

UNIVERSIDAD NACIONAL DE ASUNCIÓN

**Facultad de Ingeniería
Ingeniería Mecatrónica**



**Diseño de un modelo de Deep Learning basado en la
arquitectura Temporal Fusion Transformer para la predicción
horaria a corto plazo de la demanda eléctrica nacional**

Silvio Jose Aguilar Velazco
Santiago Dionisio Vargas Garcia

San Lorenzo - Paraguay

2023

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Miembros del Consejo Directivo

Prof. Dr. Ing. Rubén Alcides López Santacruz (Decano)

Prof. MSc. Ing. Cirilo Jorge Hernáez Medina (Vice-Decano)

Consejeros Titulares

Prof. Ing. Verónica Alexandra Blanco Bogado (Docente)

Prof. MSc. Ing. Cáceres Arce, Marco Anibal (Docente)

Prof. Dra. Ing. Saito, Miki (Docente)

Prof. Ing. Blanco Bogado, Veronica (Docente)

Prof. Ing. Poisson Spessot, Luis Vidal (Docente)

Prof. Ing. Arrom Irun, Francisco Manuel (Docente)

Ing. Segovia Lohse, Herbert Raul (Graduado)

Ing. Arias Araujo, Rodrigo Miguel (Graduado)

Ing. Stanley Lucero, Nelson Junior (Graduado)

Rodriguez Mendieta, Keyla Bianca (Estudiante)

Grau Maciel, Samuel Octavio (Estudiante)

Simon Orue, Victoria Montserrat (Estudiante)

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Consejeros Suplentes

Prof. Ing. Yambay Ferreira, Hassim Daniel (Docente)

Prof. Dr. Ing. Stalder Diaz, Diego Herbin (Docente)

Prof. Dr. Ing. Segovia Lohse, Hermann Rigoberto (Docente)

Prof. Ing. Sosa Almirón, Luis Antonio (Docente)

Prof. Ing. Duarte Rodas, Helio Ramon (Docente)

Prof. Ing. Stanley Calo, Henry Williams (Docente)

Ing. Gómez Agüero, Leticia Belen Beatriz (Graduada)

Ing. Flor De Rivas, Priscilla Tamara (Graduada)

Ing. Alfonso Duarte, Fausto Ariel (Graduado)

Agüero Gauto, Fatima Maria Andrea (Estudiante)

Machuca Recalde, Arturo (Estudiante)

Valdez Samudio, Gabriela Maria Alexandra (Estudiante)

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Agradecimientos

Con inmensa gratitud, dedico este trabajo a mi querida familia. Su amor incondicional y apoyo constante han sido la base de mi fortaleza y perseverancia. Gracias por ser mi mayor inspiración. Mis sinceros agradecimientos a mis tutores, Diego, Félix y Carlos, por su guía experta, paciencia y motivación. Su sabiduría ha sido crucial en mi trayectoria académica y en la realización de este Trabajo Final de Grado. Un reconocimiento especial a mi compañero y futuro colega Santiago, cuya amistad y colaboración han sido fundamentales en este viaje. Su apoyo y aportaciones han enriquecido enormemente este trabajo. Además, extendiendo mi gratitud a todos y cada uno de mis amigos, cuya compañía, entendimiento y aliento han sido esenciales en los momentos más desafiantes de esta aventura académica. Su presencia ha sido un regalo invaluable.

Silvio Aguilar

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Agradecimientos

A nuestros tutores Dr Diego Stalder, Msc Carlos Sauer y al Ing. Félix Morales cuya dirección fue indispensable para este trabajo. A mis compañeros y amigos de la carrera, quienes hicieron este camino mas ameno en especial a mi amigo y futuro colega Silvio. A mis padres, hermanos y a Dios.

Santiago Vargas

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Índice general

| | |
|--|-----------|
| 1. Introducción | 8 |
| 1.1. Presentación | 8 |
| 1.2. Objetivo General | 9 |
| 1.3. Objetivos Específicos | 10 |
| 1.4. Alcance y limitaciones | 10 |
| 2. Descripción del sector eléctrico del Paraguay | 13 |
| 2.1. Modelo del Sector Eléctrico del Paraguay | 13 |
| 2.2. Demanda eléctrica nacional | 16 |
| 2.2.1. Demanda anual | 18 |
| 2.2.2. Demanda mensual | 19 |
| 2.2.3. Demanda diaria | 20 |
| 2.2.4. Demanda horaria | 20 |
| 2.2.5. Sistema Interconectado Nacional | 21 |
| 2.2.6. Parque de Generación Existente | 23 |
| 2.3. Predicción de la demanda eléctrica | 25 |
| 2.4. Modelos Estadísticos vs. Modelos de Inteligencia Artificial | 26 |
| 2.5. Modelos basados en Inteligencia Artificial | 27 |
| 2.5.1. Redes Convolucionales | 29 |
| 2.5.2. Redes Neuronales Recurrentes | 30 |
| 2.5.3. Memoria a Corto y Largo Plazo (LSTM) | 30 |
| 2.5.4. Modelos basados en Atención | 31 |
| 2.6. Modelos De Pronóstico de Múltiples Horizontes | 36 |
| 2.7. Fundamentos teóricos de la Arquitectura TFT | 39 |
| 2.7.1. Mecanismos de Control en TFT | 41 |
| 2.7.2. Redes de selección de variables | 43 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

| | |
|---|-----------|
| 2.7.3. Codificador estático de covariables | 45 |
| 2.7.4. Atención Multi-Cabecal Interpretativa | 46 |
| 2.7.5. Temporal Fusion Decoder | 47 |
| 2.7.6. Capa de enriquecimiento estático | 49 |
| 2.7.7. Capa de Avance Posicional | 50 |
| 2.7.8. Salidas Cuantiles | 52 |
| 2.7.9. Interpretabilidad | 52 |
| 3. Metodología Aplicada | 55 |
| 3.1. Obtención y preparación de datos | 55 |
| 3.2. Preprocesamiento de Datos | 57 |
| 3.3. Modelos de Aprendizaje Profundo | 61 |
| 3.4. Implementación en Darts | 62 |
| 3.5. Modelo TFT | 62 |
| 3.6. Definición de las covariables | 63 |
| 3.7. Optimización de velocidad con EarlyStopper | 65 |
| 4. Análisis e Interpretación de los Resultados | 66 |
| 4.1. Análisis de convergencia con EarlyStopper | 66 |
| 4.2. Impacto de la covariable Feriados | 68 |
| 4.3. Optimización de hiperparámetros con Optuna | 72 |
| 4.3.1. Optimización con tiempo de parada | 73 |
| 4.3.2. Optimización con límite de iteraciones | 73 |
| 4.4. Análisis de los modelos TFT | 75 |
| 4.4.1. Análisis trimestral | 76 |
| 4.4.2. Análisis por día de la semana | 77 |
| 4.4.3. Análisis y evaluación del mejor modelo | 77 |
| 4.5. Comparativa de resultados | 80 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

| | |
|---|-----------|
| 5. Conclusiones y Trabajos Futuros | 83 |
| 5.1. Conclusiones | 83 |
| 5.2. Trabajos Futuros | 84 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Índice de figuras

| | |
|---|----|
| 2.1. Mapa Eléctrico del Sistema Metropolitano. | 15 |
| 2.2. Sistemas regionales del SIN. | 16 |
| 2.3. Histórico de demanda máxima | 17 |
| 2.4. Distribución de demanda por sistemas 2021-2022. | 18 |
| 2.5. Inversión por sistemas 2021-2030. | 18 |
| 2.6. Proyección de Demanda Máxima del SIN utilizada. | 19 |
| 2.7. Curvas de Carga del Día de la Demanda Máxima | 21 |
| 2.8. Destino de la energía eléctrica. | 22 |
| 2.9. Mapa eléctrico del Paraguay (2022). | 23 |
| 2.10. Origen de la energía eléctrica. | 24 |
| 2.11. Despacho proyectado por el Plan Maestro de Transmisión (PMT) 2021-2030. | 25 |
| 2.12. Arquitecturas de codificadores utilizadas en series temporales. . . | 28 |
| 2.13. Atención escalada del producto escalar. | 33 |
| 2.14. Arquitectura Transformer. | 35 |
| 2.15. Ilustración de una predicción a múltiples horizontes. | 36 |
| 2.16. Métodos iterativos vs Directos | 37 |
| 2.17. Arquitectura TFT. | 40 |
| 2.18. GRN | 42 |
| 3.1. Esquema general de la metodología aplicada. | 56 |
| 3.2. Representación gráfica de la variable Mes y sus transformaciones senoidales y cosenoidales. | 58 |
| 3.3. Representación gráfica de la variable Día de la Semana y sus transformaciones senoidales y cosenoidales | 59 |
| 3.4. Gráfico Box-Plot del SIN para las variables temporales calendario | 60 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

| | |
|--|----|
| 3.5. Distribución de los días feriados en los primeros cinco años del conjunto de datos. | 61 |
| 4.1. Comparación de valores observados con predichos del mejor día 16-08-2020 para el modelo inicial entrenado | 67 |
| 4.2. Comparación del mismo modelo con funciones de validación distintas para el EarlyStopper. | 68 |
| 4.3. Distribución del SIN en forma de Caja para Feriados y No Feriados | 69 |
| 4.4. Distribución del SIN en forma de Violin para Feriados y No Feriados | 70 |
| 4.5. Comparación de resultados para modelo con y sin covariable Feriado en la primera semana de Enero. | 71 |
| 4.6. Comparación de resultados para modelo con y sin covariable Feriado en la última semana de Diciembre. | 71 |
| 4.7. Gráficos de explicabilidad de las variables de entrenamiento del modelo | 72 |
| 4.8. Evolución del valor del SMAPE para cada modelo en la optimización por Optuna. | 75 |
| 4.9. Histograma del error pico porcentual por hora. | 80 |
| 4.10. Histograma del máximo error porcentual del día. | 80 |
| 4.11. Comparativa en los mejores días de los valores del mejor modelo con los valores reales. | 81 |
| 4.12. Comparativa en la mejor semana de los valores del mejor modelo con los valores reales. | 82 |
| 4.13. Comparativa en la mejor semana de los valores del mejor modelo con los valores reales. | 82 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Índice de tablas

| | |
|---|----|
| 2.1. Producción y consumo de energía en Paraguay, 2021-2022 | 19 |
| 3.1. Descripción de hiperparámetros en la arquitectura TFT. | 63 |
| 3.2. Valores iniciales de hiperparámetros en la arquitectura TFT. . . . | 64 |
| 3.3. Definición de las covariables. | 64 |
| 4.1. Métricas generales del primer modelo | 67 |
| 4.2. Comparación de métricas con funciones distintas de validación. . | 68 |
| 4.3. Comparación de métricas para distintos valores de paciencia y Δ usando el MAPE como función de validación. | 69 |
| 4.4. Comparación de modelos sin covariable Feriados vs modelo con covariable Feriados. | 70 |
| 4.5. Intervalo de valores utilizados en la optimización de hiperparámetros. | 73 |
| 4.6. Mejores conjuntos de hiperparámetros. | 76 |
| 4.7. Métricas aplicadas a todo el conjunto de prueba. | 76 |
| 4.8. Métricas de los modelos por trimestre. | 77 |
| 4.9. Métricas de los modelos por día de la semana. | 78 |
| 4.10. Métricas por hora del conjunto de prueba. | 79 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Índice de anexos

| | |
|---|-----|
| Anexo A: Definiciones complementarias | 89 |
| Anexo B: Documentación de la API DARTS | 93 |
| Anexo C: Código ejemplo para entrenamiento TFT en lenguaje Python | 103 |

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

CAPÍTULO 1

Introducción

1.1. Presentación

La Administración Nacional de Energía (ANDE) tiene como una de sus prioridades principales la de satisfacer de forma adecuada las necesidades de energía eléctrica del país, teniendo en cuenta la previsión de la demanda de consumo eléctrico a nivel nacional, factor primordial para poder tomar las mejores y más eficientes decisiones en ámbitos técnicos, administrativos y operativos [1]. Podemos destacar dos principales problemas que se pueden presentar si no se realiza de forma correcta este proceso: por un lado, la adquisición excesiva e innecesaria de energía, que puede desencadenar en pérdidas económicas para la entidad, o por otro lado, no contar con la cantidad necesaria para satisfacer la demanda mínima, repercutiendo directamente en los gastos del consumidor ya que se deberá recurrir a generación de contingencia, e incluso en un caso peor, generar apagones generales a niveles relativamente grandes. Cabe destacar además que toda la energía eléctrica que no se consume a nivel nacional es cedida o vendida a países vecinos, teniendo en cuenta que estos acuerdos no son necesariamente bien aprovechados en beneficio de nuestro país.

La planificación que realiza la ANDE, en distintas escalas de tiempo (diario, mensual y anual), y para distintos escenarios cuenta con cierta incertidumbre, considerando distintos factores que afectan directa e indirectamente a la demanda que se quiere predecir.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

Existen modelos de previsión de carga basados en la estadística, considerados tradicionales, como la regresión lineal, así como modelos semi-analíticos compuestos a partir del perfil de consumo de clientes, e incluso, modelos basados en redes neuronales profundas [2]. Estos modelos fueron utilizados previamente para la predicción de demanda eléctrica, siendo considerados impulsores del estudio en el área. En el mismo se establece que no existe un consenso que determine el uso particular de un modelo o método por encima de otro, generalmente suele ser resultado de las métricas de desempeño para seleccionar el mejor modelo, por lo cual la práctica común es determinar qué modelo se ajusta mejor para una determinada situación.

Como complemento, el estudio y aplicación de redes neuronales se ha convertido en una atracción para los investigadores del área, ya que han sido utilizadas para predicción de tareas que incluyen el pronóstico de carga a corto plazo (STLF: Short Term Load Forecasting) en microrredes [3]. En 2017 se introduce la arquitectura Transformer por parte de un grupo de investigadores de Google [4], creando una revolución en el área de procesamiento natural del lenguaje, implementando mecanismos de atención que son utilizados para determinar la importancia o peso de una palabra durante el proceso de traducción a otro idioma. La misma tiene una variante que se puede aplicar a la predicción de series temporales, denominada arquitectura Temporal Fusion Transformer (TFT) [5]. Esta última será tomada como línea de base para la realización de este Trabajo Final de Grado.

1.2. Objetivo General

Diseñar un modelo inteligente TFT (Temporal Fusion Transformer) para la predicción horaria a corto plazo de la demanda eléctrica nacional.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

1.3. Objetivos Específicos

- Actualizar datos históricos de suministro de energía proveída por la ANDE.
- Validar datos de variables meteorológicas a nivel país e imputar datos faltantes.
- Implementar un modelo de predicción de demanda de energía basado en la arquitectura TFT.
- Realizar un ajuste fino de los hiper-parámetros.
- Comparar diferentes modelos basados en la Deep Learning para forecasting (previsión), incluyendo trabajos finales de grado de compañeros de FIUNA relacionados al área.

1.4. Alcance y limitaciones

Las redes del Sistema Interconectado Nacional abarcan un conjunto de líneas y subestaciones, incluyendo sus equipos asociados, que transportan la energía desde las plantas de generación a las subestaciones de transformación y finalmente al consumidor final. La planificación de su operación debe asegurar un suministro de energía suficiente y de bajo costo, teniendo en cuenta diversos aspectos y factores, como la cantidad de generadores inhabilitados o equipos que no se encuentran en funcionamiento, además de factores externos y no controlables como la inestabilidad climática o los días especiales durante el año (fines de semana y feriados). Es posible afirmar en base a esto, que la demanda de potencia a nivel global en el país condiciona directamente la contratación de potencia a las centrales eléctricas, que abarcaría la mayor parte del costo operativo de la ANDE.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Teniendo en cuenta el campo y el ámbito en el que se profundizará, existen trabajos realizados con anterioridad que incursionaron en el diseño de modelos para predicción de demanda energética, basándose en la LSTM (long-short term memory), una arquitectura evolucionada de las Redes Neuronales Recurrentes, además de otros que lo aplicaron en el entorno MatLab Neural Network Toolbox, considerando factores meteorológicos y de calendario [3, 6, 7].

Si bien, en los modelos de predicción en el entorno de MatLab se arrojaron de forma satisfactoria tendencias de la curva en una buena parte de los días del año, la precisión de los mismos se veía afectada en picos altos y bajos en ciertos periodos, por motivos que pueden estar relacionados al clima o bien, información ineficaz a la hora de ser agrupada [10,11]. Además, utiliza la temperatura como atributo para la predicción, sin tener en cuenta otros factores meteorológicos o fiestas festivas u otros factores que podrían influir en la demanda eléctrica.

Aprovechando la optimización realizada en el enfoque del modelo LSTM (Long Short-Term Memory)[9], el método a utilizar en este Trabajo Final de Grado será el de pronóstico de múltiples horizontes, mediante series temporales y Machine Learning, cuya función es predecir múltiples pasos en el futuro permitiendo al usuario acceso a información durante un periodo de tiempo a diferencia de las predicciones de un punto por paso. De forma más específica, el modelo con el que se trabajará será una red TFT [5]. Su diseño se realizará utilizando softwares de carácter open-source, en un entorno de Python con sus respectivas librerías específicas para Machine Learning, teniendo en cuenta también otras arquitecturas como soporte, además de utilizar como un constante respaldo otros métodos estadísticos junto con los datos históricos existentes del SIN.

El uso de la arquitectura TFT permitirá acceder a un nivel de explicación e interpretabilidad superior al modelo presentado un año antes en cuanto al comportamiento dinámico de las series temporales, mediante el uso de mapas de calor en base a los mecanismos de atención [5].

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Se espera que este Trabajo Final de Grado pueda servir como herramienta eficiente de proyección, logrando así un equilibrio entre oferta y demanda en las distintas operaciones referentes al despacho de carga de la ANDE, ya que desde ese sector se podrá hacer uso de las curvas obtenidas por el modelo de la demanda predicha y tomar en base a las mismas las mejores decisiones operativas. Es importante mencionar que, así como se partió desde la concepción de la idea de este trabajo, el mismo puede servir como base para próximas investigaciones que buscan optimizar las mencionadas predicciones en distintos plazos.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

CAPÍTULO 2

Descripción del sector eléctrico del Paraguay

2.1. Modelo del Sector Eléctrico del Paraguay

En la mayoría de los países, los constantes cambios que paulatinamente van apareciendo en el mercado energético están obligando a rediseñar los modelos de gestión de la producción, distribución y comercialización y a desarrollar estudios de previsión de la demanda y la oferta a corto, largo o medio plazo. La cantidad de suministro está determinada por la cantidad de generación de energía y el suministro de energía en el país. La demanda cubre todas las necesidades sociales y económicas del consumidor final, por lo que éste dependerá de su comportamiento durante muchos años. Por lo tanto, se debe desarrollar una estrategia relativa que utilice estos dos parámetros para una gestión energética exitosa [8]. La Administración Nacional de Electricidad (ANDE) es un organismo nacional descentralizado creado por la Ley Nro 996 del 12 de agosto de 1964, que tiene el monopolio de la generación, transmisión, distribución y venta de energía eléctrica en el territorio del Paraguay. La ANDE es responsable de preparar los planes y programas para el desarrollo eléctrico desde la generación hasta la interconexión de los sistemas nacionales, incluyendo el diseño, construcción y adquisición de obras, equipos y bienes necesarios para la normal operación de los servicios eléctricos. La ANDE capitaliza la explotación de los recursos hidráulicos, que constituyen la principal fuente de energía eléctrica en el país. Cuenta con la central hidroeléctrica nacional de Acaray, la cual lleva a cabo

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

sus operaciones desde 1968. Además, administra dos centrales hidroeléctricas binacionales en conjunto con otros países: la central de Itaipú, compartida con la República Federativa del Brasil y operativa desde 1984, y la central de Yacyretá, operando con su contraparte argentina desde 1994. No obstante, la ANDE tiene proyecciones ambiciosas a corto plazo, entre las que se incluyen la instalación de futuras centrales hidroeléctricas a pequeña escala y la implementación de dos líneas de transmisión de 500 kV que conectarían Itaipú y Yacyretá con la subestación de Villa Hayes [1]. Esta iniciativa busca lograr la interconexión integral del sistema eléctrico nacional, con el propósito de mejorar de manera significativa la cobertura y la estabilidad en todo el territorio. Con el objetivo de lograr una distribución operativa más eficiente, la ANDE se apoya en su mapa eléctrico interno, el cual está compuesto por divisiones regionales estratégicamente establecidas y se pueden observar en la Figura 2.2. Estas divisiones permiten un control más efectivo de las operaciones en todo el sistema.

- **Sistema Este:** donde se encuentran la central hidroeléctrica binacional de Itaipú y la central hidroeléctrica de Acaray. Se vincula eléctricamente con el Sistema Metropolitano por medio de una línea de transmisión de 500 kV.
- **Sistema Metropolitano:** donde se encuentra aproximadamente el 60 % del consumo nacional, alimentado por una línea de transmisión de 500 kV del sistema este, 5 líneas de transmisión de 220 kV provenientes del sistema central y 2 líneas de transmisión de 220 kV desde el sistema sur el cual puede ser observado a detalle en la Figura 2.1.
- **Sistema Sur:** donde se encuentra la central hidroeléctrica binacional de Yacyretá, se vincula mediante 2 líneas de transmisión de 220 kV al sistema metropolitano.
- **Sistema Central:** situado entre el sistema metropolitano y el sistema este. Del mismo parte una línea de transmisión de 220 kV que llega hasta la

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

ciudad de Loma Plata, del sistema oeste, en el centro del Chaco Paraguayo, pasando por el sistema norte.

- **Sistema Norte:** alimentando en forma radial desde el sistema central por una línea de transmisión de 220 kV, que llega al sistema oeste.
- **Sistema Oeste:** alimentado por una línea de transmisión de 220 kV desde el sistema central que pasa por el sistema norte.

