuencia? (s/n): s
```

Figura 4.12 Lectura de fechas de los archivos a fusionar para iniciar el código.

4.3.2 Entrega de resultados:

Ahora se muestra el resultado final del modelo donde se cumplieron los objetivos del proyecto de tesis, donde para fines prácticos, se retoma aplicar el modelo a un nodo para demostraciones, y entregando al final los resultados de su efectividad en los nodos de la implementación.

4.3.2.1 Pronóstico del nodo 07LPZ-115

Se dispone de la base de datos PML del nodo 07LPZ-115, de La Paz, Baja California (BCS) con la que el modelo LSTM, secuencia a secuencia (secuencial) podrá aprender de las variables y entregar un pronóstico confiable. Recapitulando, se mejoró el enfoque de LSTM y el enfoque de varios puntos a futuro con una neurona llamada "Dense", que tiene la capacidad de poder darnos los números de "*steps"* solicitados (48 horas a futuro), y faculta a la neurona de salida a entregar 48 valores a predecir al futuro aprendidos del pasado inmediato. Véase Figura 4.13:

```
[86]: # Paso 7: Construcción del modelo
      # Definir el número de time steps
      time_steps = 48

      # Iniciar TimeSeriesSplit
      tscv = TimeSeriesSplit(n_splits=3)
      history = []  # Para almacenar la historia de entrenamiento de cada división

      # Función para crear un dataset a partir de los datos
      def create_dataset(X, y, time_steps=1):
          Xs, ys = [], []
          for i in range(len(X) - time_steps + 1):
              v = X.iloc[i:(i+time_steps), 1:].values  # Excluir la columna de fechas
              Xs.append(v)
              ys.append(y.iloc[i + time_steps - 1])
          return np.array(Xs), np.array(ys)

      # Construir y entrenar el modelo para cada división en TimeSeriesSplit
      for train_index, val_index in tscv.split(train_data):
          train, val = train_data.iloc[train_index], train_data.iloc[val_index]

          # Aplicar la función de creación de dataset
          X_train, y_train = create_dataset(train, train['precioMarginalLocal'], time_steps)
          X_val, y_val = create_dataset(val, val['precioMarginalLocal'], time_steps)