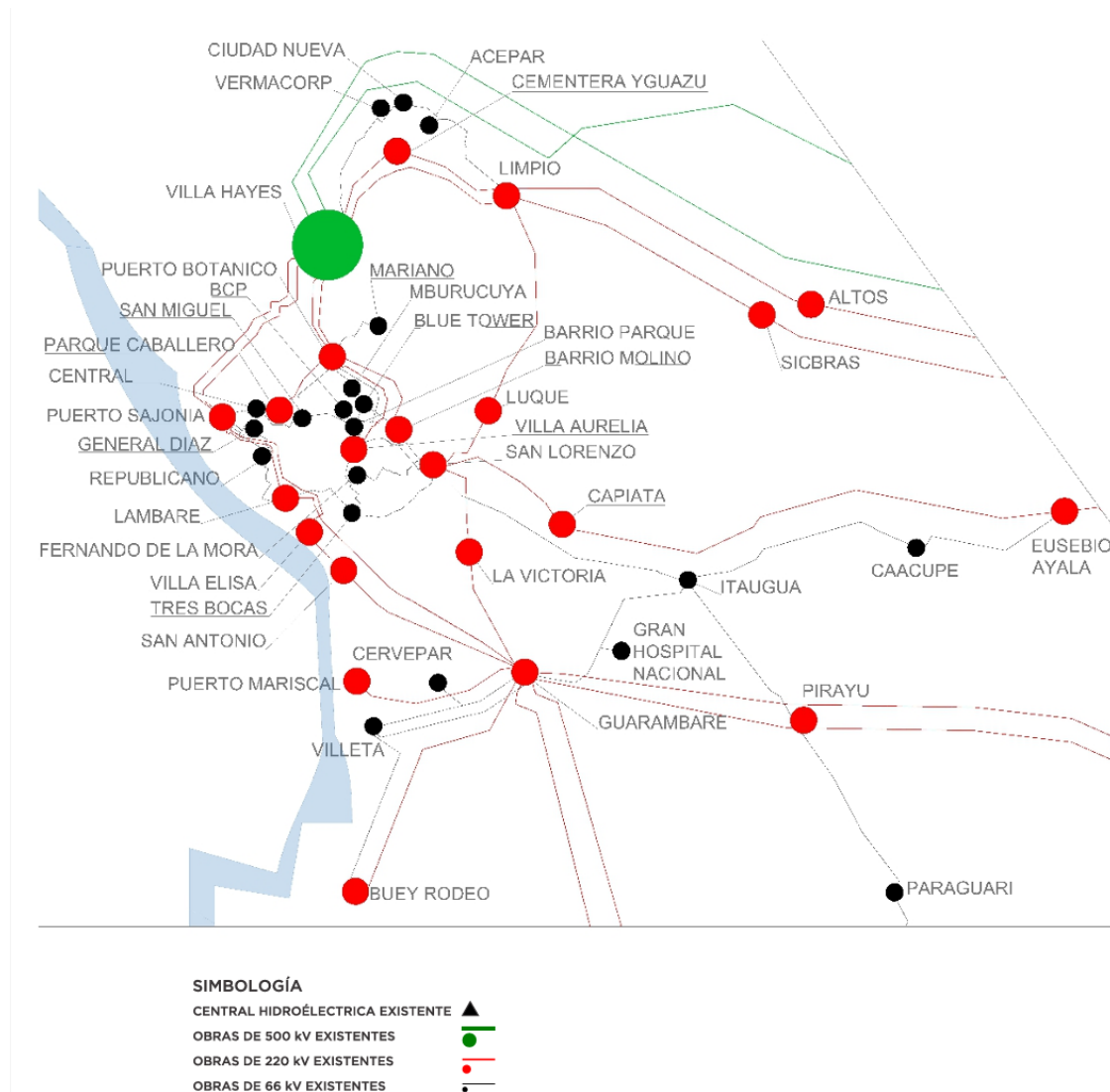


Figura 2.1: Mapa Eléctrico del Sistema Metropolitano.

Fuente: [9]

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

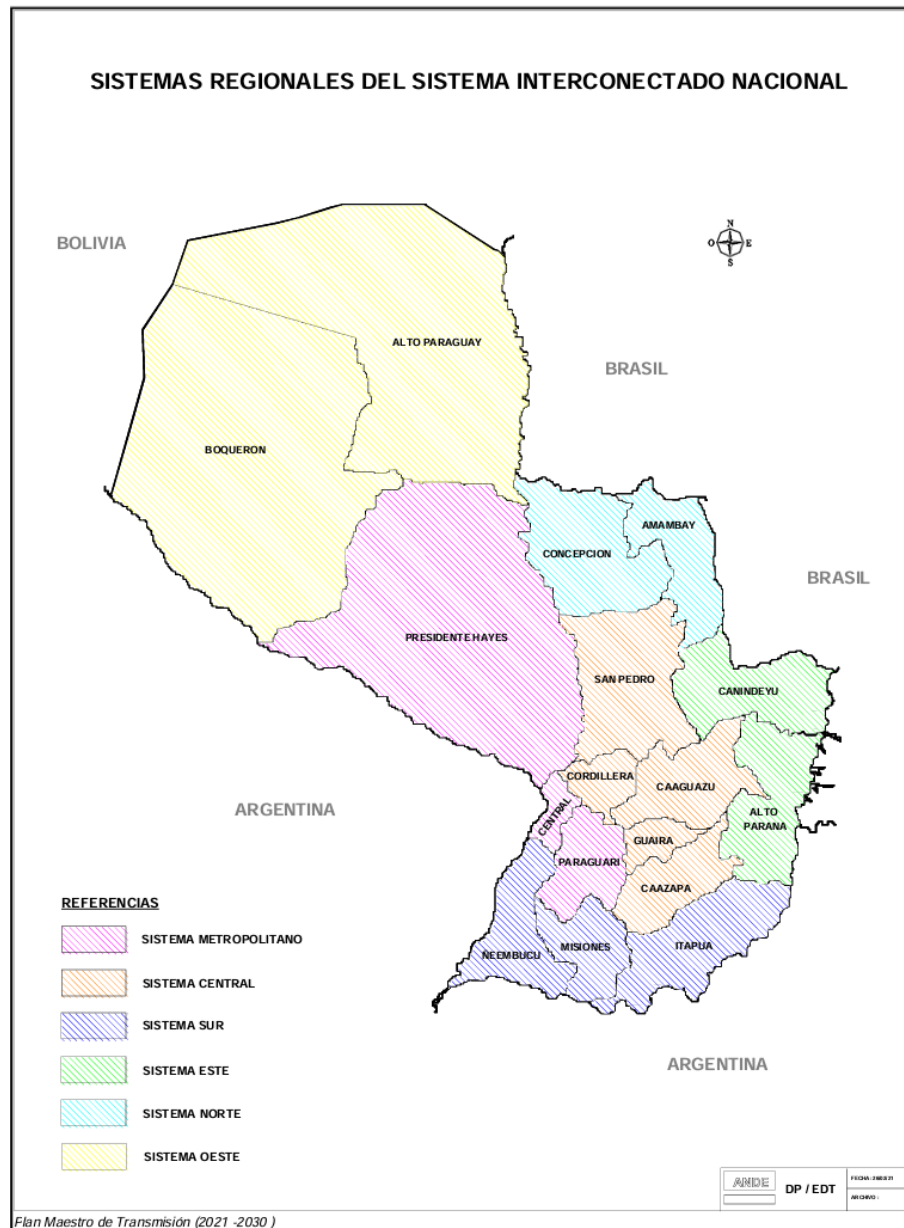


Figura 2.2: Sistemas regionales del SIN.

Fuente: [1]

2.2. Demanda eléctrica nacional

La demanda de un sistema eléctrico de potencia (SEP) es la carga solicitada a la fuente de abastecimiento del sistema en los puntos terminales durante un período de tiempo específico, expresada en términos de potencia activa. Además de la carga requerida debemos agregar la variable pérdida, que incluye las

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

causadas en la transmisión y en la distribución a lo largo de toda la red eléctrica. Este factor dificulta la predicción exacta debido a que las pérdidas requieren un análisis de predicción distinto. Dentro de lo que respecta al sistema eléctrico nacional, las crecientes necesidades energéticas de la población en sus distintas actividades fueron incidiendo gradualmente en la demanda de potencia, lo cual se ve reflejado en la Figura 2.3, en adición a las altas temperaturas y los niveles elevados de humedad que manejamos en el país.

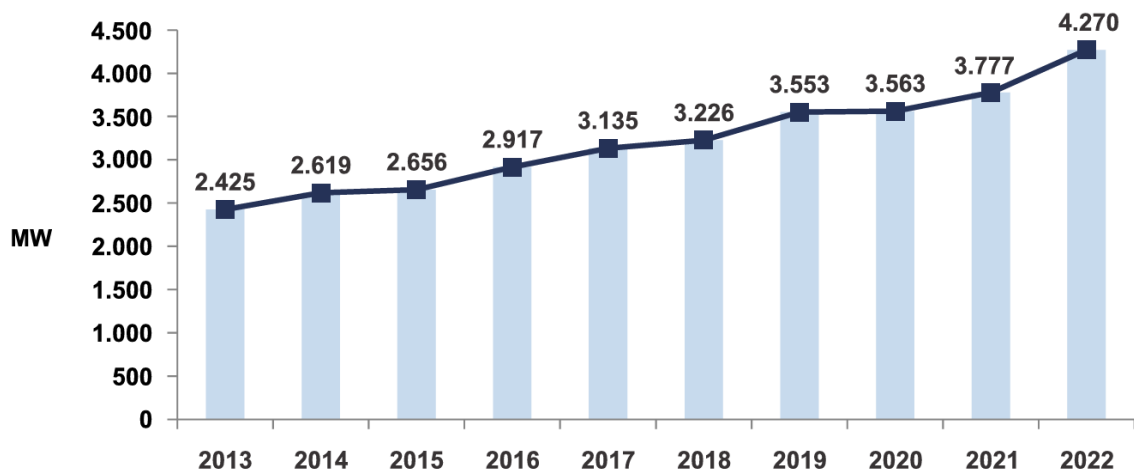


Figura 2.3: Histórico de demanda máxima.

Fuente: [9]

Podemos observar en la Figura 2.5 que existe una fuerte concentración de la demanda en el Sistema Metropolitano en nuestro sistema eléctrico paraguayo. Según los pronósticos realizados por la ANDE esta tendencia se mantendrá a largo plazo a menos que exista una fuerte incentivación industrial en localidades del interior del país y afectar de esa forma la distribución de la demanda. De hecho que las inversiones realizadas y proyectadas se concentran también en gran porcentaje en el Sistema Metropolitano lo cual se observa en la Figura 2.4. Toda la demanda total del sistema eléctrico nacional es cubierta por las tres centrales hidroeléctricas con las que se cuentan, como es posible apreciar en la tabla 2.1.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

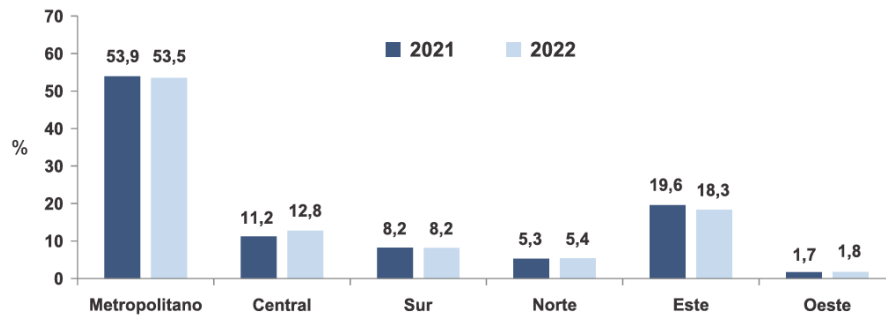


Figura 2.4: Distribución de demanda por sistemas 2021-2022.

Fuente: [9]

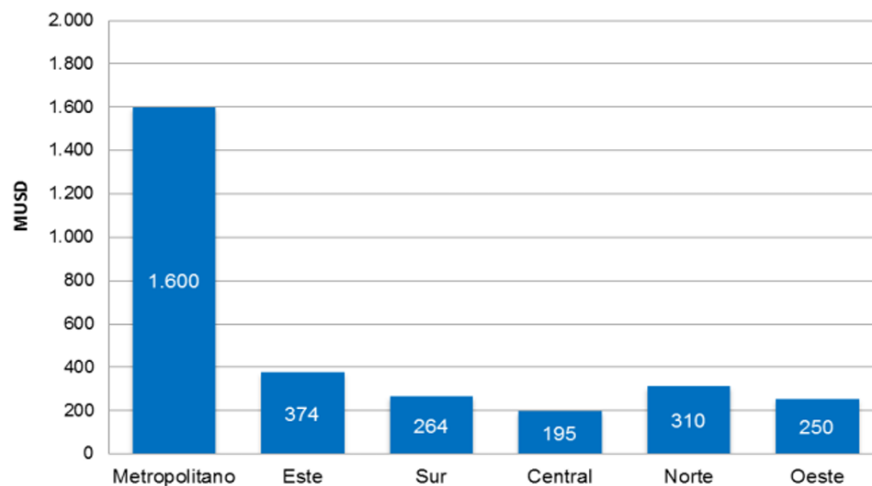


Figura 2.5: Inversión por sistemas 2021-2030.

Fuente: [1]

2.2.1. Demanda anual

Como se mencionó anteriormente, la proyección de la demanda anual a nivel nacional tiene una tendencia creciente debido a factores climáticos y de necesidades energéticas. Se presentan diferentes escenarios referenciales al crecimiento de la demanda, no obstante, con la actualización del Plan Maestro de Transmisión 2021-2030 se adopta el escenario Medio de Crecimiento de la demanda, el cual corresponde a un escenario con moderado crecimiento industrial. Esto resulta en tasas promedio de crecimiento del 6,6 % para la demanda máxi-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 2.1: Producción y consumo de energía en Paraguay, 2021-2022

| | 2021 | | 2022 | | Variación |
|----------------------------|-------------------|--------------|-------------------|--------------|--------------|
| | MWh | % | MWh | % | % |
| Origen | | | | | |
| Energía generada bruta | 624.330 | 3,3 | 782.671 | 4,0 | 25,4 |
| Energía comprada | 18.112.894 | 96,7 | 18.962.906 | 96,0 | 4,7 |
| Energía total bruta | 18.737.224 | 100,0 | 19.745.577 | 100,0 | 5,4 |
| Destino | | | | | |
| Consumo nacional | 18.585.269 | 99,2 | 19.635.952 | 99,4 | 5,7 |
| Exportación | 151.955 | 0,8 | 109.625 | 0,6 | -27,9 |

Fuente: [9]

ma simultánea y del 6,8 % para la energía requerida del Sistema Interconectado Nacional (SIN) [10], lo cual se ve reflejado en la Figura 2.6.

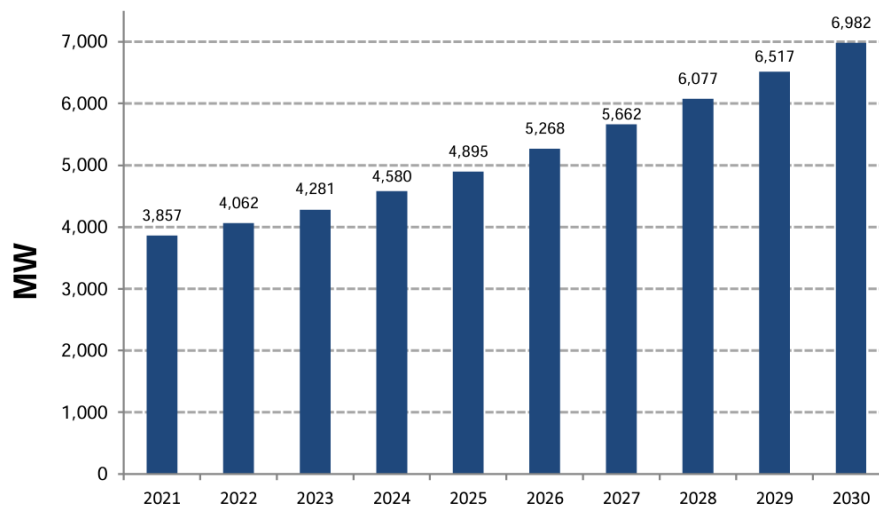


Figura 2.6: Proyección de Demanda Máxima del SIN utilizada.

Fuente: [1].

2.2.2. Demanda mensual

La demanda mensual a nivel nacional corresponde a un comportamiento típico de un país cálido del hemisferio Sur. Durante los meses de verano (diciembre, enero y febrero) la demanda de energía eléctrica es más alta en comparación

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

al resto del año. Esto puede atribuirse al uso intensivo de sistemas de enfriamiento y aire acondicionado. Asimismo, hay una presencia notable de valores atípicos, que sugieren picos de demanda aún más altos en ciertos días durante estos meses, pudiendo coincidir con los días en los cuales nos aproximamos, o incluso rompemos, al récord de temperatura alta registrada. Después del verano, la demanda disminuye en los meses de otoño e invierno, pero sigue siendo considerable en comparación con los valores más bajos al principio del año. La demanda vuelve a incrementarse levemente hacia el final del año, posiblemente debido al inicio del período cálido o a la actividad festiva del fin de año, que puede aumentar el uso de iluminación y aparatos eléctricos.

Este patrón muestra la importancia de considerar las estaciones y el clima al analizar y planificar la generación y el suministro de energía eléctrica en Paraguay.

2.2.3. Demanda diaria

El patrón de consumo diaria en una semana típica, es decir, sin considerar asuetos o feriados y tampoco fenómenos meteorológicos mayores, el perfil de consumo es mayor de lunes a viernes, probablemente debido a la actividad cotidiana laboral y escolar de las personas en estos días. Iniciando desde las últimas horas del viernes, durante el fin de semana la demanda tiende a bajar en consideración y se presentan los valores más bajos del intervalo semanal.

2.2.4. Demanda horaria

El consumo de energía eléctrica en un perfil horario que se puede observar en la Figura 2.7 presenta marcadas variaciones a lo largo de un día común. Teniendo en cuenta siempre los horarios donde se acentúan las elevadas temperaturas así como el inicio de las actividades cotidianas nocturnas son en estos tramos

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

donde se presentan las demandas más elevadas.

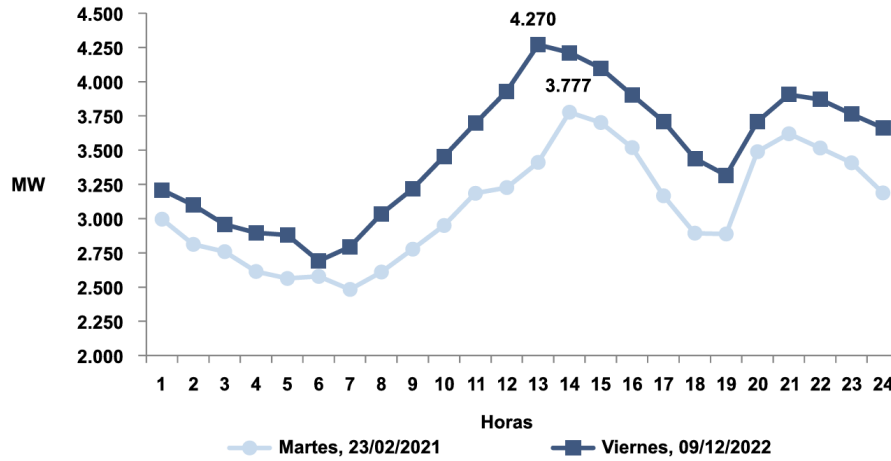


Figura 2.7: Curvas de Carga del Día de la Demanda Máxima.

Fuente: [9]

2.2.5. Sistema Interconectado Nacional

La principal variable de estudio para este Trabajo Final de Grado es el Sistema Interconectado Nacional (SIN), que es básicamente todo el sistema eléctrico encargado de abastecer la demanda de energía eléctrica a nivel nacional de forma interna. Si tomamos en cuenta las interconexiones destinadas a exportaciones ya estaríamos hablando del Sistema Interconectado (SI) en su totalidad. La diferencia porcentual anula entre consumo y exportación la podemos observar en la Figura 2.8

Esta variable es medida de forma cuantitativa en MW generalmente, en un determinado tiempo t :

$$SIN(t) = SI(t) - Ventas \quad (2.1)$$

$$SI(t) = PIPU(t) + PYAC(t) + PACY(t) + Compras \quad (2.2)$$

En términos de generación, transmisión y distribución estas dos variables de

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

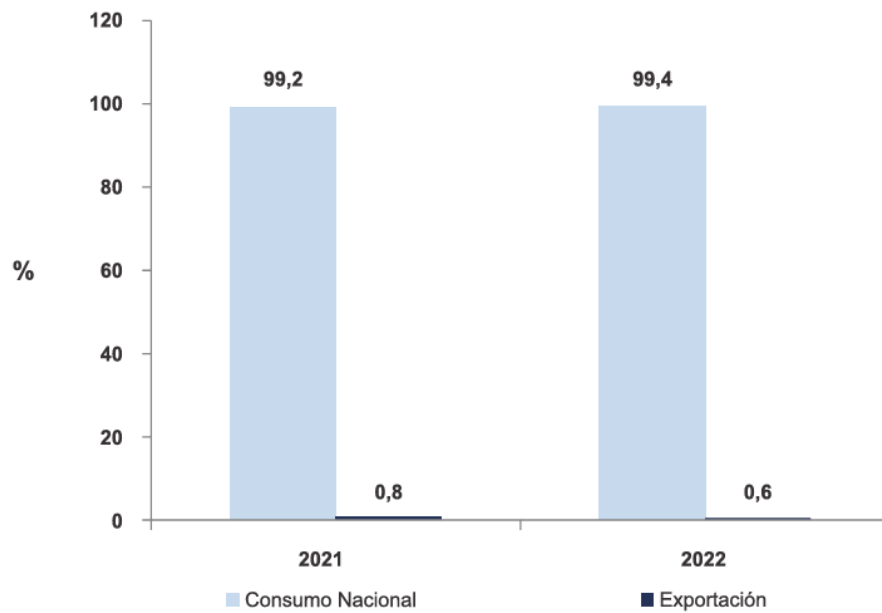


Figura 2.8: Destino de la energía eléctrica.