          # Definir y compilar la arquitectura del modelo
          model = tf.keras.models.Sequential([
              tf.keras.layers.LSTM(units=128, input_shape=(time_steps, X_train.shape[2])),
              tf.keras.layers.Dense(units=1)
          ])
          model.compile(loss='mean_squared_error', optimizer=tf.keras.optimizers.Adam(learning_rate=0.011))
```

Figura 4.13 Arquitectura de la red LSTM (1 capa de 128 neuronas y una neurona de salida).

Como sabemos se aplica su validación en este bloque de código (Figura 4.13) donde se aplica "*timeseriesplit*= 3" por lo que los resultados que se muestran a continuación resultan

satisfactorios, no solamente en el nodo 07LPZ-115, porque también fueron similares los resultados en todos los nodos. Véase Figura 4.14 a Figura 4.16:

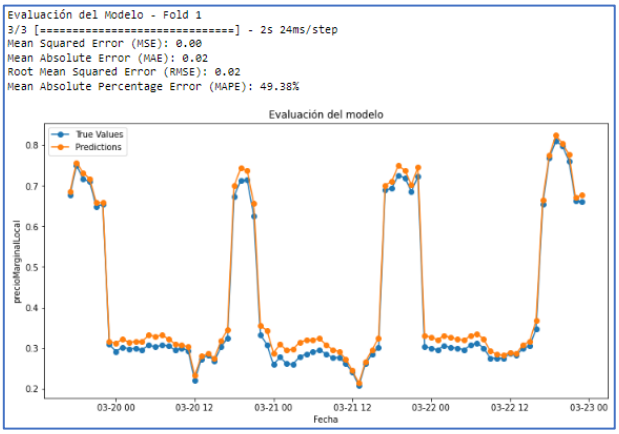


Figura 4.14 Validación del modelo en el primer Split

En la gráfica de la Figura 4.14 se muestra una generalización de la red a los valores reales casi perfecta, por lo que sus métricas respaldan que el pronóstico lo está haciendo bien.

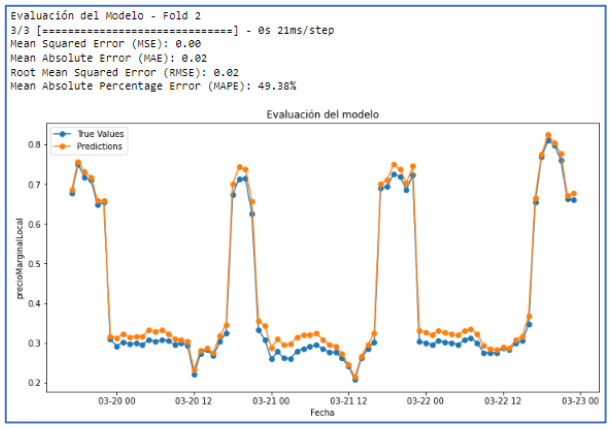


Figura 4.15 Validación del modelo en el segundo Split.

Similarmente en la Figura 4.15 tiene las mismas métricas el Split 2 como el Split 1.

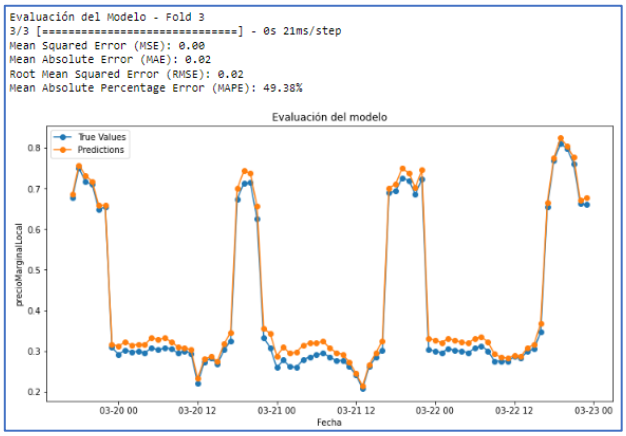


Figura 4.16 Validación del modelo en el tercer Split

En el Split 3 se visualizan los resultados (Figura 4.16), y se declara lo siguiente:

Métricas de error considerablemente bajas:

- Las métricas de error (MSE, MAE, RMSE, MAPE) son todas muy bajas, indican que las predicciones del modelo son muy cercanas a los valores reales.

Consistencia en los *Folds*:

- El hecho de que las métricas sean consistentes a lo largo de los pliegues sugiere que el modelo está generalizando bien y no está sobreajustando o sesgándose demasiado hacia un conjunto específico de datos.

MAPE del 49.38% en los folds:

- El MAPE del 49.38% podría considerarse aceptable dependiendo del contexto en pronósticos de precios; especialmente en el ámbito de la energía, puede ser desafiante obtener MAPE´s muy bajos debido a la naturaleza volátil de los precios.

La evaluación en el conjunto de prueba muestra algunas métricas que son diferentes a las obtenidas en los pliegues de entrenamiento. Las métricas MSE, MAE y RMSE son bajos en el conjunto de prueba, aunque resulta que a veces el MAPE se eleva más de lo esperado. Esto sugiere que, en promedio, las predicciones son cercanas a los valores reales. La evaluación en un conjunto de prueba independiente es crucial para medir el rendimiento real del modelo en datos no vistos. Aunque los pliegues de entrenamiento proporcionan una indicación de la capacidad de generalización, el conjunto de prueba es el verdadero juez de cómo se desempeña el modelo en la práctica. Véase Figura 4.17:

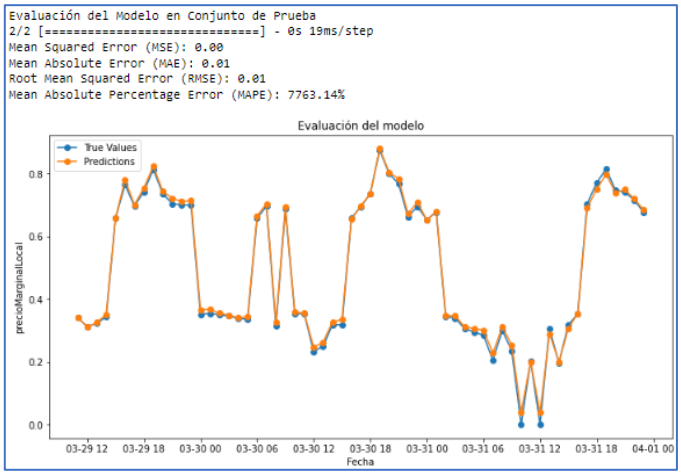


Figura 4.17 Validación del modelo en el conjunto de prueba.

La discrepancia en el MAPE entre los pliegues de entrenamiento y el conjunto de prueba sugiere que los MAPE´s son muy altos en cuanto a la exactitud con la que predice valores a futuro o datos no vistos, lo cual esto es obviamente natural, pues existen múltiples precios posibles y solo cientos de ellos aprendidos del pasado para poder sugerir como serán los próximos en el futuro, pero aprender datos pasados no pueden dar el futuro porque son naturalezas distintas. Por lo tanto, no es decepcionante su métrica del modelo para predecir valores no vistos, pues existe un sinfín de posibles precios los que podrían aparecer en sistemas inestables como la energía, no obstante, es útil visualizar los pronósticos y como generaliza el modelo para dar un veredicto en conjunto con las demás métricas.

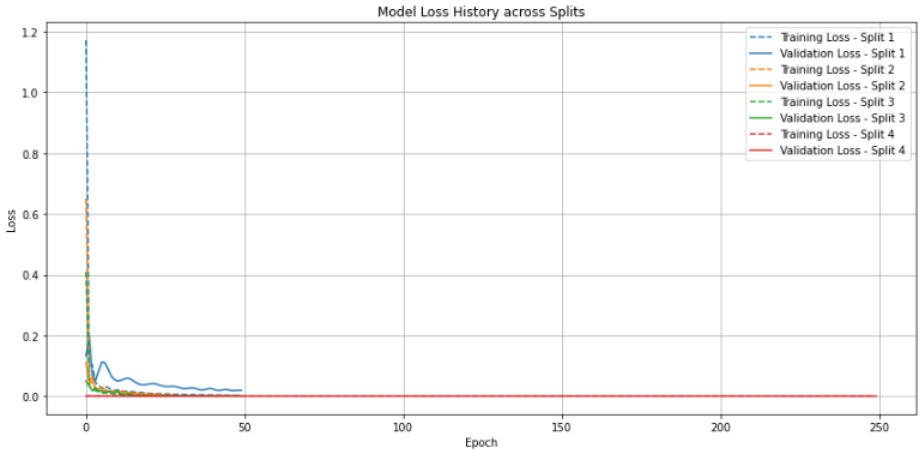


Figura 4.18 Validación del modelo LSTM

Obsérvese en la gráfica (Figura 4.18) que cada Split se redujo tanto a cero que desde antes de 80 ya logro el objetivo (MSE 0 muy cercano a cero) por lo que el modelo aprendió de la base de datos y generalizo muy bien los pronósticos con datos reales Los pronósticos obtenidos para las próximas 48 horas son los mostrados en la Figura 4.19:

```
Pronósticos para el conjunto de prueba:
              FechaHora   Pronostico
0  2023-03-27 12:00:00  2274.794434
1  2023-03-27 13:00:00  2106.889160
2  2023-03-27 14:00:00  2173.525635
3  2023-03-27 15:00:00  2288.994385
4  2023-03-27 16:00:00  4206.666504
..                 ...          ...
56 2023-03-29 20:00:00  5131.555664
57 2023-03-29 21:00:00  4725.865723
58 2023-03-29 22:00:00  4683.608398
59 2023-03-29 23:00:00  4531.918945
60 2023-03-30 00:00:00  4289.137695

[61 rows x 2 columns]

Pronósticos para el conjunto futuro:
              FechaHora   Pronostico
0  2023-04-01 00:00:00  2274.794434
1  2023-04-01 01:00:00  2106.889160
2  2023-04-01 02:00:00  2173.525635
3  2023-04-01 03:00:00  2288.994385
4  2023-04-01 04:00:00  4206.666504
..                 ...          ...
56 2023-04-03 08:00:00  5131.555664
57 2023-04-03 09:00:00  4725.865723
58 2023-04-03 10:00:00  4683.608398
59 2023-04-03 11:00:00  4531.918945
60 2023-04-03 12:00:00  4289.137695

[61 rows x 2 columns]
```

Figura 4.19 Lista de los pronósticos obtenidos del modelo LSTM.

Recordando que otra mejora aplicada es que se le complemente con Prophet; es necesario que disponga de una neurona de salida llamada "*TimeDistributed*" y es fundamental para predecir una secuencia que pueda Prophet, reconocer y lograr el ensamble de ambas herramientas útiles en series temporales. La razón como veremos a continuación para obtener un intervalo de confianza útil para nuestros pronósticos en LSTM fue Prophet. Véase sus métricas en la Figura 4.20:

```
# 3. Calcular métricas (comparando las últimas 48 horas reales con las predicciones)
y_real = df['y'].tail(48).values
y_pred = forecast['yhat'].tail(48).values

mae = mean_absolute_error(y_real, y_pred)
mse = mean_squared_error(y_real, y_pred)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_real - y_pred) / y_real)) * 100

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
print(f"MAPE: {mape}%")
```

```
MAE: 0.1254822389386309
MSE: 0.023799776824629675
RMSE: 0.15427176288818922
MAPE: 2645.9243367048643%
```

Figura 4.20 Métricas del modelo LSTM + Prophet

La proyección a futuro como se muestran en la Figura 4.21 permite saber que tan bien logra generalizar los valores pronosticados y resultan ser bastante similares:

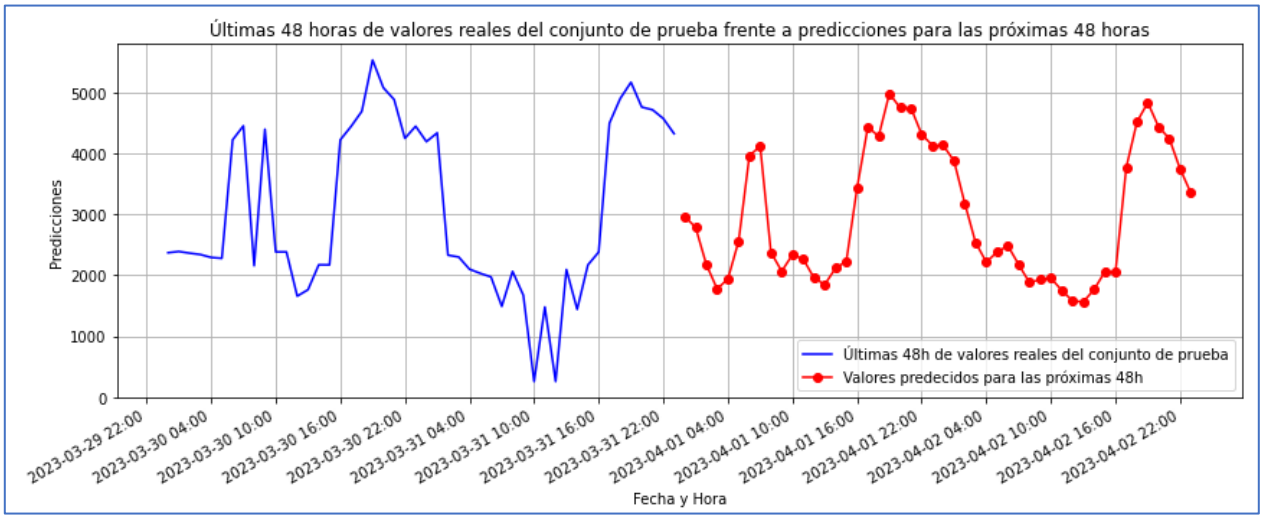


Figura 4.21 Gráfica de los pronósticos a futuro del modelo LSTM.

Además, se hizo una búsqueda en cuadricula para asignar días festivos y parámetros de relevancia para la tendencia, que busquen automáticamente el pronóstico eficiente por considerar, por ejemplo, días de alta demanda (entre semana, por ejemplo), al modelo LSTM + Prophet. Dicha búsqueda abarca 3 parámetros con 4 valores, por lo tanto, el número total de

combinaciones es 4 x 4 x 4 = 64 combinaciones a probar. véase Figura 4.22.

```python
# Definir parámetros para la búsqueda en cuadrícula
param_grid = {
    'changepoint_prior_scale': [0.001, 0.01, 0.1, 0.5],
    'seasonality_prior_scale': [0.01, 0.1, 1.0, 10.0],
    'holidays_prior_scale': [0.01, 0.1, 1.0, 10.0]
}

# Crear una cuadrícula de parámetros
grid = ParameterGrid(param_grid)

best_model = None
best_mape = float('inf')  # inicializar con infinito para luego minimizar

# Búsqueda en la cuadrícula
for params in grid:
    temp_model = Prophet(**params, yearly_seasonality=False, daily_seasonality=True)
    temp_model.fit(df)

    future = temp_model.make_future_dataframe(periods=48, freq='H')
    forecast = temp_model.predict(future)

    y_pred = forecast['yhat'].tail(48).values
    mape_temp = np.mean(np.abs((y_real - y_pred) / y_real)) * 100

    if mape_temp < best_mape:
        best_mape = mape_temp
        best_model = temp_model