Fuente: [9]

medición tienen un impacto muy grande en la optimización y calidad de los sistemas eléctricos involucrados, además ser utilizada como uno de los principales parámetros a la hora de estudiar acuerdos entre las instituciones binacionales. Existen dos subsistemas como se puede observar en la Figura 2.9 en los cuales se encuentra dividido el Sistema Interconectado Nacional (SIN):

- **Sistema Eléctrico Brasileño (SEB):** alimentado por las hidroeléctricas de Itaipú y Acaray.
- **Sistema Argentino de Interconexión (SADI):** alimentado por la hidroeléctrica de Yacyretá.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL



Figura 2.9: Mapa eléctrico del Paraguay (2022).

Fuente: [9]

2.2.6. Parque de Generación Existente

Las fuentes de generación existentes en el SIN son 100 % de origen hidroeléctrico, lo cual se puede observar en la Figura 2.10, siendo las características de las mismas, las siguientes:

- **Central Hidroeléctrica de Acaray (CH-Acaray)** Las cuatro unidades de la CH-Acaray operan inicialmente con potencias nominales de 56 MVA (Grupos 1 y 2) y 75 MVA (Grupos 3 y 4).
- **Central Hidroeléctrica de Itaipú (CH-Itaipú)** La CH-Itaipú despacha hasta diez unidades generadoras de 50 Hz de 823,6 MVA cada una para atender

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

la demanda.

- **Central Hidroeléctrica de Yacyretá (CH-EBY)** Dada la elevación de cota en la zona de influencia, alcanzándose el valor nominal de 83 msnm, se considera el despacho de las veinte unidades de la CH-EBY con una capacidad de 172,5 MVA, y operando interconectada al Sistema Argentino de Interconexión (SADI). Al mismo tiempo se tiene en consideración la ampliación de la central en dos proyectos:
 - **Motorización del Brazo Aña Cua:** Construcción en el Brazo Aña Cuá de una Casa de Máquinas con 3 unidades generadoras con una capacidad de 100 MVA cada una.
 - **Ampliación de la Casa de Máquinas de la CH-EBY:** Ampliación de la Casa de Máquinas con 3 unidades generadoras con una capacidad de 172,5 MVA cada una.

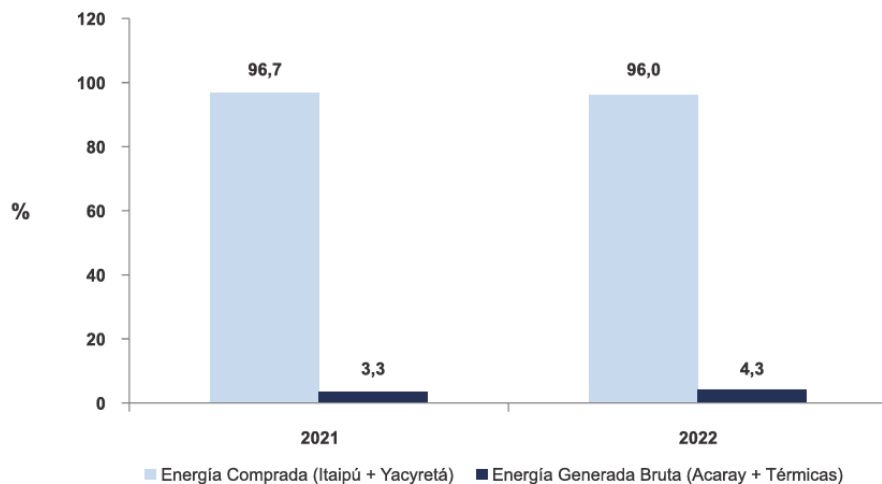


Figura 2.10: Origen de la energía eléctrica.

Fuente: [9]

El resumen del despacho de potencia de las Centrales Hidroeléctricas Existentes del SIN requerido por el Plan Maestro de Transmisión 2021-2030. como

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

se observa en la Figura 2.11 para cubrir la demanda de carga punta del periodo analizado se presenta en la Figura 2.6. El despacho durante la demanda máxima del año 2030 de las centrales sería el siguiente:

- CH-Acaray. Despacho de 256 MW (100 %).
- CH-Itaipú. Despacho de 4.385 MW (72 %).
- CH-EBY. Despacho de 1.530 MW (91 %).

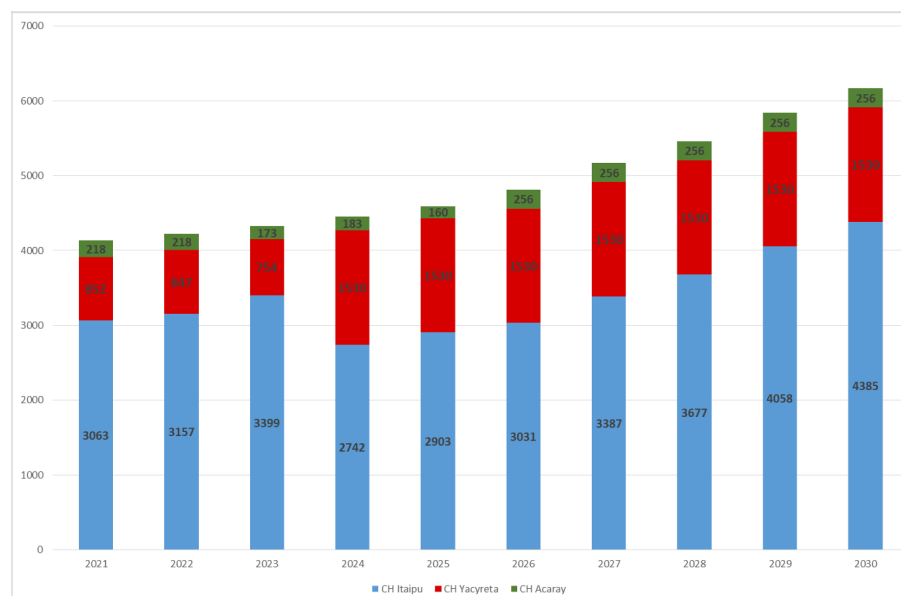


Figura 2.11: Despacho proyectado por el Plan Maestro de Transmisión (PMT) 2021-2030.

Fuente: [1]

2.3. Predicción de la demanda eléctrica

La predicción de la demanda eléctrica emerge como una cuestión crítica en el ámbito energético, siendo su precisión esencial para garantizar la confiabilidad y eficiencia del suministro eléctrico. La anticipación acertada de los patrones de consumo no solo asegura un suministro fiable, sino que también impacta en aspectos clave como la eficiencia energética, la sostenibilidad y la toma de decisiones gubernamentales [11].

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

En los últimos años, la investigación en predicción de la demanda eléctrica ha avanzado significativamente. Las redes neuronales, en especial, han demostrado ser herramientas efectivas para la predicción de series temporales, incluyendo la demanda eléctrica [12]. Este trabajo se propone evaluar la eficacia de la arquitectura TFT (Temporal Fusion Transformer) en la predicción de la demanda eléctrica a nivel nacional. Aunque la arquitectura TFT ha demostrado su eficacia en series temporales, su aplicación específica en la predicción de la demanda eléctrica requiere una investigación más profunda.

La importancia de esta evaluación radica en la posibilidad de mejorar las predicciones en comparación con otros modelos existentes, así como en comprender la relevancia de las variables que influyen en dicha predicción. Para lograr esto, es esencial realizar una revisión exhaustiva del estado del arte en relación con la arquitectura TFT, incluyendo menciones pertinentes a otras arquitecturas que se utilizarán para una comparación detallada de los resultados.

2.4. Modelos Estadísticos vs. Modelos de Inteligencia Artificial

Históricamente, los modelos estadísticos basados en series temporales, como ARIMA (Autorregresivo Integrado de Medias Móviles) [11], han dominado la predicción de la demanda eléctrica. Su eficiencia en la reproducción de diversas series estacionarias es notable. Sin embargo, con el advenimiento de modelos basados en inteligencia artificial, la perspectiva ha evolucionado [12].

Los modelos de inteligencia artificial, desde regresión lineal hasta complejos modelos de aprendizaje profundo, representan sistemas inteligentes con curvas de aprendizaje más rápidas. En la predicción de series temporales, especialmente en el consumo eléctrico, algoritmos de aprendizaje de máquina como Árboles de Decisión, Máquinas de Soporte Vectorial y Redes Neuronales Artificiales han demostrado ser estrategias efectivas [12]. Cada enfoque presenta ventajas en

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

métricas de error, porcentaje de aciertos y consideraciones computacionales, siendo crucial una evaluación exhaustiva para determinar la estrategia más adecuada.

Es importante destacar que la arquitectura TFT se encuentra dentro del conjunto de modelos fundamentados en Inteligencia Artificial, en particular, aquellos que se basan en el mecanismo de atención. No obstante, es relevante señalar que, hasta la fecha presente, continúan existiendo investigaciones relacionadas con diversas redes destinadas a la predicción de series temporales, las cuales se sustentan en enfoques que abarcan desde las convencionales redes neuronales hasta las recurrentes.

2.5. Modelos basados en Inteligencia Artificial

Los modelos de predicción de series temporales desempeñan un papel fundamental en la anticipación de valores futuros para una variable objetivo $y(i,t)$ correspondiente a una entidad dada i en el tiempo t . Cada entidad representa una agrupación lógica de información temporal, como las mediciones de estaciones meteorológicas individuales en climatología o signos vitales de diferentes pacientes en medicina o consumo eléctrico. La ecuación dada se presenta de la siguiente manera:

$$y'_{i,t+1} = f(y_{i,t-k:t}, x_{i,t-k:t}, s_i), \quad (2.3)$$

La ecuación 2.3 modela la predicción ($y'_{i,t+1}$) para el siguiente paso temporal de la variable objetivo asociada a la entidad i . La variable $y_{i,t-k:t}$ abarca las observaciones pasadas de la variable objetivo y para la entidad i a lo largo de una ventana de tiempo k , mientras que $x_{i,t-k:t}$ representa las entradas correspondientes durante el mismo período, es decir, las variables exógenas asociadas a la entidad i . Además, s_i denota los metadatos estáticos asociados a la entidad. La función $f()$, encapsulada en la ecuación, representa el mecanismo de predicción apren-

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

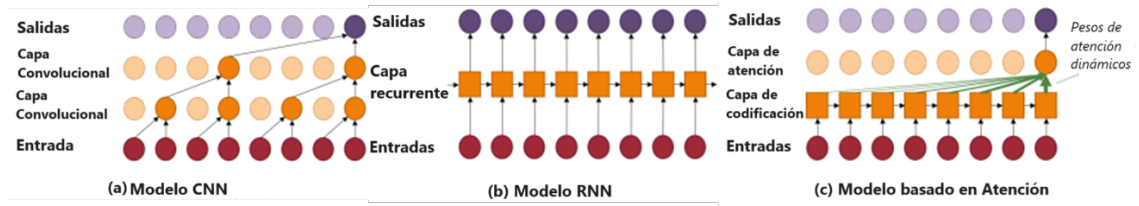


Figura 2.12: Arquitecturas de codificadores utilizadas en series temporales.

Fuente: [12]

dido por el modelo, utilizando las observaciones pasadas, las entradas actuales y los metadatos estáticos para anticipar el valor futuro de la variable objetivo para la entidad i en el tiempo $t + 1$.

Los modelos de predicción de series temporales se pueden utilizar en una variedad de aplicaciones, como:

- Pronóstico del clima
- Monitoreo de salud
- Gestión de inventario
- Control de procesos

En los últimos años, los modelos de aprendizaje automático, como las redes neuronales, se han convertido en una herramienta cada vez más popular para la predicción de series temporales. Estos modelos pueden aprender relaciones complejas entre las observaciones pasadas y las variables exógenas, lo que puede mejorar el rendimiento de la predicción [12]. Las Redes neuronales aprenden relaciones entre los datos usando capas intermedias no lineales como se presenta en la Figura 2.12. Esto se puede observar en una configuración de codificador-decodificador descrito por el siguiente conjunto de fórmulas:

$$f(y_{t-k:t}, x_{t-k:t}, s) = g_{\text{dec}}(z_t), \quad (2.4)$$

$$z_t = g_{\text{enc}}(y_{t-k:t}, x_{t-k:t}, s), \quad (2.5)$$

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

donde las funciones g_{enc} y g_{dec} de codificación y decodificación, respectivamente. Es importante destacar que el subíndice i de la ecuación (2.3) se ha excluido para simplificar la notación, por ejemplo, sustituyendo $y_{i,t}$ por y_t . Estos elementos, a saber, los codificadores y decodificadores, constituyen los elementos fundamentales en la construcción de arquitecturas de aprendizaje profundo. La elección específica de la red determina los tipos de relaciones que nuestro modelo puede aprender.

2.5.1. Redes Convolucionales

Las Redes Neuronales Convolucionales (CNN), concebidas inicialmente para conjuntos de datos de imágenes, se especializan en la extracción de relaciones locales invariables a través de dimensiones espaciales. En el contexto de las series temporales, los investigadores han modificado las CNN mediante la incorporación de múltiples capas de convoluciones causales lo que implica el uso de filtros convoluciones diseñados para garantizar la utilización exclusiva de información pasada en el proceso de predicción. Cada capa intermedia sigue las siguientes ecuaciones:

$$h_{l+1,t} = A((W \star h)_{l,t}), \quad (2.6)$$

$$(W \star h)_{l,t} = \sum_{\tau=0}^k W(l, \tau) h_{l,t-\tau}, \quad (2.7)$$

donde $h_{l,t} \in \mathbf{R}^{H_{\text{in}}}$ es un estado intermedio en la capa l en el tiempo t , \star es la operación de convolución, $W(l, \tau) \in \mathbf{R}^{H_{\text{in}} \times H_{\text{out}}}$, τ es el índice de retraso, $A()$ es una función de activación, como una función sigmoide, $W(l, t) \in \mathbf{R}^{H_{\text{out}}}$, $t \in \mathbf{R}^{H_{\text{out}}}$ es la salida convolucionada en la capa l en el tiempo t , k es el tamaño del campo receptivo, τ es el índice del retraso. Esta fórmula calcula la salida convolucionada en la capa actual, $(W \star h)_{l,t}$, en función del estado intermedio en la capa actual, $h_{l,t}$, y los pesos del filtro, $W(l, \tau)$, para los retrasos de 0 a $k-1$. Las CNN causales

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

son útiles para el pronóstico de series temporales porque pueden aprender relaciones temporales entre las variables de entrada. Esto se debe a que las CNN causales sólo utilizan información pasada para hacer predicciones.

2.5.2. Redes Neuronales Recurrentes

Han sido fuertemente utilizadas para aplicaciones de tarea del procesamiento natural del lenguaje debido a su capacidad para aprender secuencias, se caracterizan por la conexiones de salida que vuelven a ser usadas como entradas de la siguiente secuencia de una misma capa o de más. En su esencia, las celdas RNR contienen un estado de memoria interno que actúa como un resumen compacto de la información pasada. El estado de memoria se actualiza recursivamente con nuevas observaciones en cada paso de tiempo, como se muestra en la Figura 2.12 .b en la siguiente fórmula se describe el estado interno

$$z_t = v(z_{t-1}, y_t, x_t, s), \quad (2.8)$$

donde $z_t \in \mathbf{R}^h$ representa el estado interno oculto en el tiempo t , $y_t \in \mathbf{R}^m$ es la entrada en el tiempo t , v es la función de actualización de memoria aprendida.

2.5.3. Memoria a Corto y Largo Plazo (LSTM)

Debido a la ventana de retroceso infinita, las variantes más antiguas de las RNN pueden tener limitaciones en el aprendizaje de dependencias a largo plazo en los datos [13, 14], debido a problemas con gradientes que explotan y desaparecen [15]. Intuitivamente, esto puede verse como una forma de resonancia en el estado de memoria. Por este motivo, las Redes de Memoria a Corto y Largo Plazo (LSTM) [16] fueron desarrolladas para abordar estas limitaciones, mejorando el flujo de gradientes dentro de la red.

Esto se logra mediante el uso de un estado de celda C_t que almacena infor-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

mación a largo plazo, modulado a través de una serie de compuertas como se indica a continuación:

$$\text{Compuerta de Entrada: } i_t = \sigma(W_{i1}z_{t-1} + W_{i2}y_t + W_{i3}x_t + W_{i4}s + b_i), \quad (2.9)$$

$$\text{Compuerta de Salida: } o_t = \sigma(W_{o1}z_{t-1} + W_{o2}y_t + W_{o3}x_t + W_{o4}s + b_o), \quad (2.10)$$

$$\text{Compuerta de Olvido: } f_t = \sigma(W_{f1}z_{t-1} + W_{f2}y_t + W_{f3}x_t + W_{f4}s + b_f), \quad (2.11)$$

donde z_{t-1} es el estado oculto de la LSTM y $\sigma()$ es la función de activación sigmoide. Las compuertas modifican los estados ocultos y de la celda de la LSTM de la siguiente manera:

$$\text{Estado Oculto: } z_t = o_t \odot \tanh(c_t), \quad (2.12)$$

$$\text{Estado de Celda: } c_t = f_t \odot c_{t-1} + (W_{c1}z_{t-1} + W_{c2}y_t + W_{c3}x_t + W_{c4}s + b_c), \quad (2.13)$$

donde \odot es el producto Hadamard.

2.5.4. Modelos basados en Atención

El concepto de atención es fundamental en el contexto de las arquitecturas basadas en Transformers. En su forma más genérica, la atención puede describirse como la capacidad de enfocarse en partes específicas de un conjunto de datos. En el contexto de las redes neuronales, la atención permite asignar diferentes niveles de importancia a elementos individuales del input, lo que lleva a predicciones más precisas y contextuales. La atención se basa en el principio de asignar pesos a elementos de entrada, de manera que aquellos con un mayor peso reciben más atención. Esto se asemeja al funcionamiento de la atención

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

en el cerebro humano, donde diferentes partes del estímulo reciben un enfoque diferenciado. Es una función que mapea consultas (Q) y un conjunto de llaves-valores (K-V) con un resultado donde las consultas, llaves y valores son vectores y el resultado es la suma escalada de todos los valores, y los pesos asignados a cada valores es calculado por proyecciones, que es una función de compatibilidad, a las consultas que corresponden a una llave [4].

El mecanismo de atención funciona de la siguiente manera:

1. El modelo crea una representación de los datos de entrada.
2. El modelo crea una representación de la salida que se está procesando.
3. El modelo calcula una puntuación de atención para cada token de los datos de entrada.
4. El modelo normaliza las puntuaciones de atención.
5. El modelo suma los tokens de los datos de entrada ponderados por las puntuaciones de atención.

El resultado es un vector de contexto que representa la información relevante de los datos de entrada. El modelo utiliza el vector de contexto para generar la salida.

Atención por Producto de Punto Escalado

La atención por producto escalado es una función de atención simple y eficaz que se utiliza en una variedad de tareas de aprendizaje automático. La función de compatibilidad mide la similitud entre la consulta y cada clave. Las puntuaciones de atención se normalizan mediante la función `softmax()` para que sumen 1. La suma ponderada de los valores es la salida de la atención [4].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \quad (2.14)$$

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

donde Q (*Query*) como un vector que representa la consulta, K (*Key*) como un conjunto de vectores que representan las claves, V (*Value*) como un conjunto de vectores que representan los valores, y d_k como la dimensión de los vectores clave y valor. Además, se utiliza la función `softmax()` como una función de activación que normaliza las puntuaciones de atención para que sumen 1, una esquematización del mismo se observa en la Figura 2.13.

Scaled Dot-Product Attention

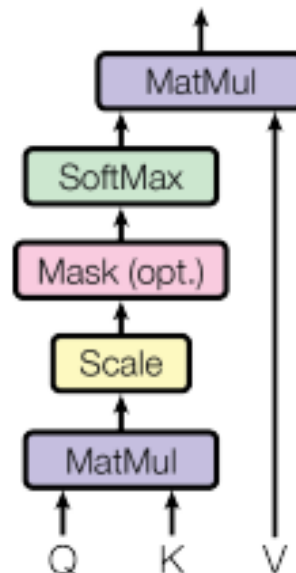


Figura 2.13: Atención escalada del producto escalar.

Fuente: [4]

Atención Multicabezal

La atención multicabezal (*multi-head attention*) es un mecanismo de atención que permite a los modelos de aprendizaje automático aprender diferentes representaciones de la entrada. Esto es beneficioso para tareas complejas como la traducción automática y el reconocimiento de voz [4].

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

La atención multicabezal funciona de la siguiente manera:

1. La entrada se proyecta en varias representaciones diferentes, llamadas cabezas de atención.
2. Cada cabeza de atención calcula una atención sobre la entrada, utilizando una función de atención diferente.
3. Las salidas de las diferentes cabezas de atención se concatenan y proyectan en una sola representación.

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O, \quad (2.15)$$

donde $head_i = Attention(W_i^Q, W_i^K, W_i^V)$ y W contiene las proyecciones en forma de matriz.

Fundamentos de la arquitectura Transformer

La arquitectura Transformer es una arquitectura de redes neuronales que se basa en un mecanismo de atención para aprender relaciones entre los datos, principalmente utilizada para tareas relacionadas al procesamiento natural del lenguaje. La arquitectura Transformer se compone de dos componentes principales como se puede observar en la Figura 2.14:

- **Encoder:** El encoder es responsable de codificar la entrada en una representación que pueda ser utilizada por el decoder.
- **Decoder:** El decoder es responsable de generar la salida a partir de la representación codificada.