print(f"Mejor MAPE: {best_mape}")
```

Figura 4.22 Búsqueda cuadricula para el modelo LSTM + Prophet.

El código realiza una búsqueda en cuadrícula para encontrar los mejores hiperparámetros para el modelo Prophet en función del MAPE. Este itera a través de todas las combinaciones de parámetros y entrena un modelo Prophet para cada combinación. Cada modelo se ajusta a los datos históricos y se utiliza para hacer predicciones futuras. A continuación, los resultados del ensamble de dos poderosos algoritmos de pronóstico, véase Figura 4.23 y Figura 4.24:

	Fecha	Predicción	Límite Inferior	Límite Superior
720	2023-04-01 00:00:00	3112.005631	2193.283494	4038.913784
721	2023-04-01 01:00:00	2770.515998	1844.436235	3736.092496
722	2023-04-01 02:00:00	2517.998256	1660.765271	3393.738719
723	2023-04-01 03:00:00	2440.362533	1537.300040	3325.764496
724	2023-04-01 04:00:00	2572.406561	1680.079793	3476.827607
725	2023-04-01 05:00:00	2813.649044	1853.277718	3698.457952
726	2023-04-01 06:00:00	2963.441176	2103.976816	3896.577917
727	2023-04-01 07:00:00	2861.869027	1912.291430	3788.422816
728	2023-04-01 08:00:00	2512.488324	1604.671978	3448.388620
729	2023-04-01 09:00:00	2073.066336	1205.472347	3027.894809
730	2023-04-01 10:00:00	1724.196337	812.573193	2594.136177
731	2023-04-01 11:00:00	1541.021564	617.754523	2419.545514
732	2023-04-01 12:00:00	1485.305152	514.042457	2402.229779
733	2023-04-01 13:00:00	1511.350722	576.648782	2516.085676

Figura 4.23 Lista de predicciones del PML para el nodo 07LPZ-115 La Paz, Baja California Sur, BCS.

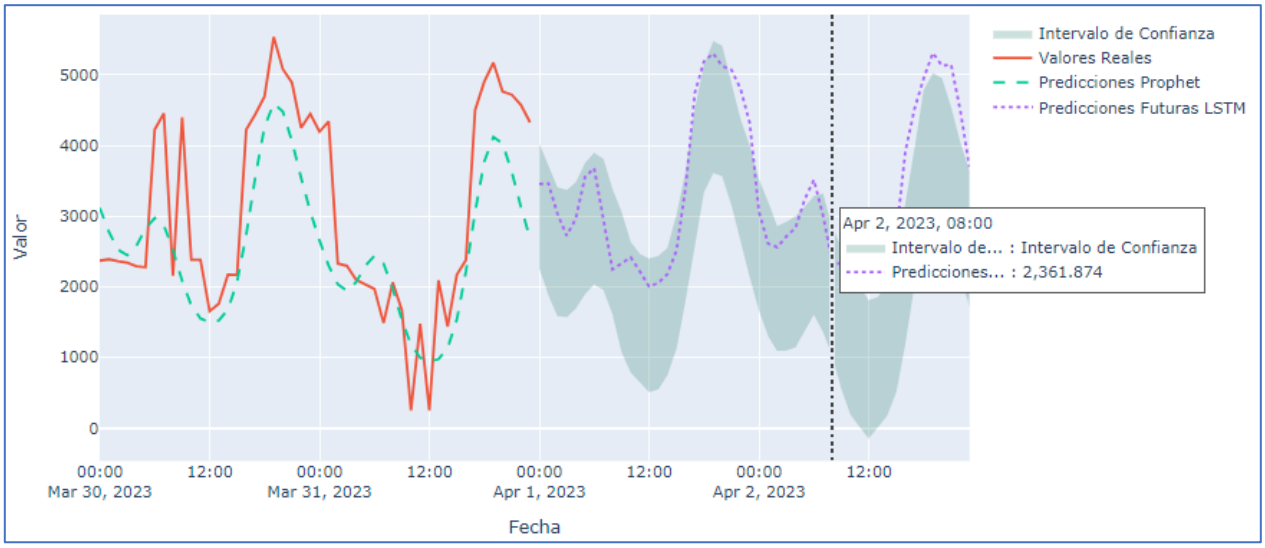


Figura 4.24 Gráfica de predicciones del PML del nodo 07LPZ-115 La Paz, Baja California Sur, BCS.

Así como gráficos interactivos que permiten hacer zoom a los puntos dentro de la Figura 4.25:

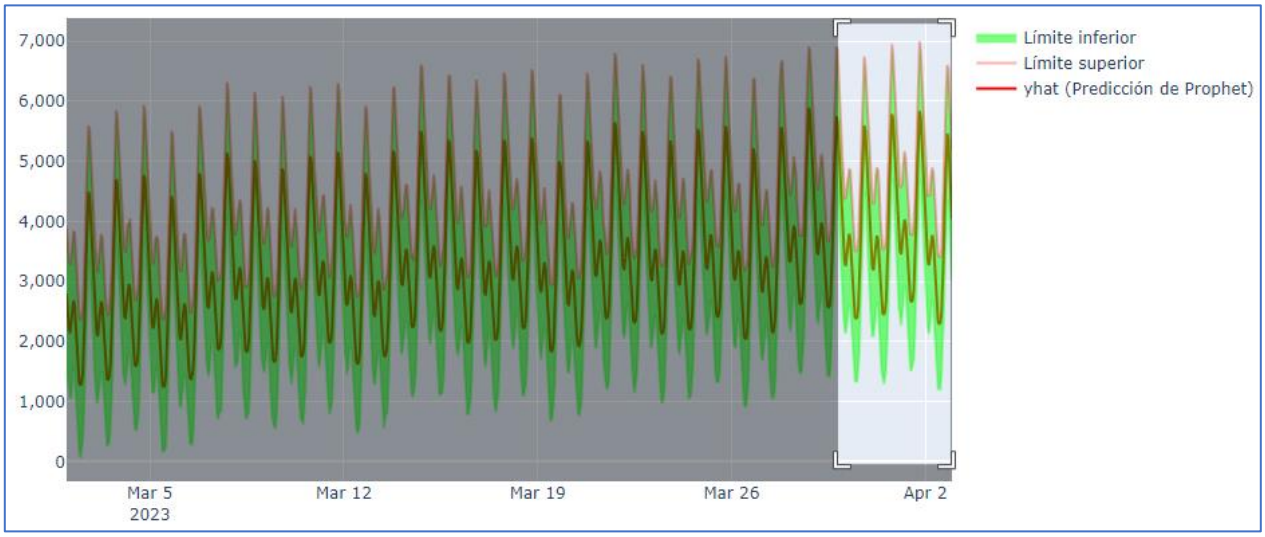


Figura 4.25 Gráfica interactiva en "*Plotly*" de las predicciones del PML del nodo 07LPZ-115.

Para las Figura 4.24 y Figura 4.25 se logran con Prophet (de Facebook), que automáticamente proporciona intervalos de confianza para las predicciones. Estos intervalos de confianza se calculan en base a la tendencia histórica aprendida en el entrenamiento; el intervalo de confianza es una medida de cuánto puede variar una estimación estadística y proporciona una gama de valores dentro de la cual es probable que caiga la observación futura. Las predicciones incluyen tres columnas relevantes para los intervalos de confianza, donde se muestran las predicciones generadas por el modelo LSTM (*Long Short-Term Memory*) en conjunto con el

algoritmo Prophet para el Precio Marginal Local (PML) y cada fila representa una predicción para una hora específica, y las columnas proporcionan la fecha correspondiente, la predicción del PML, así como los límites inferior y superior de la predicción. Véase Figura 4.23

- **Fecha:** Indica la fecha y hora para la cual se realizó la predicción.
- **Predicción:** Representa el valor estimado del PML para la hora correspondiente.
- **Límite Inferior:** Indica el límite inferior del intervalo de confianza asociado a la predicción del PML. Este valor proporciona una estimación de la variabilidad del modelo, y se espera que el valor real del PML esté por encima de este límite con 95% nivel de confianza.
- **Límite Superior:** Representa el límite superior del intervalo de confianza asociado a la predicción del PML. Similar al límite inferior, este valor establece el rango en el cual se espera que el valor real del PML esté con 95% nivel de confianza.

4.4 Tabla comparativa de las métricas de los modelos

Los resultados aplicados a todos los nodos de energía de la investigación se muestran en la siguiente Tabla 4.36 que concentra las métricas obtenidas por el modelo y poder determinar la capacidad de predicción en todos los nodos de la república, véase Tabla 4.36:

Tabla 4.36. Métricas de errores registrados en la implementación a 28 nodos del país.

		MAE					MSE					RMSE					MAPE				
SISTEMA	CLAVE	Arbol D.R	Reg. Pol.	SARIMA	LSTM	PROPHET	Arbol D.R	Reg. Pol.	SARIMA	LSTM	PROPHET	Arbol D.R	Reg. Pol.	SARIMA	LSTM	PROPHET	Arbol D.R	Reg. Pol.	SARIMA	LSTM	PROPHET
SIN	02MMT-400	41.4405	0.0043	537.9088	0.0069	0.1725	6916.956	0.000003	541655.1608	0.00007	0.0508	83.1682	0.006	735.9722	0.0086	0.2254	3.626	0.0006	46.4545	6.4788	328.2249
SIN	06MON-115	77.5268	0.0054	1447.649	0.0406	0.123	33136.87	0.000006	2512380.048	0.0045	0.0314	182.0353	0.008	1585.0489	0.0671	0.1773	17.0954	0.0012	355.0385	166.3587	518.2491
SIN	01TTH-230	59.0408	0.00435	755.3296	0.0332	0.1226	28021.91	0.000006	847216.4049	0.0023	0.0264	167.3974	0.0063	920.4435	0.048	0.1627	4.2772	0.0007	81.7216	22.9482	199.1543
SIN	08SLC-230	52.8738	0.0045	524.2447	0.007	0.1693	10704.58	0.000005	529913.6658	0.00007	0.0474	103.4629	0.0062	727.9516	0.0088	0.2178	4.6049	0.0007	43.7878	4.5753	345.8007
SIN	03AGM-400	43.1906	0.0051	675.1131	0.0151	0.0898	5450.342	0.000004	635331.2236	0.0004	0.0167	73.8264	0.0067	797.0766	0.0203	0.1294	5.1867	0.0008	84.8911	5.5082	65.7652
SIN	08PTE-115	47.3655	0.0048	503.5288	0.0057	0.0348	7018.459	0.000004	469155.6947	0.00006	0.0019	83.7762	0.0065	684.9494	0.0078	0.0439	5.3199	0.0008	49.2297	19.2553	310.1529
BCA	07SAF-115	20.2111	0.004	807.5241	0.028	0.0761	757.0735	0.000003	962394.4138	0.0012	0.0091	27.5149	0.0058	981.017	0.0353	0.0955	1.8332	0.0004	73.3821	8.0228	23.1594
BCA	07CPD-230	51.0956	0.004	620.3987	0.0494	0.0855	9634.895	7659.906	721204.8159	0.0031	0.0105	98.1574	87.5208	849.2377	0.0564	0.1029	8.6876	6.7974	56.2545	13.7006	24.4387
BCA	07TCT-69	11.4224	0.0027	752.5597	0.0211	0.0768	565.5236	0.000004	855381.7222	0.0008	0.0105	23.7807	0.0046	924.8684	0.029	0.1028	0.9403	0.0003	70.7357	5.3302	21.9695
BCA	07VPA-230	51.9494	0.0038	621.8733	0.0432	0.079	9944.47	7659.967	724391.9879	0.0032	0.01	99.7219	87.5212	851.1122	0.0567	0.1	8.6446	6.7723	56.2474	12.2253	23.161
BCS	07LRO-115	47.6165	0.0043	4545.54	0.0091	0.1467	5092.332	0.000004	29000952.05	0.0001	0.0331	71.3605	0.0064	5385.2532	0.0114	0.1822	1.77502	0.0002	233.5067	51.4104	49.4714
BCS	07CAD-115	17.2161	0.0038	4609.394	0.014	0.1782	459.2146	0.000005	29680790.5	0.0002	0.048	21.4292	0.0056	5448.0079	0.0162	0.2191	0.5998	0.0001	214.2511	88.9371	56.4005
BCS	07GAO-115	45.5725	0.0028	4401.108	0.0092	0.1605	4111.436	0.000002	27147794	0.0001	0.0389	64.1204	0.0048	5210.3544	0.0111	0.1974	1.9036	0.0002	234.8511	28.6372	54.6634
BCS	07LPZ-115	23.7731	0.0037	6641.757	0.0045	0.118	6247.49	0.000003	58859458.45	0.00003	0.0218	79.041	0.0057	7671.9918	0.0058	0.1479	0.8284	0.0001	292.1631	80.814	36.8294
BCS	07SNT-115	11.2519	0.0037	4600.364	0.0119	0.1893	210.2421	0.000003	29617618.19	0.0002	0.0528	14.4997	0.0057	5442.2071	0.0157	0.2299	0.4294	0.0001	218.1658	332.8457	58.8845
SIN	08TUM-115	43.7178	0.0029	894.0774	0.0035	0.0773	8086.904	0.000004	3262147.932	0.00005	0.0148	89.9272	0.0044	1806.1417	0.0074	0.1219	4.0026	0.0004	42.222	31.81	573.4953
SIN	06CBD-400	77.9741	0.0063	1397.345	0.056	0.1252	30367.99	0.000007	2334118.316	0.0071	0.031	174.2641	0.0086	1527.7821	0.0845	0.176	16.0759	0.0015	348.8454	167.0837	2314.218
BCA	07RUM-69	23.4918	5.5807	743.1187	0.0575	0.061	1601.336	473.701	842194.0418	0.0045	0.0059	40.0166	21.7646	917.7113	0.0677	0.0771	2.2139	0.7514	70.5532	15.7908	17.8318
SIN	03LVT-69	48.7424	0.0037	359.1993	0.0077	0.0887	10791.45	0.000003	263878.9092	0.00009	0.0146	103.8819	0.0054	513.6914	0.0096	0.1209	4.7089	0.0006	38.42	4.051	89.4614
SIN	05MVI-400	61.6294	0.0052	1178.337	0.0426	0.1132	20700.4	4.9273	1646773.792	0.0034	0.025	143.8763	0.007	1283.2668	0.0589	0.1582	11.0606	0.0015	332.2425	13.5999	85.2353
SIN	05PAR-115	85.0535	0.0057	605.0316	0.0328	0.0819	25616.39	0.000007	545231.827	0.0019	0.0101	160.0512	0.0084	738.3981	0.0442	0.1007	11.542	0.0012	112.4586	12.1817	63.5174
SIN	04GSV-115	65.1067	0.004	440.0455	0.0339	0.0546	12106.02	0.000004	365793.3568	0.0016	0.0063	110.0273	0.0064	604.8085	0.0401	0.0794	8.1548	0.0006	50.5459	8.8889	19.7533
SIN	01AFM-230	40.2828	0.0046	2280.12	0.0228	0.1101	14200.38	0.000004	6667687.794	0.0009	0.0303	119.1653	0.0063	2582.1866	0.0301	0.1741	2.9683	0.0008	404.7835	19.6202	157.7258
SIN	01ANL-85	61.3424	0.006	759.6432	0.0309	0.1243	30275.01	0.000009	863052.5361	0.0025	0.0282	173.9971	0.0096	929.0062	0.0506	0.1681	4.4264	0.00091	80.6032	15.3546	201.2359
SIN	04PLD-230	124.255	0.0034	344.3625	0.0331	0.0683	69786.55	0.000002	197559.5895	0.0017	0.0083	264.1714	0.0052	444.4767	0.0421	0.0914	44.8935	0.0047	1109.71	71.6874	70.8578
SIN	06CEN-138	78.9798	0.0054	755.8242	0.0401	0.0567	43818.68	0.000007	736480.3616	0.003	0.0063	209.3291	0.0084	858.1843	0.0553	0.0799	25.8139	0.0016	162.6107	69.96	50.0497
SIN	03AMX-69	46.9064	0.0036	373.049	0.0137	0.1107	9250.619	0.000003	285536.7824	0.0002	0.023	96.1801	0.0056	534.3564	0.0171	0.1518	4.3341	0.0005	38.7915	7.1294	106.4686
SIN	02CBE-400	39.9155	0.0049	676.2495	0.0128	0.1347	10037.99	0.000004	612856.0327	0.0002	0.0317	100.1897	0.0066	782.8512	0.0149	0.1782	3.3534	0.0007	114.4917	12.3911	231.8003

La Tabla 4.36 nos brinda una perspectiva general de capacidad de prediccion de los

distintos modelos al comparar sus metricas de errores, resaltando el orden del mayor al menor acertivo en porcentaje de MAPE a continuacion:

1. Regresion polinominal
2. Arbol de decision para regresion
3. LSTM
4. LSTM + Prophet
5. SARIMA

La disposición en el orden de los modelos indica simplemente la magnitud de los errores, sin embargo, es fundamental destacar que no representa cual es mejor entre ellos, como por ejemplo, el modelo de regresión polinómica no es la elección más idónea para predecir con precisión las características dinámicas de los precios a futuro. Esto se debe a que sus pronósticos generan una curva de segundo grado, sin el factor temporal asociado a la dinámica de los precios. Por lo que 48 precios del PML a futuro se veran como una recta un poco curvada.

Lo mismo podriamos decir del arbol de decisiones para regresión, pues sus pronósticos a futuro serian muy semejantes (sino es que identicos) a los de una linea curva (como el caso de la regresion polinominal), y se debe a que buscan ajustarse a una regresion numerica minimizando los errores, pero, la ventaja del arbol es que logra determinar variables importantes para un modelo predictivo y ademas permite visualizar la razón de su decision (en este caso el menor MSE para predecir el PML), lo que lo volvio una herramienta util en la investigacion aquí presentada: pero inevitablemente ambos son modelos regresivos y tenderán a una linea los pronosticos a futuro.