El encoder y el decoder están formados por una serie de capas de atención, que permiten a la red aprender a centrarse en las partes más relevantes de la entrada para realizar una tarea determinada. Sobre el resto de los componentes:

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

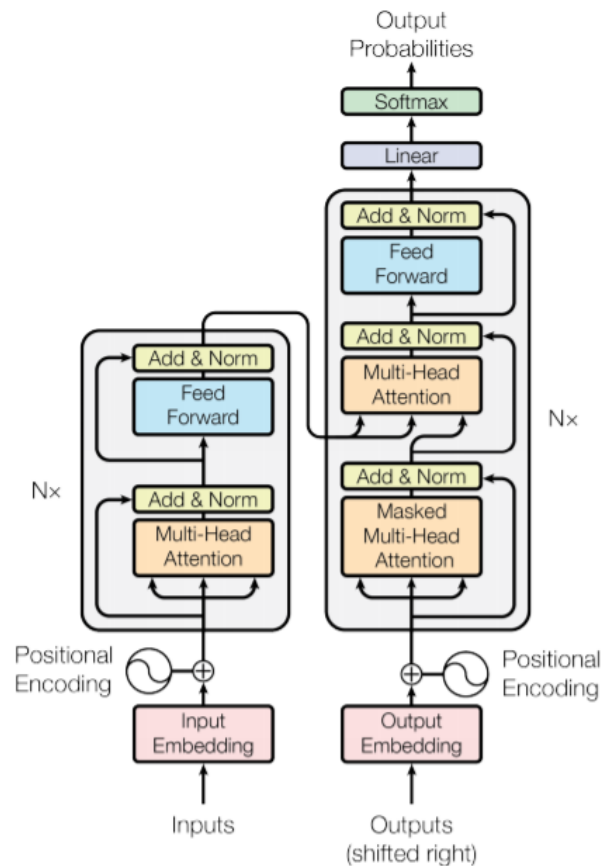


Figura 2.14: Arquitectura Transformer.

Fuente: [4]

- **Autoatención multicabezal:** La autoatención multicabezal es un mecanismo de atención que permite a una red neuronal aprender relaciones a largo plazo entre los tokens de una secuencia.
- **Conexión residual:** Una conexión residual es una conexión que permite a una red neuronal aprender funciones más complejas saltando capas anteriores de la red.
- **Normalización de capas:** La normalización de capas es una técnica que ayuda a estabilizar el entrenamiento de redes neuronales profundas.
- **Enmascaramiento:** El enmascaramiento es una técnica que se utiliza para evitar que un modelo de aprendizaje automático aprenda relaciones rele-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

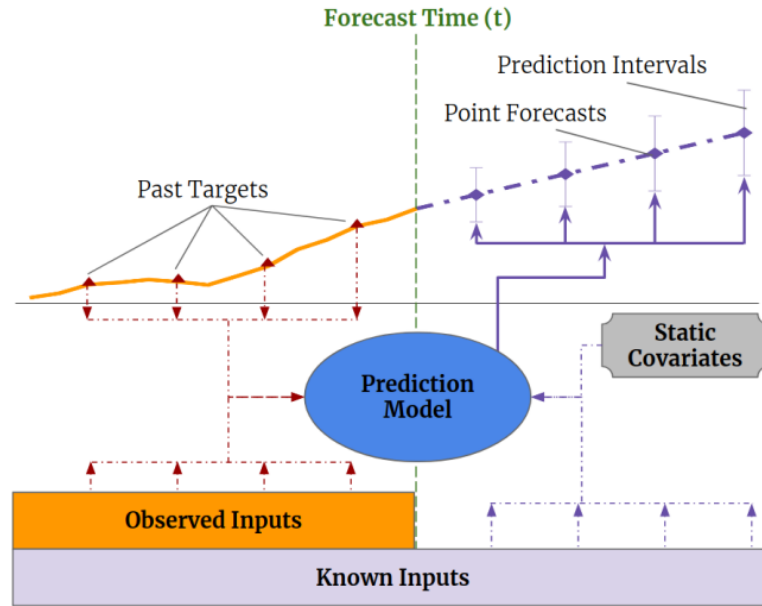


Figura 2.15: Ilustración de una predicción a múltiples horizontes.

Fuente: [5]

vantes entre los datos.

2.6. Modelos De Pronóstico de Múltiples Horizontes

Es beneficioso tener acceso a estimaciones predictivas en múltiples puntos futuros, permitiendo visualizar tendencias a lo largo de un horizonte futuro y optimizar acciones esto es observable en la figura 2.15. Desde un punto de vista estadístico, el pronóstico de múltiples horizontes se considera una modificación del problema de predicción de un paso adelante, representado por la ecuación:

$$y'_{t+\tau} = f(y_{t-k:\tau}, x_{t-k:\tau}, u_{t-k:\tau}, s, \tau), \quad (2.16)$$

donde $y'_{t+\tau}$ es el valor predicho de la variable objetivo en el tiempo $t + \tau$, τ es el horizonte de pronóstico discreto, u_t son entradas futuras conocidas, x_t son entradas que solo se pueden observar históricamente, s representa covariantes adicionales o información relevante, y f es la función de predicción.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

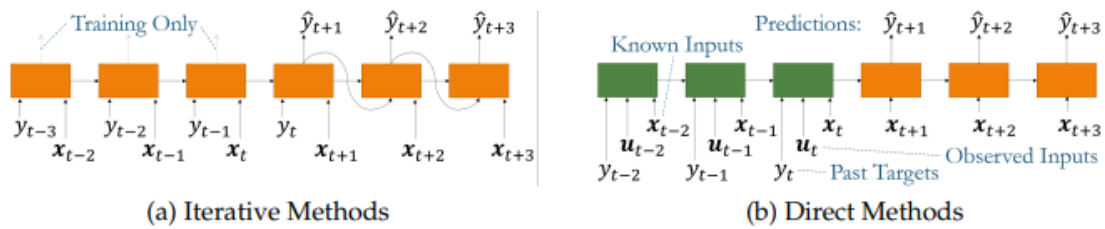


Figura 2.16: Métodos iterativos vs Directos

Fuente: [12]

Las arquitecturas de aprendizaje profundo para el pronóstico de múltiples horizontes se pueden dividir en métodos iterativos y directos [12].

Método Iterativos

Los métodos iterativos, también conocidos como métodos secuenciales o autorregresivos, son una arquitectura de aprendizaje profundo comúnmente utilizada para el pronóstico de múltiples horizontes. Esto significa que el modelo se utiliza para predecir el valor de la variable objetivo en el tiempo $t + 1$, y luego utiliza esta predicción como entrada para predecir el valor en el tiempo $t + 2$, y así sucesivamente. Estos métodos generan predicciones paso a paso, prediciendo un paso adelante a la vez. El enfoque recursivo se ilustra en la Figura 2.16.

Para generar pronósticos de múltiples horizontes con un método iterativo, el modelo se entrena en un conjunto de datos históricos para aprender las relaciones entre la variable objetivo y las características de entrada. Una vez entrenado, el modelo se utiliza para predecir los pasos temporales futuros. Para realizar una predicción para el tiempo $t + 1$, el modelo se alimenta con los valores reales de las características de entrada en el tiempo t y del valor previsto de la variable objetivo en el tiempo t . Luego, el modelo genera una predicción para la variable objetivo en el tiempo $t + 1$. Este proceso se repite recursivamente para generar predicciones para todos los horizontes futuros de interés.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

Sin embargo, los métodos iterativos tienen limitaciones, siendo costosos computacionalmente para horizontes largos, acumulando errores con el tiempo y asumiendo que todas las entradas, excepto la objetivo, son conocidas en tiempo de ejecución [12].

A pesar de estas limitaciones, los métodos iterativos son populares para el pronóstico de múltiples horizontes debido a su simplicidad, facilidad de implementación y capacidad para manejar datos altamente autorregresivos.

Métodos Directos

Los métodos directos para el pronóstico de múltiples horizontes generan predicciones para todos los horizontes de pronóstico deseados simultáneamente. Para ello, utilizan una arquitectura de aprendizaje profundo que puede procesar toda la información disponible, tanto la información pasada como la información futura [12].

Una arquitectura típica de método directo para el pronóstico de múltiples horizontes es la siguiente:

$$\text{Encoder: } y(t-k:t), x(t-k:t), u(t-k:t) \xrightarrow{h_{\text{enc}}} h$$

$$\text{Decoder: } h, u(t+1:\tau) \xrightarrow{h_{\text{dec}}} y(t+1:\tau),$$

donde: $y(t-k:t)$ es la serie temporal de la variable objetivo hasta el tiempo t , $x(t-k:t)$ es la serie temporal de las entradas a la variable objetivo hasta el tiempo t , $u(t-k:t)$ es la serie temporal de las entradas futuras conocidas hasta el horizonte de pronóstico τ , h es la representación latente de la información pasada y presente, y $y(t+1:\tau)$ son las predicciones de la variable objetivo para los horizontes $t+1, t+2, \dots, \tau$.

El codificador, representado por la función h_{enc} , es responsable de generar una representación latente de la información pasada y presente. Esta represen-

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

tación se utiliza luego por el decodificador, representado por la función h_{dec} , para generar las predicciones de la variable objetivo para los horizontes de pronóstico deseados.

Modelos Híbridos

Los modelos híbridos combinan las ventajas de ambos enfoques. Incorporan el conocimiento del dominio de los modelos estadísticos y utilizan la flexibilidad de los modelos de aprendizaje automático para capturar dependencias no lineales en los datos [12].

Estos modelos se clasifican en dos categorías principales:

1. **Modelos híbridos estadísticos:** Combina un modelo estadístico con un modelo de aprendizaje automático para mejorar el rendimiento de la predicción. Un ejemplo es el Exponential Smoothing RNN (ES-RNN), que combina suavizado exponencial con una red neuronal recurrente.

2. **Modelos híbridos probabilísticos:** Combina un modelo probabilístico con un modelo de aprendizaje automático para mejorar el rendimiento de la predicción. Un ejemplo es el DeepAR, que combina un modelo de regresión de movimiento aleatorio con una red neuronal recurrente.

Estos modelos híbridos han demostrado ser efectivos en diversas aplicaciones de predicción de series temporales. Especialmente, han mostrado superioridad sobre modelos puramente estadísticos o de aprendizaje automático en problemas con conjuntos de datos pequeños o con dependencias no lineales.

2.7. Fundamentos teóricos de la Arquitectura TFT

La arquitectura Transformers for Time Series Forecasting (TFT) es una extensión de las redes neuronales basadas en Transformers, que originalmente se desarrollaron para tareas de procesamiento de lenguaje natural [4]. Los Transfor-

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

mers son especialmente adecuados para modelar relaciones en secuencias de datos, lo que los hace prometedores en la predicción de series temporales. De la Figura 2.17 podemos destacar los siguiente elementos:

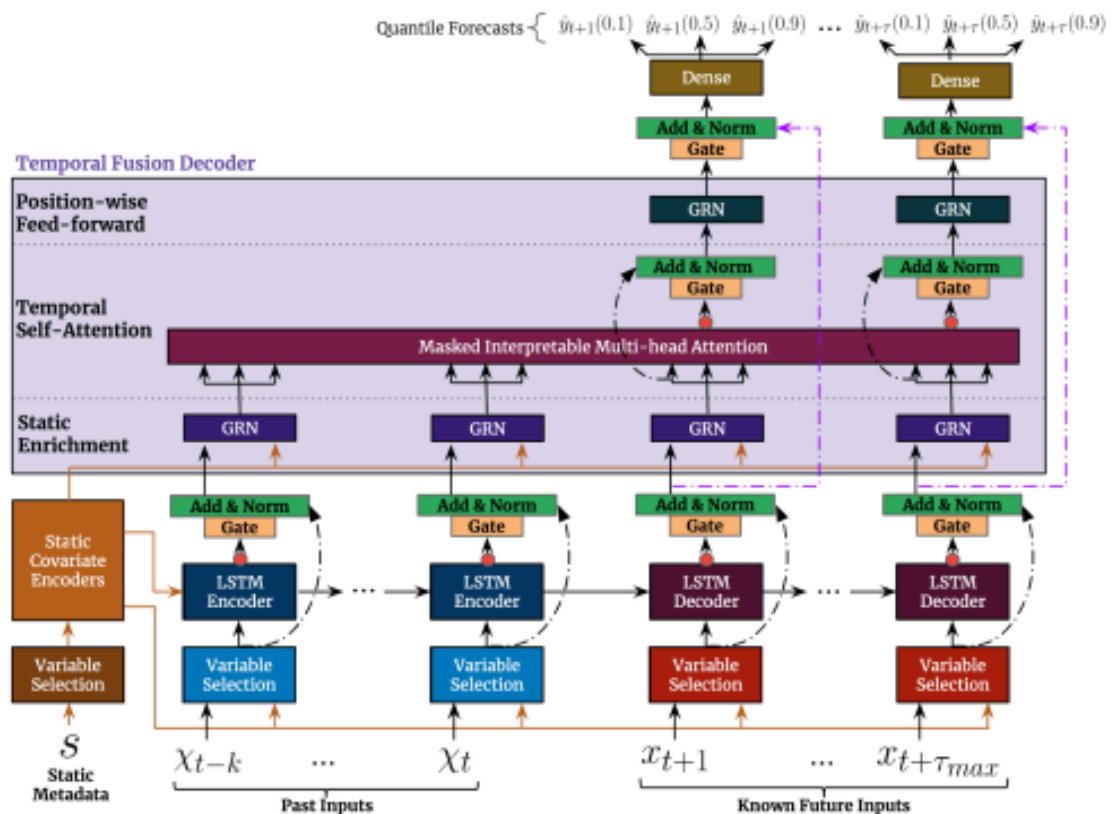


Figura 2.17: Arquitectura TFT.

Fuente: [5]

- **Mecanismos de Control:** La arquitectura TFT utiliza mecanismos de control (gating mechanisms) que permiten omitir componentes no utilizados de la arquitectura, proporcionando adaptabilidad en cuanto a la profundidad y complejidad de la red, haciéndola adecuada para una amplia gama de conjuntos de datos y escenarios.
- **Redes de Selección de Variables:** TFT incorpora redes de selección de variables que eligen las variables de entrada relevantes en cada paso tem-

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

poral, permitiendo al modelo determinar qué variables son importantes en cada momento, lo que es esencial para la adaptación a datos cambiantes.

- **Codificadores de Covariables Estáticas:** TFT integra codificadores que incorporan características estáticas en la red. Esto se logra mediante la codificación de vectores de contexto que condicionan la dinámica temporal. Las características estáticas, como la ubicación o el tipo de entidad, pueden influir en las tendencias a lo largo del tiempo.
- **Procesamiento Temporal:** TFT se encarga de aprender las relaciones temporales a largo y corto plazo a partir de las entradas variables en el tiempo (observadas y conocidas). Para el procesamiento a nivel local, se emplea una capa secuencia-a-secuencia. Además, TFT utiliza un innovador bloque de atención multi-cabezal interpretable para capturar las dependencias a largo plazo.
- **Intervalos de Predicción:** TFT proporciona intervalos de predicción que indican el rango de valores probables para el objetivo en cada horizonte de predicción. Esto es útil para comprender la incertidumbre en las predicciones y tomar decisiones basadas en riesgos.

2.7.1. Mecanismos de Control en TFT

Los mecanismos de control son una parte crucial de la arquitectura Temporal Fusion Transformer (TFT) y se utilizan para proporcionar flexibilidad al modelo en términos de cuánto procesamiento no lineal se aplica a los datos. Estos mecanismos determinan qué parte de la arquitectura de TFT es relevante y necesaria para realizar predicciones precisas en un conjunto de datos específico.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Gated Residual Network (GRN)

Los Mecanismos de Control se basan en un componente llamado "Gated Residual Network"(GRN) el cual se ve representado por la Figura 2.18 [5]. El GRN toma una entrada principal a y un vector de contexto opcional c . Su función es combinar y procesar estas entradas de manera que permita controlar la cantidad de procesamiento no lineal aplicado. La salida del GRN se calcula mediante la siguiente ecuación:

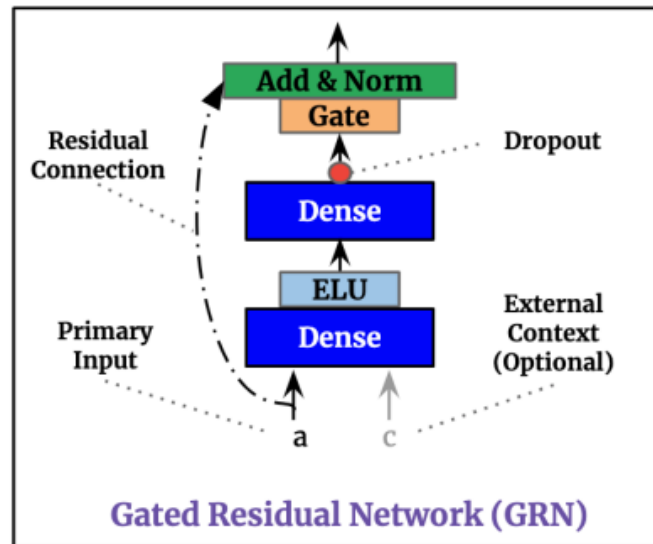


Figura 2.18: GRN

Fuente: [5]

$$\text{GRN}_{\omega}(a, c) = \text{LayerNorm}(a + \text{GLU}_{\omega}(\eta_1)) \quad (2.17)$$

$$\eta_1 = W_{1,\omega}\eta_2 + b_{1,\omega} \quad (2.18)$$

$$\eta_2 = \text{ELU}(W_{2,\omega}a + W_{3,\omega}c + b_{2,\omega}) \quad (2.19)$$

donde η_1 y η_2 son capas intermedias calculadas como parte del proceso, LayerNorm es una técnica de normalización de capa, y ω es un índice que denota el

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

uso compartido de pesos.

Gated Linear Units (GLU)

El componente principal que permite la flexibilidad de procesamiento es el uso de *Gated Linear Units* (GLUs). GLU es una unidad de procesamiento que utiliza una función sigmoide para controlar hasta qué punto se aplica una transformación lineal a los datos de entrada. Esto significa que TFT puede decidir omitir completamente la capa GRN si es necesario, ya que las salidas de GLU podrían ser cercanas a cero, lo que suprime la contribución no lineal.

$$GLU_{\omega}(\gamma) = \sigma(W_{4,\omega}\gamma + b_{4,\omega}) \odot (W_{5,\omega}\gamma + b_{5,\omega}), \quad (2.20)$$

donde: $\sigma(\cdot)$ es la función de activación sigmoide, $W_{(\cdot)}$ y $b_{(\cdot)}$ son pesos y sesgos, y \odot es el producto Hadamard que representa la multiplicación elemento por elemento.

Uso de Contexto Cero: En situaciones en las que no hay un vector de contexto disponible ($c = 0$), el GRN simplemente trata el contexto como cero ($c = 0$ en la ecuación) [5].

Dropout en Entrenamiento: Durante el proceso de entrenamiento, se aplica la técnica de *dropout* antes de la capa de control y la normalización de capa. Esto ayuda a prevenir el sobreajuste y mejora la generalización del modelo [5].

2.7.2. Redes de selección de variables

Se encarga de la selección de variables, es decir, cómo decide qué variables son relevantes para realizar predicciones precisas en series temporales. Relevancia de Variables Desconocidas: En muchos casos, cuando se trabaja con series temporales, hay múltiples variables disponibles, pero no se conoce de antemano cuál de ellas es relevante y cuál contribuye de manera específica a las

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

predicciones. TFT está diseñado para abordar este desafío al proporcionar la capacidad de seleccionar variables de manera individual para cada instancia de tiempo. Esto significa que en cada paso del tiempo, el modelo puede decidir cuáles son las variables más importantes para realizar predicciones. Además, esta selección de variables también permite a TFT eliminar las entradas ruidosas y no necesarias que podrían tener un impacto negativo en el rendimiento del modelo.

Utilización de Redes de Selección de Variables: TFT utiliza redes de selección de variables para realizar esta tarea de selección de variables. Estas redes se aplican tanto a las covariables estáticas como a las covariables dependientes del tiempo. La idea es obtener información sobre cuáles variables son más significativas para resolver un problema de predicción. Esto es particularmente útil en conjuntos de datos de series temporales del mundo real, donde algunas características pueden tener menos contenido predictivo, y seleccionar solo las más relevantes puede mejorar significativamente el rendimiento del modelo.

Representación de Variables Categóricas y Continuas: Para llevar a cabo la selección de variables, TFT utiliza la representación de variables categóricas a través de `.entity embeddings` transformaciones lineales para variables continuas. Estas representaciones se utilizan para convertir cada variable de entrada en un vector de dimensiones "dmodel" que se corresponde con las dimensiones de las capas posteriores del modelo, lo que permite conexiones "skip" (conexiones de salto) en el modelo.

Redes de Selección de Variables por Tipo: TFT utiliza diferentes redes de selección de variables para variables estáticas, pasadas y futuras. Estas redes están diseñadas específicamente para lidiar con cada tipo de entrada. Por ejemplo, en el caso de las entradas pasadas, se utiliza una red de selección de variables que toma en cuenta el contexto, mientras que para las entradas estáticas, el contexto no se considera necesario ya que estas entradas no cambian con el tiempo.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Procesamiento No Lineal y Ponderación de Variables: En cada paso de tiempo, se aplica una capa adicional de procesamiento no lineal a cada variable a través de su propia red de selección de variables. Luego, las características procesadas se ponderan por sus pesos de selección de variables y se combinan para formar un vector de características agregadas.