Despues podemos mencionar a los algoritmos de series temporales, los cuales tienen el derecho de ser los más adecuados para el pronóstico porque todos ellos capturan los componentes inherentes de las series de tiempo (tendencia, estacionalidad y residuales) que permiten al pronóstico seguir ese patron temporal con sus caracteristicas intrinsecas en sus componentes. Tenemos en primer lugar a las redes LSTM (dense) donde el enfoque de predecir cierto numero de valores a futuro desde su estructura permite a los pronósticos cumplir con los requisitos y metricas. En segundo lugar tenemos al ensamble de LSTM + Prophet (que en la Tabla 4.36 se colocó como nada más PROPHET) porque es más grande en sus errores de pronóstico. Y en

ultimo lugar, tendriamos al mas errado de los algoritmos para predecir en terminos de metricas; el modelo SARIMA, que a pesar de tener autoarima en su busqueda de mejores componentes estacionarios que se asemeja a una búsqueda en cuadrícula, el modelo SARIMA enfrenta dificultades para realizar predicciones precisas en el contexto de los PML. Este método, al ser más un enfoque estadístico tradicional, carece de la capacidad de aprender y retener los patrones específicos presentes en la base de datos. En lugar de ello, evalúa las observaciones en función de medias estadísticas y las compila de manera generalizada para captar una tendencia promedio. Esta naturaleza limitada del modelo SARIMA resulta en pronósticos notoriamente desfasados en comparación con las técnicas más avanzadas como LSTM y Prophet, que demuestran un rendimiento sustancialmente superior.

El modelo más destacado es la red LSTM dense, cuyos pronósticos se caracterizan por su notable precisión y puntualidad. Aunque existe la posibilidad de fallos, ya que seleccionar un valor específico entre infinitas posibilidades en el futuro, es inherentemente desafiante, esta red demostró su eficacia, como se confirmó en los datos de prueba. Las métricas obtenidas reflejan su capacidad superior en comparación con las demás técnicas presentadas. Pero no podemos menospreciar al modelo LSTM secuencial con Prophet, ya que resulta el más razonable en sus pronosticos a futuro y obtener intervalos de confianza por lo que la propuesta definitiva es un ensamble de las ventajas de ambos algoritmos como se demostró en esta investigación.

4.5 Conclusion

El cuarto capítulo revela los frutos de una implementación meticulosa y sistemática. Destacan los hallazgos derivados de la aplicación de árboles de decisión para regresión, junto con el análisis de correlación, que arroja una clara identificación de variables cruciales para mejorar la precisión en la predicción del PML. Asimismo, se presentan los resultados de la validación del modelo con "walkforward", que por la lógica subyacente de su validacion y las gráficas de desempeño avalan una implementación critica del modelo a gran escala. Los resultados de todos los modelos de pronósticos se condensan y exponen en una tabla de métricas, proporcionando una visión ideal para comparar el rendimiento de los modelos en los nodos. Con la culminación de esta etapa, se cumple la metodología, la entrega de resultados, y la evaluación exhaustiva del proyecto. A continuación solo queda exponer la conclusión definitiva del proyecto.

CONCLUSIÓN:

En el marco de esta investigación integral para el pronóstico del Precio Marginal Local (PML) en nodos de energía, se lograron avances significativos y se obtuvieron resultados prometedores. Se aplicó el enfoque de Machine Learning para abordar el desafío del pronóstico del Precio Marginal Local (PML) en nodos de energía. Algunas conclusiones clave son:

1. Integración de Datos Completa: La incorporación adecuada de datos tanto de componentes de energía como de variables climáticas permitió la construcción de un modelo robusto y completo, aprovechando una riqueza de información para la precisión de los pronósticos.
2. Visualizaciones Dinámicas y Análisis Estadísticos Preliminares: La implementación de la interfaz gráfica de usuario "PML Explorer", es una herramienta valiosa de exploración de los datos con visualizaciones dinámicas de *Tkinter* en la plataforma de *jupyter*. Además, la aplicación de análisis estadísticos preliminares ha sido crucial para obtener una comprensión más profunda de los resultados, facilitando la interpretación y validación de los pronósticos.
3. Pronósticos a 48 horas para Nodos representativos: El logro de pronósticos a 48 horas en 28 nodos representa un avance significativo hacia la capacidad de extender estos modelos a los más de 2,300 nodos en la República Mexicana, brindando una perspectiva más completa del sistema de energía de toda la nación como también una herramienta complementaria para el MEM o quien requiera analizar y predecir nodos específicos.
4. Análisis de Sensibilidad y ajuste de hiperparámetros riguroso: La aplicación de análisis de sensibilidad y la búsqueda cuidadosa de hiperparámetros, junto con validación utilizando técnicas como *timeseriesplit* y *walk-forward*, contribuyó a aprobar la robustez y la capacidad de generalización del modelo.
5. Árboles de Decisión para selección de variables: La implementación de árboles de decisión permitió realizar un reconocimiento eficiente de las variables más relevantes para el pronóstico del PML. Esta técnica de Machine Learning facilitó la identificación de factores clave en un conjunto de datos complejo y multidimensional.