2.7.3. Codificador estático de covariables

En contraste con otras arquitecturas de pronóstico de series temporales, el TFT ha sido cuidadosamente diseñado para integrar información de metadatos estáticos, utilizando codificadores GRN separados para generar cuatro vectores de contexto diferentes: c_s , c_e , c_c , y c_h . Estos vectores de contexto se conectan en varias ubicaciones en el decodificador de fusión temporal, donde las variables estáticas desempeñan un papel importante en el procesamiento [5]. Específicamente, esto incluye contextos para:

- **Selección de Variables Temporales (c_s):** Este contexto tiene como propósito ayudar en la selección de variables temporales relevantes. Contribuye a determinar cuáles características variables en el tiempo son más importantes para la tarea de pronóstico. La información codificada en c_s se deriva de la salida de la red de selección de variables estáticas (ζ) y es procesada por GRN_{c_s} .
- **Procesamiento Local de Características Temporales (c_c y c_h):** Estos contextos, c_c y c_h , se utilizan en el procesamiento local de características temporales. Juegan un papel en el análisis de variables que cambian con el tiempo y sus interacciones. Su propósito es mejorar la comprensión de las dinámicas temporales dentro de los datos.
- **Enriquecimiento de Características Temporales con Información Estática (c_e):** El contexto c_e se utiliza para enriquecer las características tem-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

porales con información estática. Esto significa que combina los datos que cambian con el tiempo con los metadatos constantes y estáticos, proporcionando una representación más completa e informativa para el pronóstico.

Los codificadores de covariables estáticas producen estos vectores de contexto, que se integran en diversas partes del decodificador de fusión temporal para permitir un pronóstico más preciso y consciente del contexto. Este enfoque permite que el TFT aproveche tanto la información que cambia con el tiempo como la información estática para obtener pronósticos mejorados.

2.7.4. Atención Multi-Cabecal Interpretativa

El mecanismo de autoatención del TFT está diseñado para aprender relaciones a largo plazo entre diferentes pasos de tiempo. Esto se hace mediante el uso de atención multicabecal, que permite al modelo centrarse en diferentes aspectos de los datos en diferentes momentos. El cual originalmente está configurado de la siguiente manera:

$$\text{Attention}(Q, K, V) = A(Q, K)V, \quad (2.21)$$

donde A es una función de normalización. Comúnmente se elige el producto punto escalado. Para mejorar la capacidad de aprendizaje del mecanismo de atención estándar, se propone la atención multi-cabecal en, utilizando diferentes cabezas para diferentes subespacios de representación:

$$\text{MultiHead}(Q, K, V) = [H_1, \dots, H_m]W_H, \quad (2.22)$$

$$H_h = \text{Attention}(QW(h)Q, KW(h)K, VW(h)V) \quad (2.23)$$

Dado que se utilizan diferentes valores en cada cabeza, los pesos de atención por sí solos no serían indicativos de la importancia de una característica en par-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

ticular. Por lo tanto, se emplea la agregación aditiva de todas las cabezas. Esto significa que las salidas de todas las cabezas se suman para obtener la salida final. Esto ayuda a mejorar el rendimiento del modelo [5].

$$\text{InterpretableMultiHead}(Q, K, V) = HW\tilde{H} \quad , \quad (2.24)$$

$$\tilde{H} = \tilde{A}(Q, K)VWV \quad , \quad (2.25)$$

$$\tilde{H} = \frac{1}{H} \sum_{h=1}^H A(QW^{(h)}Q, KW^{(h)}K) VWV \quad , \quad (2.26)$$

$$\tilde{H} = \frac{1}{H} \sum_{h=1}^H \text{Attention}(QW^{(h)}Q, KW^{(h)}K, VWV) \quad , \quad (2.27)$$

donde $WV \in \mathbf{R}^{d_{\text{model}} \times d_V}$ son pesos de valor compartidos entre todas las cabezas, y $WH \in \mathbf{R}^{d_{\text{attn}} \times d_{\text{model}}}$ se utiliza para el mapeo lineal final. A partir de la ecuación (15), observamos que cada cabeza puede aprender diferentes patrones temporales, mientras atiende a un conjunto común de características de entrada, lo que se puede interpretar como un simple conjunto sobre los pesos de atención en la matriz combinada $\tilde{A}(Q, K)$ en la ecuación (14). En comparación con $A(Q, K)$ en la ecuación (10), $\tilde{A}(Q, K)$ proporciona una mayor capacidad de representación de manera eficiente.

2.7.5. Temporal Fusion Decoder

El decodificador está compuesto por las siguientes capas:

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

Mejoramiento local con capas secuencia a secuencia

La significancia de los valores a menudo se identifica mediante su relación con los valores circundantes, ya que estos ayudan a describir anomalías, puntos de cambio y patrones cíclicos. Al construir capas que utilicen esta información, se mejora el rendimiento de las arquitecturas basadas en atención. Una forma de mejorar la calidad es mediante las capas de secuencia a secuencia.

La capa de secuencia a secuencia se puede dividir en tres partes principales [5]:

1. **El codificador:** Toma como entrada una secuencia de valores pasados, $\tilde{\xi}_{t-k:t}$, y genera una representación oculta de estos valores. Esta representación oculta captura la información importante de los valores pasados que es relevante para el procesamiento local.
2. **El decodificador:** Toma como entrada una secuencia de valores futuros, $\tilde{\xi}_{t+1:t+\tau_{\max}}$, y genera una secuencia de características temporales uniformes, $\phi(t, n) \in \{\phi(t, -k), \dots, \phi(t, \tau_{\max})\}$, donde n es un índice de posición. Estas características temporales uniformes se pueden utilizar como entradas para el decodificador de fusión temporal.
3. **La conexión de salto con puerta:** Permite al modelo aprender la forma más adecuada de procesar los datos de entrada. En particular, el modelo puede aprender si se debe pasar la entrada directamente a la salida o aplicar una transformación a la entrada antes de pasarla a la salida.

La capa de secuencia a secuencia tiene varias ventajas sobre otros enfoques para mejorar la localidad en datos de series temporales. En particular, es capaz de:

- Manejar el número diferente de entradas pasadas y futuras: Esto es importante para datos de series temporales, ya que el número de entradas

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

pasadas y futuras puede variar.

- Generar un conjunto de características temporales uniformes: Las características temporales uniformes son más fáciles de interpretar y utilizar que las características temporales irregulares.
- Permitir que los metadatos estáticos influyan en el procesamiento local: Esto puede mejorar el rendimiento de las arquitecturas basadas en atención para tareas como la predicción de series temporales y la clasificación de anomalías.

2.7.6. Capa de enriquecimiento estático

Descrito de la siguiente manera:

$$\theta(t, n) = \text{GRN}_{\theta} \left(\tilde{\phi}(t, n), \text{ce} \right) \quad (2.28)$$

Aquí, los pesos de la Red Residual Gated (GRN) para el enriquecimiento estático se comparten a lo largo de toda la capa, y se representa un vector de contexto proveniente de un codificador de variables estáticas [5].

Capa temporal de autoatención

Siguiendo a la capa anterior, se implementa esta nueva capa, donde todas las características estáticas enriquecidas se agrupan en una matriz $\Theta(t) = [\theta(t, -k), \dots, \theta(t, \tau_{\max})]^T$. Luego, la capa de atención multi-cabezal interpretativa se aplica a cada predicción. En el decodificador, se utiliza una máscara en la capa de atención multi-cabezal para garantizar que cada dimensión temporal pueda enfocarse en una sola característica. Además, para preservar el flujo de información de manera causal, la capa de autoatención permite aprender dependencias

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

a largo plazo. A la capa de atención también se le aplica una compuerta definida de la siguiente manera:

Siguiendo el enriquecimiento estático, aplicamos a continuación la autoatención. Todas las características temporales enriquecidas estáticamente se agrupan primero en una única matriz, es decir, $\Theta(t) = [\theta(t, -k), \dots, \theta(t, \tau)]^T$, y se aplica una atención multi-cabecal interpretable (ver Sección 4.4) en cada instante de pronóstico (con $N = \tau_{\max} + k + 1$):

$$B(t) = \text{InterpretableMultiHead}(\Theta(t), \Theta(t), \Theta(t)), \quad (2.29)$$

donde, para obtener $B(t) = [\beta(t, -k), \dots, \beta(t, \tau_{\max})]$. Se eligen $d_V = d_{\text{attn}} = \frac{d_{\text{model}}}{m_H}$, donde m_H es el número de cabezas. Se aplica enmascaramiento del decodificador [17, 12] a la capa de atención multi-cabecal para garantizar que cada dimensión temporal solo pueda atender a las características que la preceden. Además de preservar el flujo de información causal mediante el enmascaramiento, la capa de autoatención permite que TFT capture dependencias a largo plazo que podrían ser desafiantes para las arquitecturas basadas en RNN. Después de la capa de autoatención, también se aplica una capa adicional de gateo para facilitar el entrenamiento:

$$\delta(t, n) = \text{LayerNorm}(\theta(t, n) + \text{GLU}_{\delta}(\beta(t, n))) \quad (2.30)$$

2.7.7. Capa de Avance Posicional

Se aplica un procesamiento no lineal adicional a las salidas de la capa de autoatención. Este procesamiento se realiza mediante redes residuales gated (GRNs), que son una variante de las redes neuronales recurrentes.

En la capa de avance posicional (*Position-wise Feed-forward Layer*), se aplica un procesamiento no lineal adicional a las salidas de la capa de autoatención.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Este procesamiento se realiza mediante redes residuales gated (GRNs), que son una variante de las redes neuronales recurrentes [5].

Las GRNs permiten al modelo aprender relaciones más complejas entre las características temporales. Esto se debe a que las GRNs pueden aprender funciones no lineales de la entrada, lo que permite al modelo capturar dependencias más complejas entre los valores de tiempo.

Además del procesamiento no lineal, la capa de avance posicional también puede aplicar una conexión residual gestionada. Una conexión residual es una técnica que permite al modelo aprender si se debe pasar la entrada directamente a la salida o aplicar una transformación a la entrada antes de pasarla a la salida.

La conexión residual gestionada permite al modelo preservar la simplicidad del modelo si no se necesita complejidad adicional. Esto se debe a que la conexión residual permite al modelo omitir la capa de avance posicional si no es necesario realizar el procesamiento no lineal adicional.

La capa de avance posicional se puede expresar matemáticamente de la siguiente manera:

$$\psi(t, n) = \text{GRN}_{\psi}(\delta(t, n)), \quad (2.31)$$

donde $\psi(t, n)$ son las salidas de la capa de avance posicional y $\delta(t, n)$ son las salidas de la capa de autoatención. GRN_{ψ} es una red residual gated con pesos compartidos.

La conexión residual gestionada se puede expresar matemáticamente de la siguiente manera:

$$\psi(t, n) = \text{LayerNorm}(\phi(t, n) + \text{GLU}_{\psi}(\psi(t, n))), \quad (2.32)$$

donde $\psi(t, n)$ son las salidas de la capa de avance posicional con la conexión residual gestionada, $\phi(t, n)$ son las salidas de la capa de autoatención con la conexión residual gestionada, GLU_{ψ} es una función de activación de la unidad

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

lineal con puerta, y LayerNorm es una capa de normalización por capas.

2.7.8. Salidas Cuantiles

Producción de intervalos de predicción

El modelo TFT también puede generar intervalos de predicción sobre las predicciones puntuales. Esto se logra mediante la predicción simultánea de varios percentiles (por ejemplo, el 10º, 50º y 90º) en cada paso temporal.

Las predicciones de cuantiles se generan mediante una transformación lineal de la salida del decodificador de fusión temporal. La transformación lineal se define de la siguiente manera:

$$y'(q, t, \tau) = W_q \psi'(t, \tau) + b_q, \quad (2.33)$$

donde $y'(q, t, \tau)$ es la predicción del cuantil q en el tiempo t para el horizonte τ , $\psi'(t, \tau)$ es la salida del decodificador de fusión temporal, W_q es un vector de pesos lineal para el cuantil q , y b_q es un sesgo lineal para el cuantil q . Los coeficientes lineales W_q y b_q se aprenden durante el entrenamiento del modelo. Cabe destacar que las predicciones solo se generan para horizontes en el futuro, es decir, $t \in \{1, \dots, \tau_{\max}\}$. Esto significa que el modelo no puede predecir valores pasados.

2.7.9. Interpretabilidad

La interpretabilidad es la capacidad de comprender cómo y por qué un modelo genera una determinada predicción. En el contexto de los modelos de aprendizaje automático, la interpretabilidad es un desafío, ya que estos modelos suelen ser muy complejos y sus decisiones pueden ser difíciles de explicar.

La interpretabilidad se puede abordar de dos maneras [12]:

Interpretabilidad post-hoc: Esta estrategia consiste en aplicar técnicas de

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

interpretación a un modelo ya entrenado. Estas técnicas pueden ser de naturaleza estadística, gráfica o narrativa.

Interpretabilidad incorporada: Esta estrategia consiste en diseñar modelos que sean intrínsecamente interpretables. Esto se puede lograr utilizando arquitecturas de modelos que sean fáciles de entender o que proporcionen explicaciones explícitas de sus decisiones.

Interpretabilidad en Modelos de Series Temporales

La interpretabilidad en el ámbito de los modelos de series temporales es esencial para comprender y explicar cómo un modelo toma decisiones y realiza predicciones en datos temporales. Este aspecto adquiere una importancia fundamental en aplicaciones críticas, donde la transparencia y la comprensión del razonamiento del modelo son tan cruciales como su precisión. Proporcionar interpretabilidad permite a usuarios, analistas y profesionales validar y entender el proceso de toma de decisiones del modelo.

Interpretabilidad en TFT (Temporal Fusion Transformer)

Los modelos TFT incorporan elementos de interpretabilidad en su arquitectura, destacando [5] :

Capas de Atención: Permiten identificar las partes más relevantes de la secuencia de entrada para una predicción específica. Los pesos de atención indican la importancia relativa de diversas partes de la serie temporal.

Análisis de Importancia de Variables: Realizan un análisis detallado de la importancia de cada variable en la predicción. Examinan variables estáticas, entradas pasadas y futuras para comprender las características más influyentes.

Visualización de Patrones Temporales Persistentes: Utilizan los pesos de atención para visualizar patrones temporales persistentes, permitiendo entender

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

cómo el modelo considera ciertos momentos más relevantes.

Identificación de Regímenes y Eventos Significativos: La capacidad del modelo para identificar cambios significativos en los patrones temporales agrega otra capa de interpretabilidad. Revela cómo el modelo reacciona a eventos y cambios en la dinámica temporal.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

CAPÍTULO 3

Metodología Aplicada

El presente capítulo detalla el enfoque metodológico empleado para abordar la predicción de la demanda de energía a nivel nacional en este estudio. Se describe de manera exhaustiva el proceso de recolección, preparación y manipulación de los datos provenientes de la Administración Nacional de Electricidad (ANDE), así como la selección de variables y el ajuste de la arquitectura y la evaluación del sistema de predicción. Se presentan las fases clave del preprocesamiento de datos, el diseño experimental, la configuración de los modelos y las estrategias de evaluación utilizadas para validar y optimizar la precisión de las predicciones. La Figura 3.1 sintetiza los pasos aplicados en cada etapa. Este capítulo proporciona una visión detallada del marco metodológico que sustenta el desarrollo y la evaluación del modelo predictivo de demanda energética.

3.1. Obtención y preparación de datos

Es importante destacar que los datos preliminares utilizados en este trabajo son considerados datos públicos y oficiales de las respectivas entidades. El conjunto de datos ya preprocesado en gran parte en [7] fue reutilizado y complementado con datos actuales y recientes de la Dirección de Planificación de la ANDE, manteniendo el intervalo horario e incluyendo los años 2020, 2021 y 2022.

Se encuentran agrupadas todas las series relevantes en crudo para su análisis.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

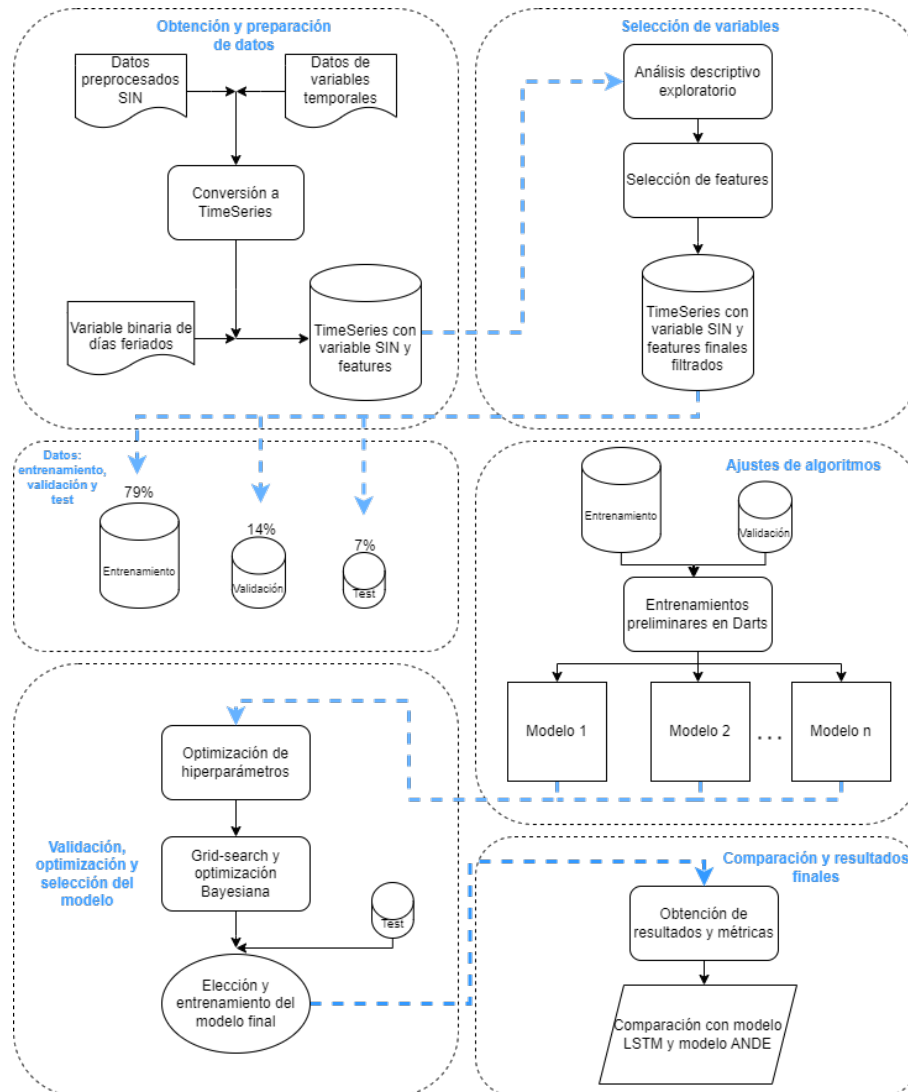


Figura 3.1: Esquema general de la metodología aplicada.

Fuente: Elaboración propia.

sis e inclusión en los distintos algoritmos de entrenamiento y validación. Además de la variable objetivo o target que es el SIN (MW), se incluyen series temporales de variables que son consideradas exógenas, que pueden impactar en el desenvolvimiento de la variable objetivo y dar lugar a predicciones más precisas. Por un lado, tenemos las variables meteorológicas disponibles que son la temperatura seca del aire y la humedad relativa, de distintas ciudades y sistemas regionales del país y, por otro lado, generamos variables temporales que indican la continuidad del tiempo en distintos intervalos (anual, mensual, semanal).

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

Para el caso de las variables exógenas meteorológicas, las mismas fueron preprocesadas en [7]. Para la temperatura se tuvo en cuenta el factor ponderación de cada ciudad, calculando la temperatura aparente mediante la fórmula de Steadman [17], mientras que para la humedad se utilizó la escala de Beaufort [18].

Las variables exógenas temporales de calendario fueron generadas por código, categorizando las mismas en Año, Mes y Día de la Semana y así poder interpretar su impacto en el consumo eléctrico. La serie correspondiente al Año consta del valor numérico respectivo al año analizado (2009, 2010, 2011, etc). Para los casos del Mes y el Día de la Semana, de forma a obtener una continuación secuencial lógica, en términos cuantitativos, se obtienen sus formas senoidales y cosenoidales.

3.2. Preprocesamiento de Datos

Las tareas necesarias para lograr un preprocesamiento típico a la hora de construir un análisis de datos pueden ser resumidas en cinco principales [19]: limpieza, reducción y selección, escalamiento y normalización, transformación y por último partición.

La primera tarea fue omitida para este trabajo debido al trabajo realizado en [7], así como gran parte de la reducción y selección de datos, adoptando intervalos de tiempo idénticos para lograr una comparación más precisa. Escalamiento y normalización La importancia de esta etapa radica en verificar y asegurarse que las escalas de las variables de entrada se encuentren en la misma escala, optando por normalizarlas en una escala [0, 1] y así mejorar el rendimiento y estabilidad del entrenamiento del modelo. La normalización de los datos como tal se realiza con métodos proveídos por la librería Darts que cuenta con su escalador y transformador nativo y se realiza en todo el conjunto de datos que abarca

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

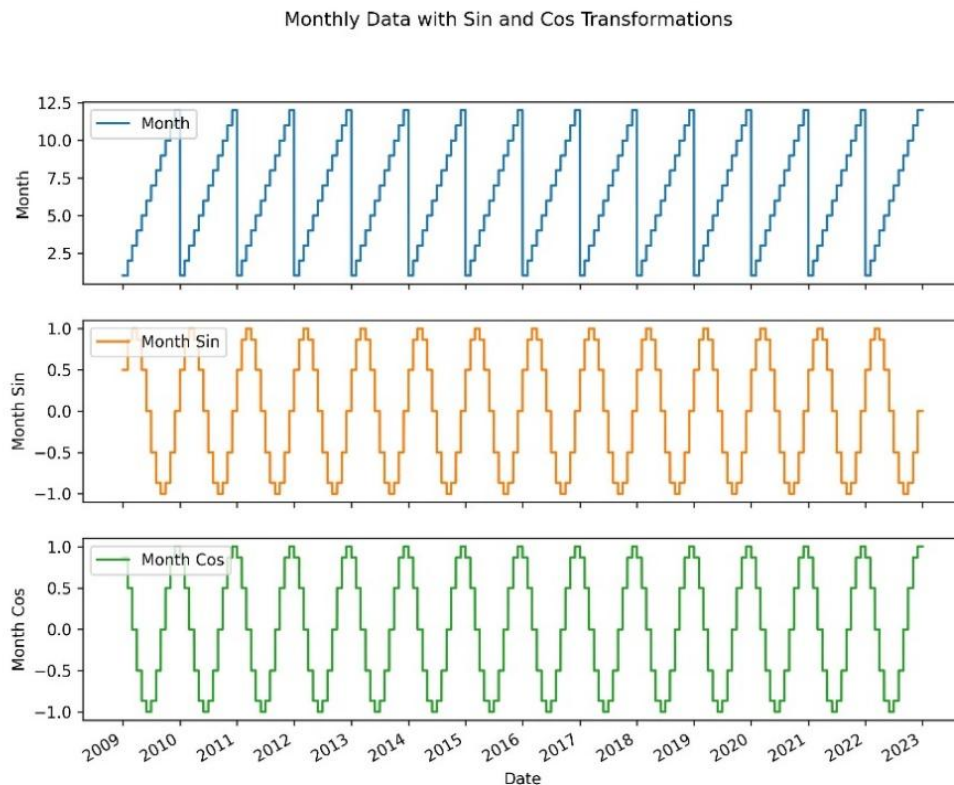


Figura 3.2: Representación gráfica de la variable Mes y sus transformaciones senoidales y cosenoidales.

Fuente: Elaboración propia.

desde el 2009 hasta el 2022.

Escalamiento y normalización

La importancia de esta etapa radica en verificar y asegurarse que las escalas de las variables de entrada se encuentren en la misma escala, optando por normalizarlas en una escala $[0, 1]$ y así mejorar el rendimiento y estabilidad del entrenamiento del modelo. La normalización de los datos como tal se realiza con métodos proveídos por la librería Darts que cuenta con su escalador y transformador nativo y se realiza en todo el conjunto de datos que abarca desde el 2009 hasta el 2022.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

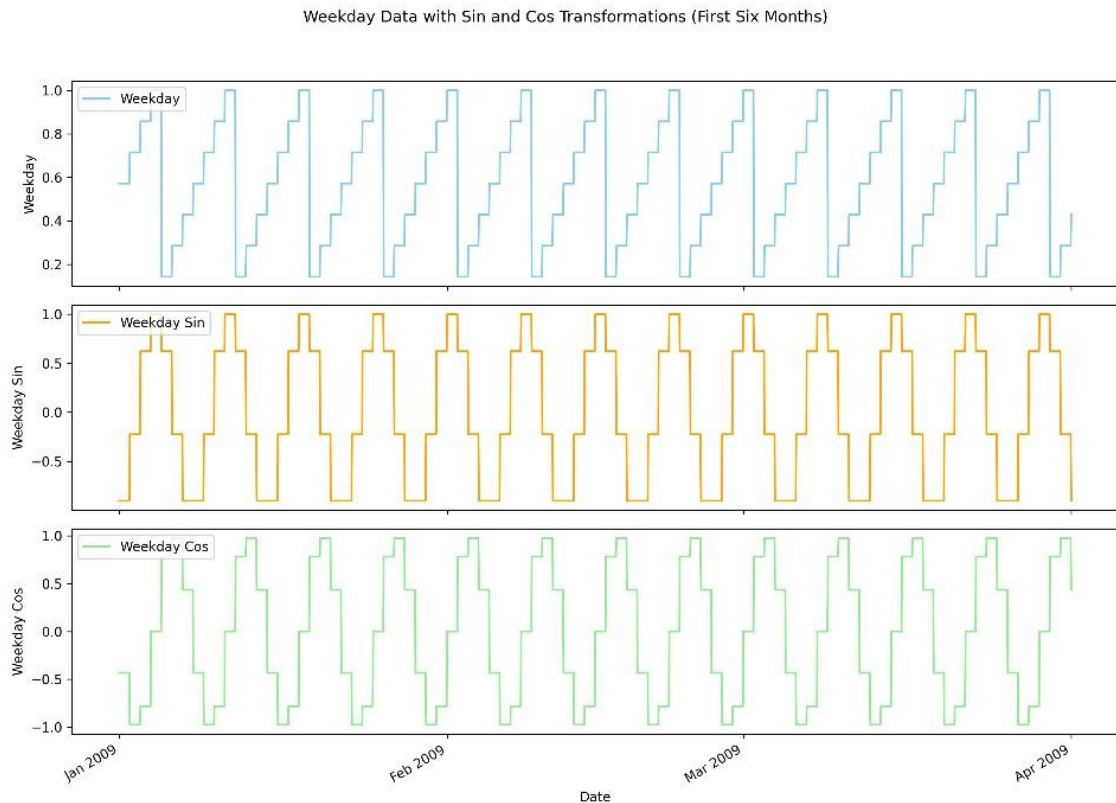


Figura 3.3: Representación gráfica de la variable Día de la Semana y sus transformaciones senoidales y cosenoidales.

Fuente: Elaboración propia.

Transformación de Datos

Es necesaria una conversión previa de formato para favorecer la compatibilidad con la librería utilizada en Python. Esta conversión consiste en pasar de DataFrames a TimeSeries, clase principal utilizada para el tratamiento de las series temporales en el entrenamiento de todos los modelos de lenguaje de máquina y aprendizaje profundo. Esto se consigue de forma rápida con métodos y funciones de programación del lenguaje utilizado, haciendo énfasis en que nuestro index es siempre la fecha en intervalos horarios.

Con la conversión realizada es posible concatenar la variable binaria temporal de feriados, que indica un valor alto (uno) en caso de ser una fecha festiva o un

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

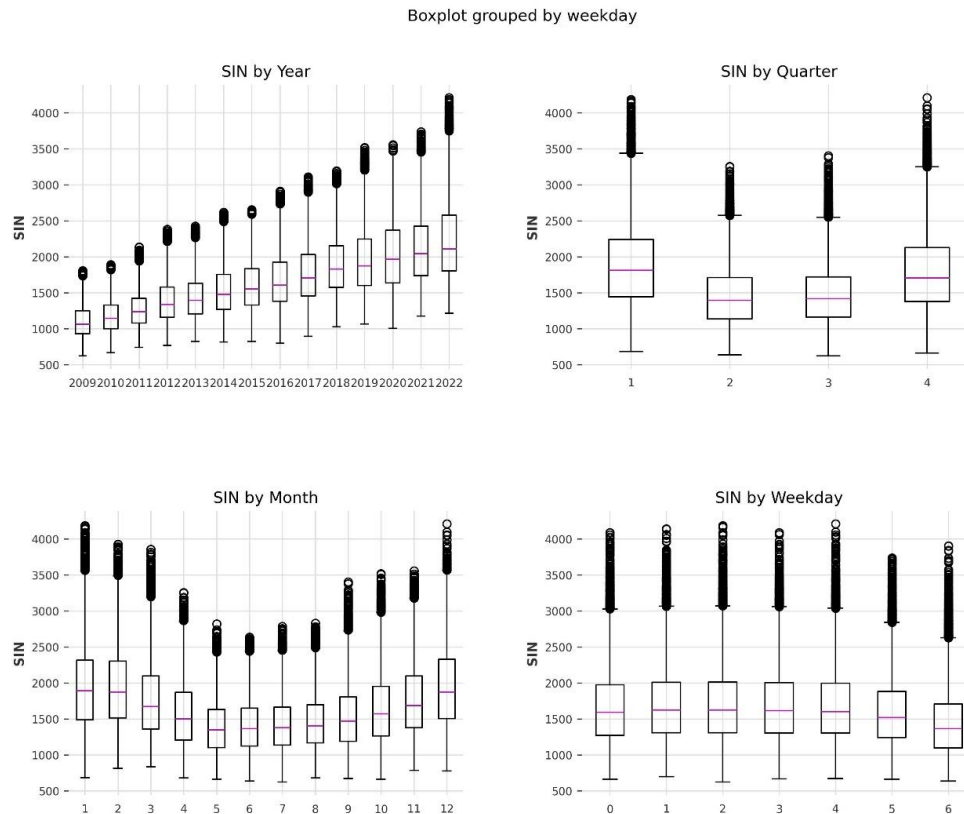


Figura 3.4: Gráfico Box-Plot del SIN para las variables temporales calendario

Fuente: Elaboración propia.

valor bajo (cero) en caso contrario. La consideración de esta variable tiene importancia en este trabajo ya que la demanda eléctrica generalmente se encuentra influenciada por el comportamiento de las personas en estas fechas [20]

Los feriados y festividades son determinadas por el gobierno nacional, departamental o municipal por ley. Sin embargo, estas fechas son sometidas a variaciones dependiendo de ciertos factores, creando pequeñas grietas en los patrones de comportamiento del modelo entrenado lo cual se ve reflejado en la Figura 3.5.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

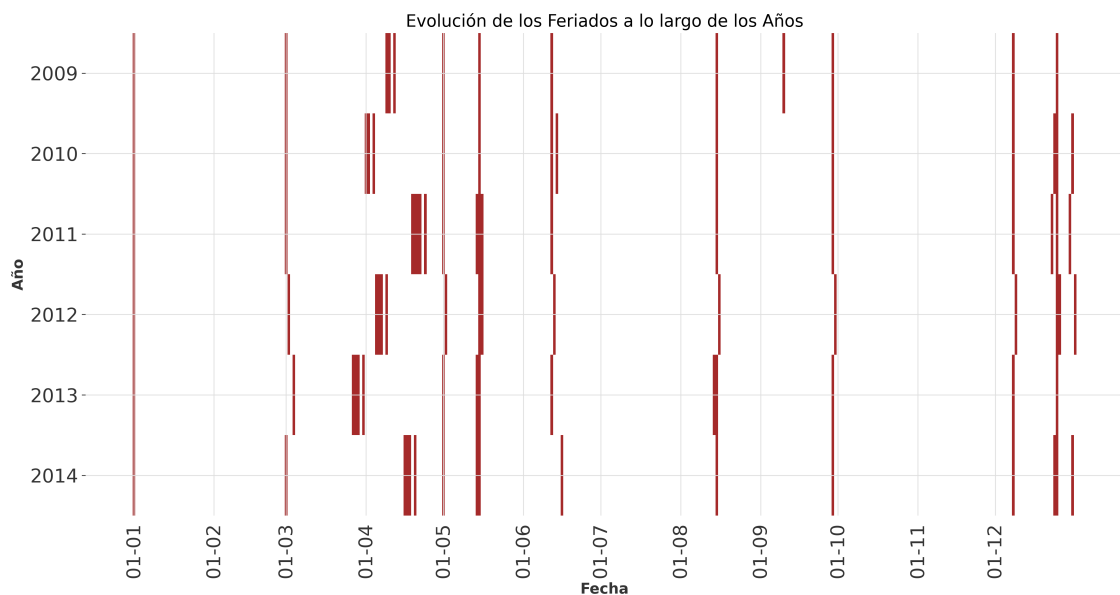


Figura 3.5: Distribución de los días feriados en los primeros cinco años del conjunto de datos.

Fuente: Elaboración propia.

Partición de Datos

El conjunto de entrenamiento se utiliza para entrenar el modelo. El conjunto de validación es utilizado para optimizar la precisión del modelo ajustando los hiperparámetros del modelo. Con el conjunto de prueba realizamos la comparación final de datos predichos con los observados para determinar las métricas correspondientes.

3.3. Modelos de Aprendizaje Profundo

La arquitectura del modelo utilizado en este trabajo corresponde a una RNN (Redes Neuronales Recurrentes) debido a la capacidad de aprender dinámicamente patrones temporales y poder tener acceso a un gran rango de fuentes de datos, incluyendo información sobre el futuro (variables temporales de calendario), otras series temporales exógenas (consumo histórico del SIN) e incluso variables estáticas que no son utilizadas para este caso, [5] aprovechando las

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

características de un modelo clásico de Transformer pero aplicado en series temporales [4].

3.4. Implementación en Darts

En el proceso de desarrollo del modelo se investigaron múltiples librerías en Python que pudieran ser útiles, dando con una en particular llamada Darts, una librería mucho más amigable al usuario para realizar predicciones y detectar anomalías en series temporales, tanto para modelos tradicionales estadísticos como para los más complejos, incluyendo el TFT. El entrenamiento de los modelos se realiza de la misma forma utilizando y basándose en la clásica librería scikit-learn, manteniendo las funciones `fit()` y `predict()`.

La función `fit` toma como argumento un conjunto de datos de datos de entrenamiento en formato `TimeSeries` y `predict` devuelve el resultado de la predicción también en formato `TimeSeries`, siendo prácticamente el único formato que manipula la librería.

3.5. Modelo TFT

El desarrollo de un modelo TFT en Darts se basa en la implementación realizada por otra librería conocida: PyTorch. El modelo aplica consultas de atención multi-cabeza a las entradas futuras a partir del conjunto de datos ingresado de forma obligatoria como *future_covariates*. Si se especifican codificadores futuros con el parámetro *add_encoders*, se pueden generar automáticamente variables futuras, lo que permite omitir la entrada del parámetro *future_covariates* al aplicar el entrenamiento (`fit`) y la predicción (`predict`).

Por defecto, el modelo usa la probabilidad de *QuantileRegression* (Regresión Cuantil), teniendo la capacidad de realizar predicciones probabilísticas. La arquitectura del modelo cuenta con ciertos hiperparámetros que deben ser elegidos

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

en base a la cantidad de datos que se va a manejar, la potencia computacional y el tiempo disponible, partiendo siempre desde un conjunto de valores iniciales, que se deberán ir ajustando con la realización de múltiples experiencias que puede tomar varios días e incluso semanas. En la Tabla 3.1 se describen los principales hiperparámetros que se tienen en cuenta para el entrenamiento del modelo, con el nombre utilizado dentro de las funciones de Darts:

Tabla 3.1: Descripción de hiperparámetros en la arquitectura TFT.

| Hiperparámetros | Descripción |
|---------------------|---|
| input_chunk_length | Número de pasos, en horas, que se mira para atrás |
| output_chunk_length | Número de pasos, en horas, predichos a partir del punto de predicción |
| hidden_size | El hiperparámetro principal y común a lo largo de la arquitectura TFT |
| lstm_layers | Número de capas para los codificadores y decodificadores LSTM |
| num_attention_heads | Número de cabezas de atención de la arquitectura |
| dropout | Fracción de neuronas afectadas por el dropout |
| batch_size | Número de pasos (input y output) utilizado en cada iteración de entrenamiento |

Fuente: Elaboración propia.

Una práctica estandarizada es la de dejar valores o propiedades fijas a ciertos hiperparámetros y al resto, comúnmente los relevantes al desempeño, para un algoritmo metaheurístico que optimice esos valores o propiedades [21]. Para el caso de este trabajo se eligieron valores considerados razonables, Tabla 3.2 para partir de un entrenamiento inicial e ir modificando según los resultados.

3.6. Definición de las covariables

Anteriormente se mencionó la inclusión de variables exógenas que acompañan a la variable objetivo SIN (MW) en el conjunto de datos. Estas en total son siete y serán consideradas como covariables de entrada para el modelo, propor-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 3.2: Valores iniciales de hiperparámetros en la arquitectura TFT.

| Hiperparámetros | Valor inicial |
|---------------------|-----------------------------|
| input_chunk_length | 336 pasos de tiempo (horas) |
| output_chunk_length | 24 pasos de tiempo (horas) |
| hidden_size | 64 |
| lstm_layers | 1 |
| num_attention_heads | 4 |
| dropout | 0.1 |
| batch_size | 16 |

Fuente: Elaboración propia.

cionando información adicional y contexto para mejorar la precisión de la predicción. Es necesario definirlas como covariables pasadas, covariables futuras o covariables estáticas y en base a [22] se observa lo mencionado en la Tabla 3.3.

Probablemente llame la atención que todas sean definidas como covariables futuras, esto es debido a la siguiente regla: Si los valores son conocidos por adelantado, entonces son covariables futuras (o pueden ser utilizados también como covariables pasadas). Caso contrario, son covariables pasadas [22]. Esto depende fuertemente también del punto y horizonte de predicción que tomemos según los intervalos temporales que manejemos, siendo nuestro horizonte de predicción el año 2022, periodo que tomamos como conjunto de datos de prueba. Por lo tanto al conocer de antemano los valores de las covariables en este año, se consideran como futuras.

Tabla 3.3: Definición de las covariables.

| Covariables | Tipo de Covariable |
|-------------------------------|--------------------|
| Temperatura | Covariable futura |
| Humedad | Covariable futura |
| Año | Covariable futura |
| Mes (cosenoidal) | Covariable futura |
| Mes (senoidal) | Covariable futura |
| Día de la Semana (cosenoidal) | Covariable futura |
| Día de la Semana (senoidal) | Covariable futura |

Fuente: Elaboración propia.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

3.7. Optimización de velocidad con EarlyStopper

Debido a la arquitectura compleja del modelo TFT, el tiempo de entrenamiento y la posterior optimización de hiperparámetros puede resultar bastante largo, incluso contando con GPUs consideradas potentes como en nuestro caso, al poseer una NVIDIA RTX 3070 Ti (Laptop). Es por eso que la librería Darts acepta como parámetro de entrada un EarlyStopper, una función heredada de PyTorch, que permite controlar la mejoría y convergencia del entrenamiento en cada iteración y parar el mismo en caso de no presentar mejoras, según la función de evaluación definida y la paciencia establecida. Esta función puede ser el valor de pérdida en la validación (`val_loss`), el valor de pérdida en el entrenamiento (`train_loss`), que no es recomendado, o bien, alguna función de métricas importada de PyTorch como el MAPE.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

CAPÍTULO 4

Análisis e Interpretación de los Resultados

En este capítulo se exponen las distintas etapas atravesadas para llegar a un modelo estable y preciso, basándonos en múltiples entrenamientos experimentales y preliminares, junto con las métricas correspondientes. Por último, se establecen las comparativas con los modelos de [7] y la ANDE para determinar la viabilidad de la arquitectura en este contexto, teniendo en cuenta que el manejo de datos y los features y covariables utilizados tienen leves diferencias.

4.1. Análisis de convergencia con EarlyStopper

A partir de las consideraciones iniciales y definiciones de covariables el modelo puede ser entrenado para obtener resultados preliminares. Para inicializar los primeros entrenamientos nos enfocamos en definir y seleccionar las funciones de pérdida que mejor realicen el control de la convergencia del modelo, así como sus respectivos valores para el EarlyStopper. Para el primer entrenamiento seleccionamos como función de validación al MAPE (Mean Absolute Percentage Error), a ser comparado cada quince iteraciones de entrenamiento, con un delta mínimo de 0,05, logrando un modelo inicial Tabla 4.1 luego de 5 horas de entrenamiento.

Comparando la totalidad de los resultados se realiza un análisis más profundo para poder encontrar el día con mejor desempeño en el intervalo de validación, dando como resultado la fecha 16-08-2020, según podemos observar en la Figu-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.1: Métricas generales del primer modelo

| | |
|----------|---------------|
| R2 | 0,63163 |
| MSE | 106746,695 |
| RMSE | 326,721 (MW) |
| MAXERROR | 2096,142 (MW) |

Fuente: Elaboración propia.

ra 4.1, obteniendo un valor RMSE de 50.2723 (MW).

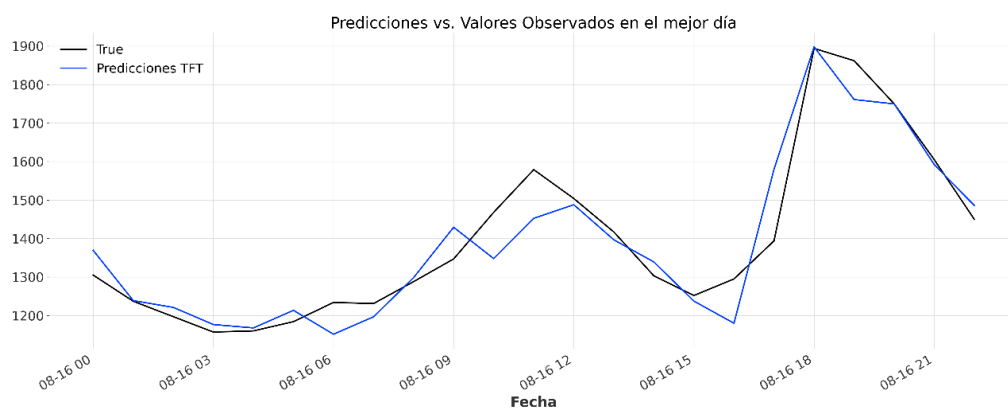


Figura 4.1: Comparación de valores observados con predichos del mejor día 16-08-2020 para el modelo inicial entrenado

Fuente: Elaboración propia.

A partir de los primeros resultados, evaluamos la posibilidad de cambiar la función de pérdida para el EarlyStopper y así verificar si hay mejoría y, además, mayor o menor velocidad de convergencia, incluyendo entrenamientos con `val_loss` y `train_loss`. Se observa que con estas dos funciones el modelo empeoró según las métricas, especialmente para el caso de `train_loss`, como se muestra en la Tabla 4.2, fijamos al MAPE y re-evaluamos los valores de iteraciones a controlar y los deltas mínimos.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.2: Comparación de métricas con funciones distintas de validación.

| Métrica | Valor con val_loss | Valor con train_loss |
|---------------|--------------------|----------------------|
| R2 | 0,58931 | 0,54169 |
| MSE | 119013,6 | 132810 |
| RMSE (MW) | 344,983 | 364,431 |
| MAXERROR (MW) | 2096,142 | 1745,375 |

Fuente: Elaboración propia.