6. Redes LSTM dense y Secuencial: La utilización de redes LSTM (*Long Short-Term Memory*) en el contexto de las series temporales permitió capturar patrones secuenciales y

dependencias temporales en los datos. Este enfoque de Machine Learning demostró ser especialmente efectivo para el modelado de series temporales en comparación con enfoques tradicionales.

7. Enfoque Secuencial Integrado: permite a las redes LSTM representar una aplicación avanzada de Machine Learning que mejoró significativamente la credibilidad de los pronósticos. Este enfoque permitió capturar de manera más efectiva las complejidades temporales presentes en los datos de energía.
8. Ensamble de Modelos (LSTM + Prophet): El uso de un ensamble que combina modelos LSTM y Prophet implica una estrategia de Machine Learning más avanzada. Este enfoque no solo mejoró la precisión, sino que también proporcionó una evaluación más completa de la incertidumbre al considerar intervalos de confianza.
9. Ajuste de Hiperparámetros con Búsqueda en Cuadrícula: La búsqueda en cuadrícula de hiperparámetros es una técnica común en Machine Learning que se aplicó en este estudio. Este enfoque riguroso garantizó que los modelos estuvieran optimizados para obtener el mejor rendimiento posible en la tarea de pronóstico.

En conjunto, la aplicación de técnicas de Machine Learning fue esencial para abordar la complejidad inherente de los datos de series temporales y lograr mejoras sustanciales en la precisión de los pronósticos de PML en los nodos de energía. Estos enfoques avanzados demuestran la capacidad de Machine Learning para contribuir significativamente a la resolución de desafíos en el sector energético.

Esta investigación demostró la efectividad de un enfoque integral, desde la integración de datos hasta la implementación de modelos avanzados y técnicas de ajuste, validación y análisis. Los resultados obtenidos sientan las bases para futuros desarrollos y aplicaciones prácticas en el ámbito energético de México.

RECOMENDACIONES

Todo proyecto está sujeto a mejoras, y este no es la excepción. Reconociendo los alcances y logros obtenidos, es valioso también señalar las recomendaciones clave del proyecto, especialmente aquellas que podrían considerarse en trabajos futuros:

1. Mejora Continua de la Interfaz de Usuario: Aunque se han logrado avances significativos con la interfaz gráfica "PML Explorer", es esencial reconocer que este es un prototipo inicial. Para futuros desarrollos, se propone una mejora continua. Un desafío clave es la adaptación de las redes neuronales para prever “n” datos futuros, lo que requiere flexibilidad en la manipulación de “*arrays”* desiguales o incompatibles, lo que podría requerir un proyecto especializado para garantizar flexibilidad y realizar pruebas exhaustivas.
2. Desarrollo de una aplicación de escritorio o Software: Con el objetivo de simplificar el acceso y la implementación de modelos de pronóstico, se sugiere la creación de una aplicación de escritorio o software integral que englobe todas las fases del proyecto. Esta solución proporcionaría una experiencia unificada a los usuarios finales, mejorando la practicidad del sistema.
3. Análisis detallado de nodos muy inestables: La recomendación de enfocarse en nodos con resultados variables en momentos específicos implica un análisis detallado para comprender las razones subyacentes a estas variaciones. Se sugiere explorar técnicas como DBSCAN para clasificar nodos similares, y así identificar similitudes más allá de los valores numéricos. Esto puede incluir factores como actividades locales, cultura u otros elementos no cuantitativos que podrían impactar en el comportamiento de los nodos.

La ejecución de estas mejoras requerirá una inversión sustancial en tiempo, desarrollo y pruebas de campo a mediano plazo ya que fortalecerá la efectividad y eficiencia modelo de pronóstico del PML en un contexto tan diverso y extenso como el de los nodos energéticos.

BIBLIOGRAFÍA

(INEEL), I. N. (2008). *Misión, visión y política de la empresa.* Cuernavaca, Morelos.

A.Livas-García, Tzuc, O., May, E., RasikhTariq, & Torres, M. (2022). Forecasting of locational marginal price components with artificial intelligence and sensitivity analysis: A study under tropical weather and renewable power for the Mexican Southeast. *ELSEVIER*, 206.

Ahmad Amine Loutfi, Mengtao Sun, Ijlal Loutfi, & Solibakke, P. B. (2022). Empirical study of day-ahead electricity spot-price forecasting: Insights into a novel loss function for training neural networks. *ELSEVIER*, 319.

Alonso, & Gabriel. (2017). *Reporte anual del Mercado electrico mayorista 2016.* CDMX, México.: Monitor independiente del Mercado.

Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, & Cerisola, S. (2006). Análisis de la relación precio marginal y demanda de electricidad mediante conglomerados. *X Congreso de Ingenieria de Organización.* Valencia, España.

Antonopoulos, I., Robu, V., Benoit Couraud, Desen Kirli, & Norbu, S. (2020). Artificial intelligence and machine learning approaches to energy demand-side response: a systematic review. *ELSEVIER*, 130.