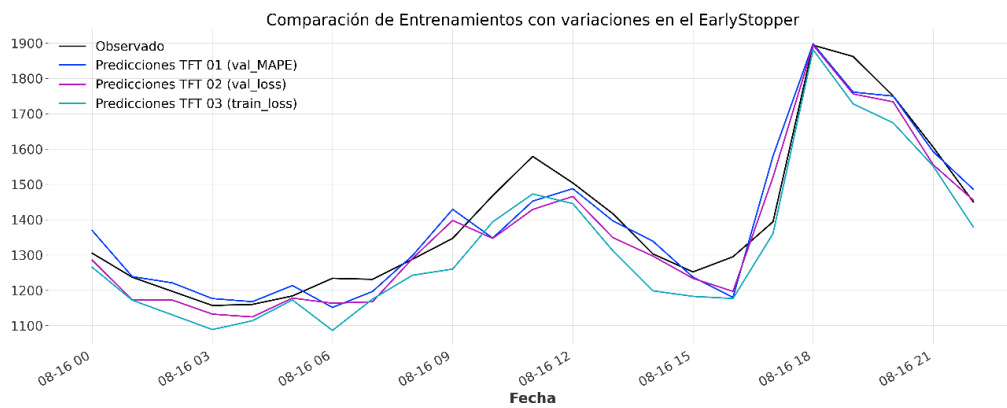


Figura 4.2: Comparación del mismo modelo con funciones de validación distintas para el EarlyStopper.

Fuente: Elaboración propia.

El modelo en la Figura 4.2 tiende a converger más rápido y devolver mejores resultados a medida que se ingresan valores más estrictos para el control según se observa en la Tabla 4.3, finalizando esta etapa preliminar con un valor de paciencia de 2 y un delta mínimo de 0,0001. Es importante resaltar que estos modelos iniciales fueron entrenados con un valor máximo de 50 iteraciones o epochs y en ningún caso se llegó hasta el final debido al EarlyStopper.

4.2. Impacto de la covariable Feriados

Los primeros análisis y entrenamientos de los modelos fueron realizados con las covariables mencionadas en la Tabla 3.3 sin considerar los días feriados. Una vez determinada la mejor función de convergencia podemos proceder a con-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.3: Comparación de métricas para distintos valores de paciencia y Δ usando el MAPE como función de validación.

| Función | Paciencia | Δ | R2 | MSE | RMSE | MAXERROR |
|---------|-----------|----------|------|----------|---------|----------|
| MAPE | 20 | 0,1 | 0,67 | 95532,83 | 309,08 | 1617,742 |
| MAPE | 15 | 0,15 | 0,68 | 94099,35 | 306,756 | 1805,332 |
| MAPE | 2 | 0,0001 | 0,74 | 79900,2 | 282,666 | 1851,334 |

Fuente: Elaboración propia.

catenar esta variable y establecer si el patrón de comportamiento de la misma incide de forma positiva o negativa en las métricas del modelo. Primeramente, podemos inferir la relación entre la variable Feriados y la variable objetivo SIN utilizando gráficas de la Figura 4.3, que nos permite comparar la tendencia central y la propagación del consumo de energía en días feriados o no feriados, y de la Figura 4.4, que nos da una perspectiva acerca de la densidad y estructura de la distribución de los datos del SIN en los mencionados días. Según lo observado se puede afirmar que en días feriados el consumo a nivel nacional es levemente inferior a los días normales, pero a un nivel que se puede considerar significativo como para ser incluido como covariable dentro del modelo.

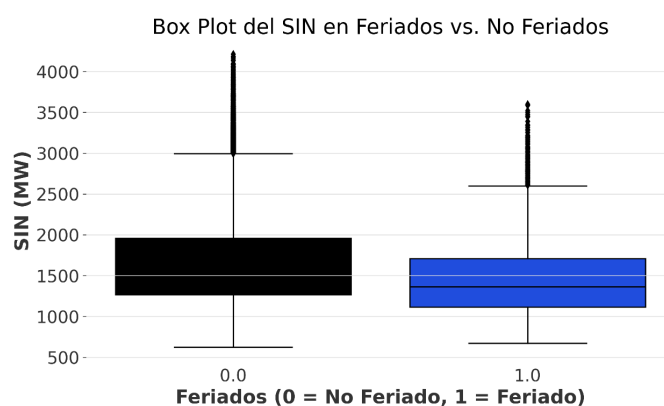


Figura 4.3: Distribución del SIN en forma de Caja para Feriados y No Feriados

Fuente: Elaboración propia.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

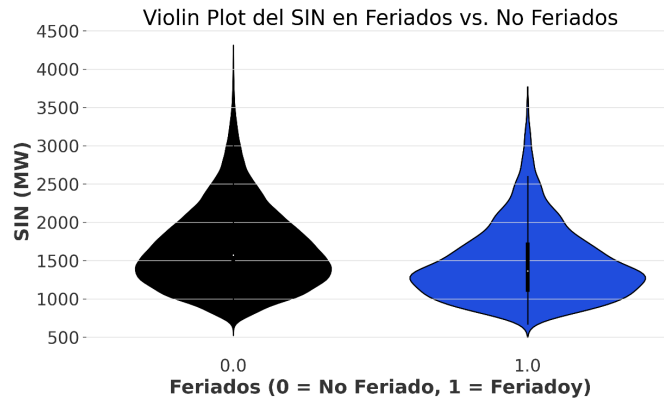


Figura 4.4: Distribución del SIN en forma de Violin para Feriados y No Feriados

Fuente: Elaboración propia.

Como se mencionó anteriormente en la metodología, se agrega esta covariable de Feriados utilizando una función nativa del Darts a la lista ya existente y se entrena nuevamente el modelo sin modificar los valores de los hiperparámetros, manteniendo el último EarlyStopper más óptimo. Los resultados arrojados se encuentran bastante cercanos al modelo sin Feriados según podemos observar en la Tabla 4.4. Comparando las métricas analizadas por sobre todo el conjunto de datos de validación existe una diferencia leve pero con valores mayores indicando un rendimiento mínimamente inferior (de aproximadamente unos 7 MW).

Tabla 4.4: Comparación de modelos sin covariable Feriados vs modelo con covariable Feriados.

| Métrica | Valor sin "Feriados" | Valor con "Feriados" |
|---------------|----------------------|----------------------|
| R2 | 0,74 | 0,711 |
| MSE | 79900,2 | 83712,97 |
| RMSE (MW) | 282,666 | 289,34 |
| MAXERROR (MW) | 1851,334 | 1980,201 |

Fuente: Elaboración propia.

No obstante, el objetivo de incluir esta covariable es por su comportamiento en períodos específicos de concentración de días festivos, siendo los meses más relevantes Enero y Diciembre. Es por ello que se realiza un enfoque en los re-

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

sultados predichos en semanas puntuales en estos intervalos mencionados, tal y como se observa en las Figuras 4.5 y 4.6.

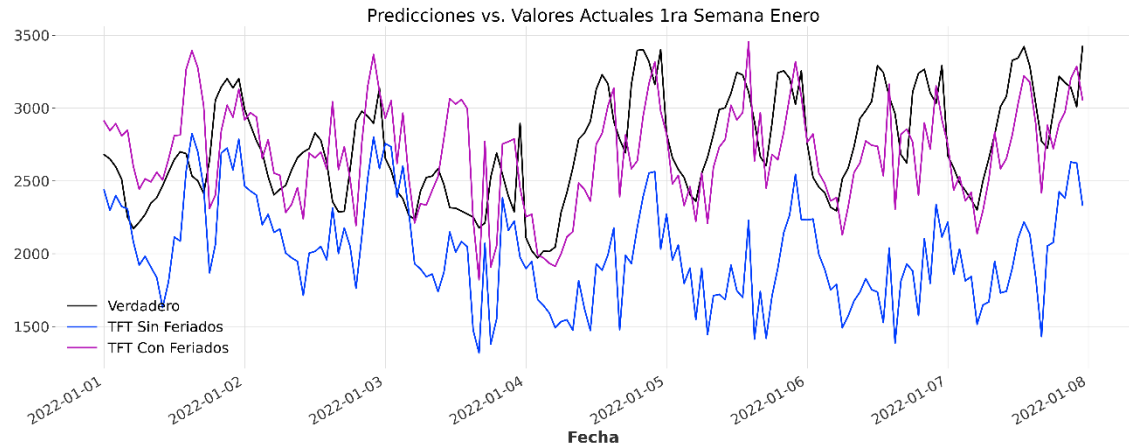


Figura 4.5: Comparación de resultados para modelo con y sin covariable Feriado en la primera semana de Enero.

Fuente: Elaboración propia.

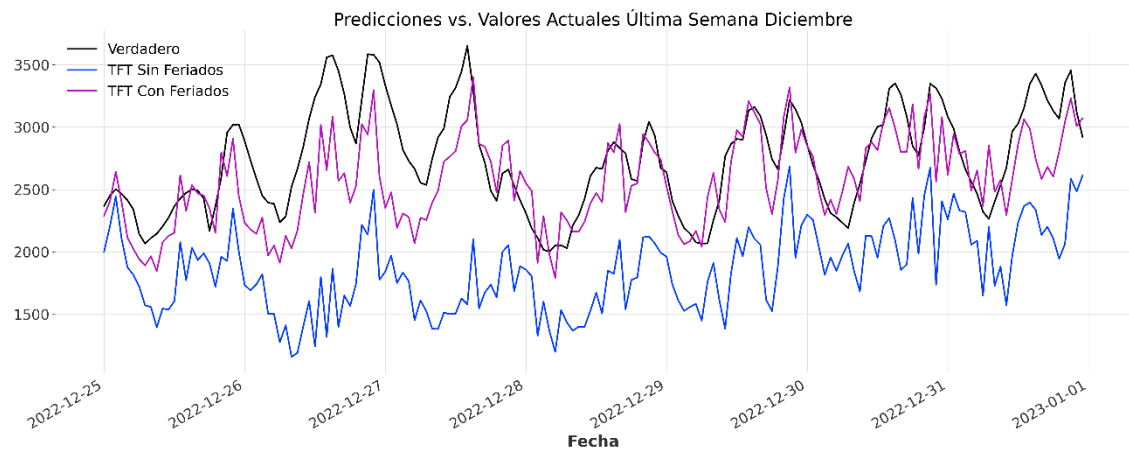


Figura 4.6: Comparación de resultados para modelo con y sin covariable Feriado en la última semana de Diciembre.

Además, aprovechando las herramientas proveídas por Darts, podemos obtener gráficos de “explicabilidad” que, según lo mencionado en la metodología, nos permite visualizar la importancia dada por nuestro modelo a la covariable Feriados según la Figura 4.7. Se observa que para el codificador de entrada la covariable Feriado (holidays) ocupa el cuarto lugar en importancia y atención, a

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

la par de las covariables de Mes tanto senoidal como coseinodal, así como el tercer lugar para el decodificador, solamente por detrás de la Temperatura y la variable temporal de calendario ascendente Año, ambas de suma importancia en el entrenamiento y predicción de cualquier modelo de predicción con series temporales.

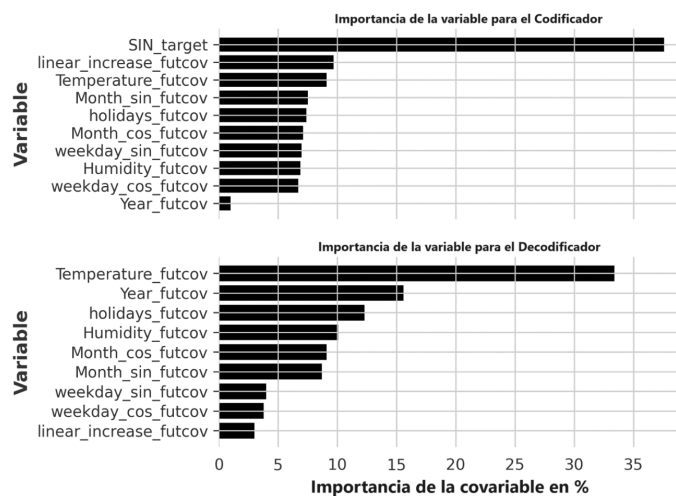


Figura 4.7: Gráficos de explicabilidad de las variables de entrenamiento del modelo

Fuente: Elaboración propia.

Considerando toda esta información y teniendo en cuenta que la caída de precisión es muy pequeña considerando que los feriados pueden tener mayor impacto si aumentan en gran cantidad, decidimos dejar esta covariable como fija para los siguientes entrenamientos en donde se busca la optimización del modelo.

4.3. Optimización de hiperparámetros con Optuna

Con los hiperparámetros iniciales utilizados para los entrenamientos preliminares y teniendo definida la totalidad de covariables a utilizar junto con el EarlyStopper adecuado, es posible realizar la optimización de los hiperparámetros de modo a obtener el modelo más eficiente en este contexto. Existen múltiples ma-

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

neras de realizarlo y para este trabajo se utilizó el framework Optuna [23], debido a su versatilidad y eficiencia en la optimización de modelos de lenguaje de máquinas y aprendizaje profundo. El intervalo de valores que pueden adoptar los hiperparámetros se observa en la Tabla 4.5. , mientras que la función utilizada para comparar cada modelo devuelto en el proceso de optimización es el SMAPE (Error simétrico medio porcentual absoluto).

Tabla 4.5: Intervalo de valores utilizados en la optimización de hiperparámetros.

| Hiperparámetro | Valor mínimo | Valor máximo |
|---------------------|---------------------|---------------------|
| input_chunk_length | 14 días (336 horas) | 30 días (720 horas) |
| output_chunk_length | 1 día (24 horas) | 1 día (24 horas) |
| hidden_size | 32 | 64 |
| lstm_layers | 1 | 2 |
| num_attention_heads | 2 | 4 |
| dropout | 0,0 | 0,4 |
| batch_size | 16 | 32 |

Fuente: Elaboración propia.

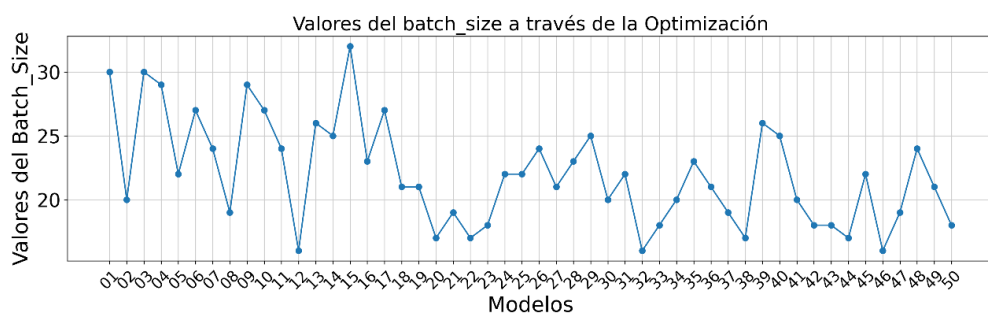
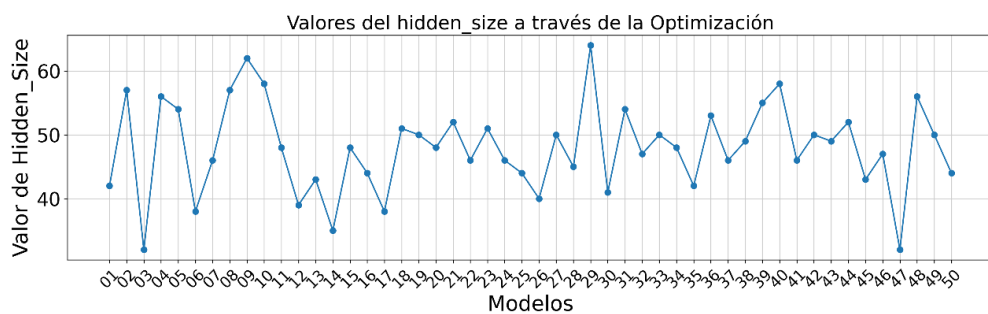
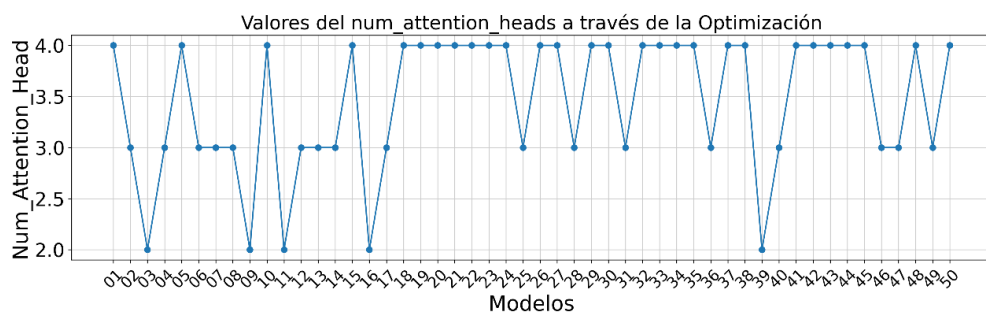
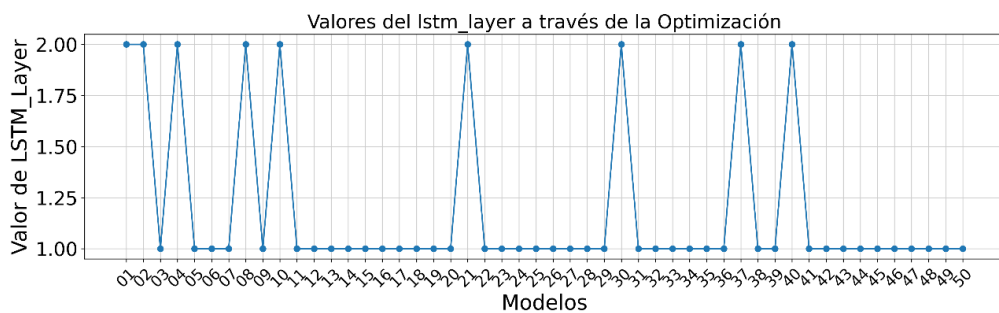
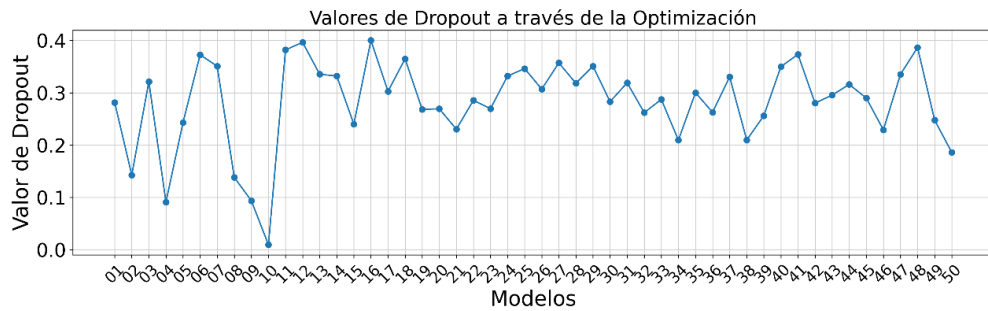
4.3.1. Optimización con tiempo de parada

Se realiza primeramente un estudio inicial de optimización utilizando la funcionalidad de tiempo de parada, con un valor de 7200 segundos, debido a la gran demanda computacional que genera el entrenamiento del modelo, que corta el proceso de optimización una vez finalizado el tiempo asignado.

4.3.2. Optimización con límite de iteraciones

A pesar de obtener resultados relativamente aceptables, se procedió a realizar una optimización completa en base a un número fijo de 50 iteraciones, con la posibilidad de obtener un mejor utilizando un tiempo aproximado de 4 días completos.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL



DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

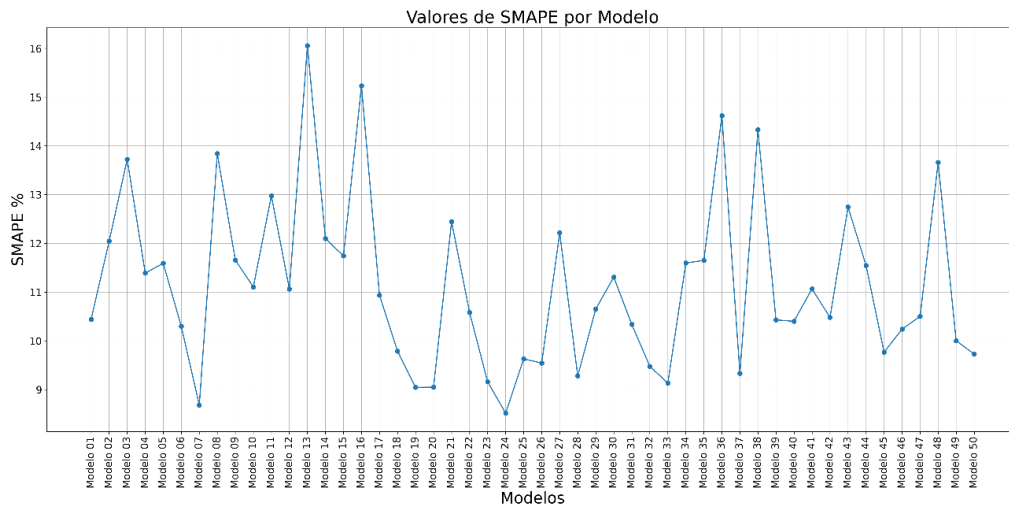


Figura 4.8: Evolución del valor del SMAPE para cada modelo en la optimización por Optuna.

Fuente: Elaboración propia.

Es posible observar en la Figura 4.8 el valor que va tomando cada hiperparámetro a lo largo de la optimización con el objetivo de abarcar un amplio espectro de valores en el intervalo asignado.

Según la Figura 4.8, se determina que el mejor modelo obtenido es el número 24, cuyo análisis exhaustivo se realiza posteriormente. Aún así, se toman los 5 mejores modelos junto con sus respectivos conjuntos de valores para realizar las comparaciones pertinentes de los mismos.

4.4. Análisis de los modelos TFT

El resumen de los mejores cinco conjuntos de hiperparámetros extraídos de la optimización se observa en la Tabla 4.6. A partir de estos valores se realiza un re-entrenamiento de estos modelos para obtener las métricas correspondientes. Es importante aclarar que se realiza la desnormalización de los resultados para obtener una mejor visualización y panorama en las comparaciones finales.