Canavos, G. C. (1998). *PROBABILIDAD Y ESTADÍSTICA Aplicaciones y métodos.*

CENANCE. (Abril de 2022). *Programa de Ampliación y Modernización de la RNT y de la RGD*. Obtenido de www.cenace.gob.mx: https://www.cenace.gob.mx/Paginas/SIM/ProgramaRNT_RDG.aspx

Deepak Singhal, & Swarup, K. (2011). Electricity price forecasting using artificial neural networks. *ELSEVIER*, 550-555.

Díaz Ramírez, J. (2021). Aprendizaje Automático y Aprendizaje Profundo. *Ingeniare. Revista chilena de ingeniería, vol. 29 N°2*, 182-183. Recuperado de https://www.scielo.cl/pdf/ingeniare/v29n2/0718-3305-ingeniare-29-02-180.pdf.

Elena Dumitrescu, Sullivan Hué, Christophe Hurlin, & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *ELSEVIER*, 1178-11192.

George F. Luger, & Stubblefield, W. A. (1993). *Artificial Intelligence.* Benjamin/Cummings Publishing Company.

Haugeland, J. (1985). *La inteligencia artificial.* siglo XXI editores.

HuiLiu∗, & ZhihaoLong. (2020). An improved deep learning model for predicting stock market price time series. *ELSEVIER*, 1-16.

Hweon Ki, J., Song Hyun, K., & Chang Lak, K. (2022). Proposal of a new method for learning of diesel generator sounds and detecting abnormal sounds using an unsupervised deep learning algorithm. *Nuclear Engineering and Technology, ISSN 1738-5733,* Recuperado de https://doi.org/10.1016/j.net.2022.10.019.

Indranil Ghosh, Chaudhuri, T. D., Alfaro-Cortés, E., & Gámez, M. (2022). A hybrid approach to forecasting futures prices with simultaneous consideration of optimality in ensemble feature selection and advanced artificial intelligence. *ELSEVIER*, 181.

Instituto Nacional de Electricidad y Energías Limpias. (s.f.). Obtenido de Gobierno de México: https://www.gob.mx/ineel/que-hacemos

Instituto Nacional de Electricidad y Energías Limpias. (2023). Obtenido de Gobierno de México: https://www.gob.mx/ineel/que-hacemos

Jeovani E. Santiago López, & García, F. G. (2018). Precios Marginales Locales: Análisis de Volatilidad del MEM. *RVP-AI/2018* (pág. 18). Acapulco Guerrero: IEEE.

Jiawei Long, Zhaopeng Chen, Weibing He, Taiyu Wu, & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application chinese stock exchange market. *ELSEVIER*, 5-9.

Jincheng Zhang, Xiaowei Zhao, Siya Jin, & Greaves, D. (2022). Phase-resolved real-time ocean wave prediction with quantified uncertainty based on variational Bayesian machine learning. *ELSEVIER*, 234.

Kyung Keun Yun, Sang Won Yoon, & Won, D. (2022). Interpretable stock price forecasting model using genetic algorithm-machine learning regressions and best feature subset selection. *ELSEVIER*, 118803.

Li, C., Yu, R., Yu, W., & Wang, T. (2022). Fault-tolerant control system for once-through steam generator based on reinforcement learning algorithm. *Nuclear Engineering and Technology, Volume 54, Issue 9, ISSN 1738-5733*, 3283-3292. Recuperado de https://doi.org/10.1016/j.net.2022.04.014.

Lima Álvarez, M., Ramírez Fiallo, O., Fernández García, S., & Pérez Barrios, R. J. (2011). Sistema para diagnóstico de fallas en transformadores a través de lógica difusa. *Revista Científica de Ingeniería Energética. Vol. XXIV, No. 3/2003*, 34-41. Recuperado de https://www.researchgate.net/publication/266484172_Sistema_para_diagnostico_de_fallas_en_transformadores_a_traves_de_logica_difusa.

Md Faisal kabir, Tianjie Chen, & Ludwig, S. A. (2022). A performance analysis of dimensionality reduction algorithms in machine learning models for cancer prediction. *ELSEVIER*, 1001125.

Moreno, A., Armengol, E., Béjar, J., Belanche, L., & Cortés, U. (1998). Aprendizaje automático. *Edicions UPC*, 6-15. Recuperado de https://upcommons.upc.edu/bitstream/handle/2099.3/36157/9788483019962.pdf?sequence=1&isAllowed=y.

Nexus.Energia.mx. (2021). *informe del Mercado Eléctrico Mayorista Diciembre 2021.* Ciudad de México: Nexus energia mx.

Pintarelli, M. B. (s.f.). Intervalos de confianza.

Ponce, P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería.* México: Alfaomega Grupo Editor, S. A. de C. V.

SatvikKhuntia, AtharHanif, QadeerAhmed, JohnLahti, & MaartenMeijer. (2022). Cabin Load Prediction Using Time Series Forecasting in Long-haul Trucks for Optimal Energy Management. *ELSEVIER*, 342-347.

Wang, M., Zhao, L., Duc, R., Wang, C., & Chen, L. (2018). A novel hybrid method of forecasting crude oil prices using complex network science and artificial intelligence algorithms. *ELSEVIER*, 480-495.

WEI Wei, & Chuan, J. (2022). A combination forecasting method of grey neural network based on genetic algorithm. *ELSVIER*, 191-196.

Xiyun Yang, Liwei Fan, Xiangjun Li, & Meng, L. (2022). Day-ahead and real-time market bidding and scheduling strategy for wind power participation based on shared energy storage. *ELSEVIER*, 108903.

Žarković, M., & Stojković, Z. (2017). Analysis of artificial intelligence expert systems for power transformer condition monitoring and diagnostics. *Electric Power Systems Research, Volume 149, ISSN 0378-7796*, 125-136. Recuperado de https://doi.org/10.1016/j.epsr.2017.04.025.

Zexian Sun, Mingyu Zhao, & Zhao, G. (2022). Hybrid model based on VMD decomposition, clustering analysis, long short memory network, ensemble learning and error complementation for short-term wind speed forecasting assisted by Flink platform. *ELSEVIER*, 261.