El resultado de las métricas aplicadas en el conjunto de datos de prueba se puede observar en la Tabla 4.7.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.6: Mejores conjuntos de hiperparámetros.

| Modelo | Input Chunk Length | Output Chunk Length | Hidden Size | LSTM Layers | Num Attention | Drop out | Batch Size |
|--------|--------------------------|---------------------------|----------------|----------------|------------------|-------------|---------------|
| 1 | 456 | 24 | 46 | 1 | 3 | 0.3505 | 24 |
| 2 | 480 | 24 | 50 | 1 | 4 | 0.2681 | 21 |
| 3 | 408 | 24 | 48 | 1 | 4 | 0.2693 | 17 |
| 4 | 408 | 24 | 46 | 1 | 4 | 0.3319 | 22 |
| 5 | 432 | 24 | 50 | 1 | 4 | 0.2872 | 18 |

Fuente: Elaboración propia.

Tabla 4.7: Métricas aplicadas a todo el conjunto de prueba.

| Modelo | MSE | RMSE | R2 | MAXERROR |
|--------|----------|--------|-------|----------|
| 1 | 76757.93 | 277.05 | 0.735 | 1728.80 |
| 2 | 92928.09 | 304.84 | 0.679 | 1700.03 |
| 3 | 68073.19 | 260.91 | 0.765 | 1761.95 |
| 4 | 62221.84 | 249.44 | 0.785 | 1599.63 |
| 5 | 87992.68 | 296.64 | 0.696 | 1923.10 |

Fuente: Elaboración propia.

El análisis de los resultados puede desglosarse en intervalos y/o períodos temporales para poder sacar mejores conclusiones y, para este Trabajo Final de Grado, se analizarán períodos trimestrales, semanales y por hora.

4.4.1. Análisis trimestral

Analizando el comportamiento de los modelos a partir de las métricas obtenidas por trimestre en la Tabla 4.8, podemos determinar que para todos los modelos sin excepción los mejores resultados se obtienen para el primer trimestre (Enero, Febrero y Marzo), mientras que los resultados menos satisfactorios se producen durante el tercer trimestre (Julio, Agosto y Septiembre

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.8: Métricas de los modelos por trimestre.

| Modelo | Trimestre | Métricas | | | |
|--------|-----------|-----------|--------|-------|----------|
| | | MSE | RMSE | R2 | MAXERROR |
| 1 | 1 | 53990.67 | 232.36 | 0.787 | 1213.32 |
| | 2 | 67452.35 | 259.72 | 0.686 | 1518.46 |
| | 3 | 103686.80 | 322 | 0.723 | 1728.80 |
| | 4 | 81901.91 | 286.19 | 0.657 | 1456.05 |
| 2 | 1 | 63178.36 | 251.35 | 0.750 | 1113.70 |
| | 2 | 72920.55 | 270.04 | 0.661 | 1448.72 |
| | 3 | 132089.12 | 363.44 | 0.647 | 1700.03 |
| | 4 | 103524.36 | 321.75 | 0.566 | 1553.45 |
| 3 | 1 | 44674.4 | 211.36 | 0.824 | 1287.02 |
| | 2 | 61631.02 | 248.26 | 0.714 | 1498.21 |
| | 3 | 91657.27 | 302.75 | 0.755 | 1761.95 |
| | 4 | 74330.06 | 272.64 | 0.688 | 1243.17 |
| 4 | 1 | 46229.84 | 215.01 | 0.817 | 1203.28 |
| | 2 | 56023.16 | 230.27 | 0.754 | 1419.63 |
| | 3 | 85890.81 | 293.07 | 0.770 | 1599.63 |
| | 4 | 63743.53 | 252.47 | 0.733 | 1152.52 |
| 5 | 1 | 54183.82 | 232.77 | 0.786 | 1122.37 |
| | 2 | 74018.86 | 272.06 | 0.656 | 1399.09 |
| | 3 | 116656.70 | 341.55 | 0.688 | 1923.10 |
| | 4 | 107111.32 | 327.28 | 0.551 | 1686.27 |

Fuente: Elaboración propia.

4.4.2. Análisis por día de la semana

Según las métricas obtenidas presentes en la Tabla 4.9 para el análisis semanal de los modelos, se observa que en cuatro de los cinco modelos el día con mejor desempeño es el Sábado, mientras que, para el caso del desempeño menos preciso, también en cuatro de los cinco modelos, el día es Lunes.

4.4.3. Análisis y evaluación del mejor modelo

A partir de esta sección nos enfocamos únicamente en el mejor modelo producido del conjunto de hiper-parámetros evaluados anteriormente, el número 4.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.9: Métricas de los modelos por día de la semana.

| Modelo | Día | Métricas | | | |
|--------|-----|-----------|--------|-------|----------|
| | | MSE | RMSE | R2 | MAXERROR |
| 1 | Lun | 122139.63 | 349.49 | 0.660 | 1728.80 |
| | Mar | 112899.72 | 336.00 | 0.693 | 1658.36 |
| | Mie | 125033.40 | 353.60 | 0.671 | 1477.85 |
| | Jue | 117768.40 | 343.17 | 0.688 | 1354.39 |
| | Vie | 112404.48 | 335.26 | 0.688 | 1354.39 |
| | Sáb | 88111.66 | 296.84 | 0.696 | 1420.18 |
| | Dom | 94121.79 | 306.79 | 0.658 | 1206.80 |
| 2 | Lun | 148681.11 | 385.59 | 0.586 | 1700.03 |
| | Mar | 126834.34 | 356.14 | 0.655 | 1683.07 |
| | Mie | 141248.64 | 375.83 | 0.628 | 1576.28 |
| | Jue | 136481.56 | 369.43 | 0.611 | 1275.70 |
| | Vie | 137070.53 | 370.23 | 0.619 | 1519.44 |
| | Sáb | 117311.05 | 342.51 | 0.596 | 1551.43 |
| | Dom | 107854.40 | 328.41 | 0.608 | 1299.94 |
| 3 | Lun | 121467.76 | 348.52 | 0.662 | 1538.69 |
| | Mar | 106072.48 | 325.69 | 0.712 | 1761.95 |
| | Mie | 120224.66 | 346.73 | 0.684 | 1460.10 |
| | Jue | 110263.80 | 332.06 | 0.685 | 1317.03 |
| | Vie | 99787.47 | 315.89 | 0.723 | 1408.52 |
| | Sáb | 78439.66 | 280.07 | 0.730 | 1376.68 |
| | Dom | 82883.00 | 287.89 | 0.699 | 1181.81 |
| 4 | Lun | 105489.96 | 324.79 | 0.706 | 1599.63 |
| | Mar | 85192.68 | 291.88 | 0.768 | 1448.26 |
| | Mie | 98948.99 | 314.56 | 0.740 | 1288.15 |
| | Jue | 93233.84 | 305.34 | 0.734 | 1313.93 |
| | Vie | 89683.37 | 299.47 | 0.751 | 1223.42 |
| | Sáb | 77266.11 | 277.97 | 0.734 | 1313.04 |
| | Dom | 79349.66 | 281.69 | 0.712 | 1280.35 |
| 5 | Lun | 168049.16 | 409.94 | 0.532 | 1923.10 |
| | Mar | 135742.75 | 368.43 | 0.631 | 1632.48 |
| | Mie | 142770.89 | 377.85 | 0.624 | 1461.38 |
| | Jue | 129298.36 | 359.58 | 0.631 | 1434.50 |
| | Vie | 125213.62 | 353.85 | 0.652 | 1332.63 |
| | Sáb | 114749.53 | 338.75 | 0.604 | 1511.92 |
| | Dom | 130612.31 | 361.40 | 0.526 | 1529.69 |

Fuente: Elaboración propia.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Tabla 4.10: Métricas por hora del conjunto de prueba.

| HR | MSE | RMSE | R2 | MAX ERROR | HR | MSE | RMSE | R2 | MAX ERROR |
|----|-------|------|------|-----------|----|--------|------|------|-----------|
| 00 | 62539 | 250 | 0.85 | 1023 | 12 | 132247 | 364 | 0.67 | 1600 |
| 01 | 57017 | 239 | 0.84 | 874 | 13 | 121732 | 349 | 0.73 | 1448 |
| 02 | 42249 | 206 | 0.86 | 874 | 14 | 93002 | 305 | 0.79 | 1309 |
| 03 | 47216 | 217 | 0.82 | 856 | 15 | 65272 | 255 | 0.81 | 810 |
| 04 | 58995 | 243 | 0.73 | 744 | 16 | 118011 | 344 | 0.47 | 1034 |
| 05 | 43787 | 209 | 0.73 | 955 | 17 | 85736 | 293 | 0.50 | 994 |
| 06 | 34850 | 187 | 0.76 | 812 | 18 | 94471 | 307 | 0.51 | 980 |
| 07 | 47418 | 218 | 0.71 | 1071 | 19 | 137181 | 371 | 0.48 | 1313 |
| 08 | 63780 | 253 | 0.66 | 833 | 20 | 110496 | 332 | 0.69 | 1071 |
| 09 | 92315 | 304 | 0.58 | 969 | 21 | 93141 | 305 | 0.79 | 1059 |
| 10 | 72476 | 269 | 0.67 | 958 | 22 | 86622 | 294 | 0.81 | 1078 |
| 11 | 97292 | 312 | 0.66 | 1339 | 23 | 298461 | 546 | 0.39 | 1288 |

Fuente: Elaboración propia.

Primeramente, podemos observar el comportamiento de este modelo para cada hora en el día como se observa en la Tabla 4.10, para determinar durante que horas se desempeña mejor o peor.

Consideramos importante también realizar el análisis de los errores en distintas formas que se presentan en los valores predichos. Se puede observar en la Figura 4.9 la distribución del porcentaje del error pico, en intervalos horarios, a lo largo del conjunto de datos. El error máximo se calcula encontrando el error absoluto máximo para cada período de 24 horas, y luego se normaliza al dividir por el valor máximo de entre los valores reales para ese día en particular. Los valores menores a 0 indican que los resultados de la predicción fueron menores en comparación a los valores reales, mientras que los mayores a 0 indican justamente lo contrario.

La distribución del porcentaje de error se centra alrededor de cero sugiere que en una gran mayoría de los días el error pico fue muy pequeño porcentualmente entre los valores predichos y los reales, además de que el rango en el cual caen los errores es relativamente pequeño, indicando un rendimiento constante del modelo a lo largo de los días en el conjunto de datos.

El histograma de la Figura 4.10 indica una distribución de los errores máximos que se alcanzan durante los días, es decir, el peor caso en cuanto a error para

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

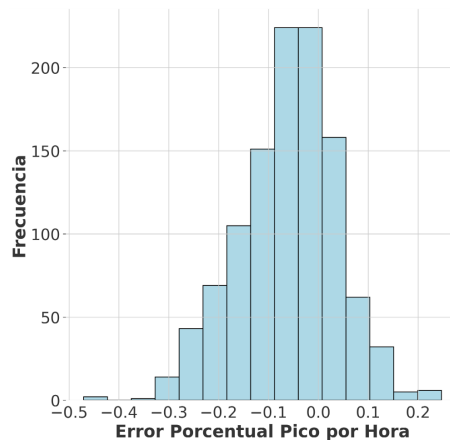


Figura 4.9: Histograma del error pico porcentual por hora.

Fuente: Elaboración propia.

cada día evaluado dentro del conjunto de datos. Se observa que está sesgado a la derecha, indicando que hay muchos más días con errores bajos porcentuales, cuyo rango más poblado es entre 0,1 y 0,2.

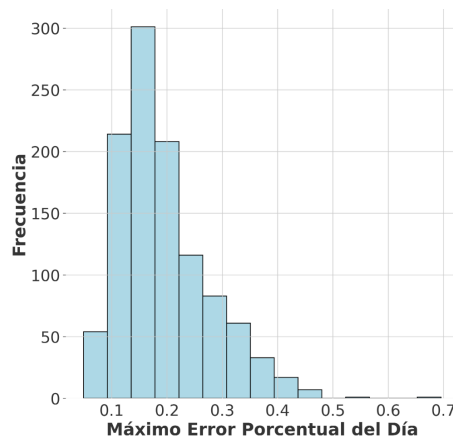


Figura 4.10: Histograma del máximo error porcentual del día

Fuente: Elaboración propia.

4.5. Comparativa de resultados

Es posible observar el comportamiento del modelo en los considerados mejores días según la métrica del MSE según la Figura 4.11. Vemos como el modelo sigue la tendencia según los picos y valles para cada hora en todos los días

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

mostrados, buscando adaptarse a cada cambio brusco que se presenta.

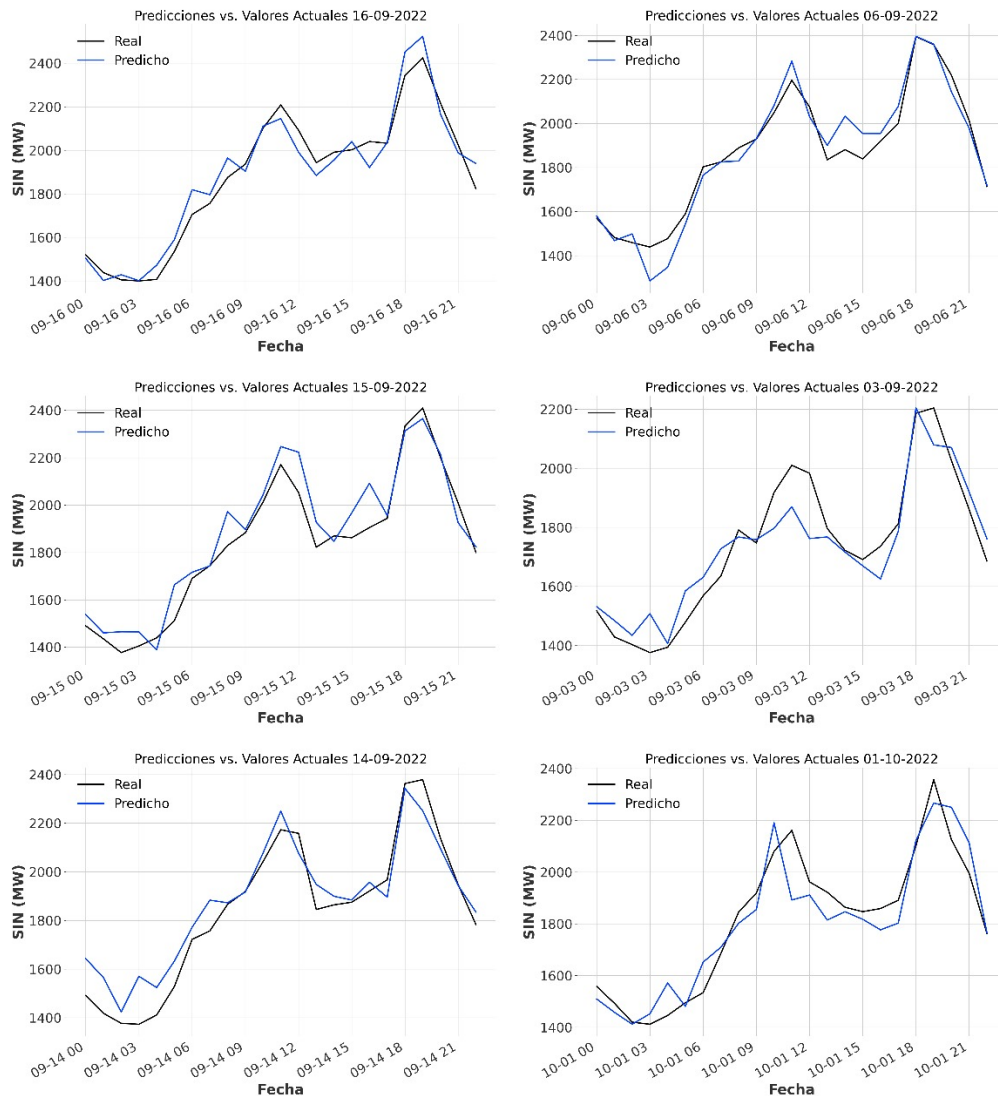


Figura 4.11: Comparativa en los mejores días de los valores del mejor modelo con los valores reales.

Fuente: Elaboración propia.

Además de observar el comportamiento en cada hora por día, podemos observar el desempeño en la mejor y peor semana del modelo según las métricas en las Figuras 4.12 y 4.13 respectivamente.

En la mejor semana (12-09-22 al 19-09-22) la curva de los valores predichos sigue bastante cerca a la de los valores reales en el conjunto de datos de prueba, indicando una gran precisión en este período con desviaciones muy pequeñas.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

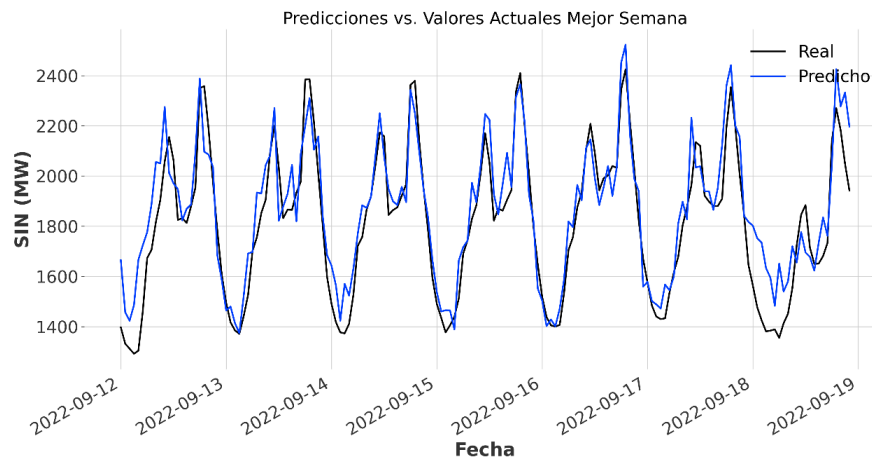


Figura 4.12: Comparativa en la mejor semana de los valores del mejor modelo con los valores reales.

Fuente: Elaboración propia.

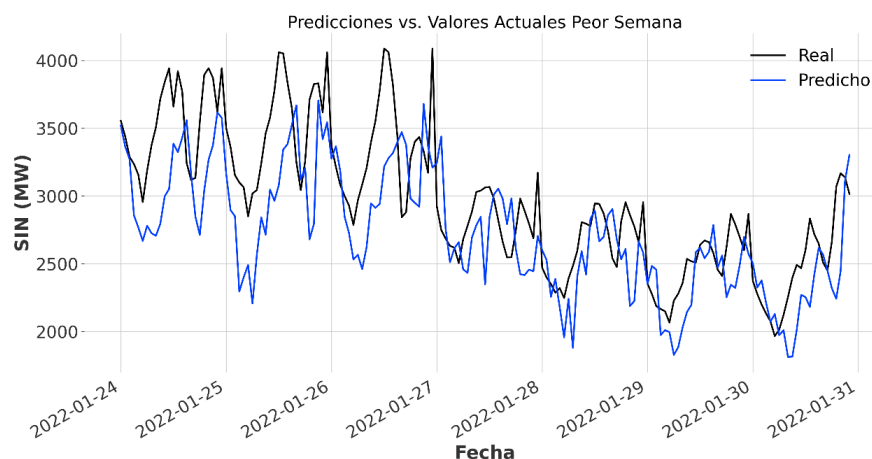


Figura 4.13: Comparativa en la peor semana de los valores del mejor modelo con los valores reales.

Fuente: Elaboración propia.

En el caso de la peor semana (24-01-2022 al 31-01-2022) existen discrepancias mucho más notables debido a los cambios bruscos en los valores reales, que no siguen un patrón muy claro ni constante, destacando que esta semana corresponde al mes de Enero, donde es más probable que se presente este tipo de comportamientos.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

CAPÍTULO 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

En el desarrollo de este Trabajo Final de Grado, se implementaron las distintas fases de preprocesamiento de datos. Esto incluyó la escalabilidad, normalización, transformación y partición adecuada del conjunto de datos suministrado, permitiendo su adaptación a los requisitos de la arquitectura y el modelo seleccionados.

Los datos históricos de la ANDE fueron actualizados con la información más reciente proveniente de fuentes oficiales, para el periodo 2007-2022.

Se implementó con éxito un modelo basado en la arquitectura TFT, que se distingue por la integración de varios conceptos y arquitecturas internas, para predecir la demanda de energía a nivel nacional. Este modelo hizo uso de herramientas especializadas como la biblioteca Darts, orientada al entrenamiento de modelos de series temporales.

A través de iteraciones y pruebas con diferentes configuraciones y valores, se logró optimizar los parámetros relevantes dentro de la arquitectura TFT, se agregaron nuevas variables (como el feriado), mejorando la precisión de los modelos resultantes.

Las comparaciones entre los valores predichos por el modelo y los valores reales del Sistema Interconectado Nacional (SIN) revelaron niveles aceptables de precisión, especialmente en periodos desafiantes como los picos y variaciones

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

observados durante el verano. Asimismo, se destacó la capacidad del modelo propuesto en la adaptación al conjunto de datos de prueba.

Estos hallazgos consolidan la efectividad del modelo propuesto para predecir la demanda energética, sin embargo, su aplicabilidad y generalización requerirían un análisis más exhaustivo en contextos operativos y de planificación energética a nivel nacional.

5.2. Trabajos Futuros

Futuras investigaciones deben centrarse en evaluar nuevos modelos derivados del presentado en este trabajo mediante técnicas de transferencia de aprendizaje y ajuste fino [24]. Estos modelos deben ser adaptados para los nuevos datos de demanda eléctrica y condiciones climáticas que puedan surgir y que no estén contemplados en el conjunto de datos utilizado en este estudio.

El progreso hacia la obtención de predicciones más precisas y una comprensión más profunda de cómo las variables utilizadas para el entrenamiento afectan dichos modelos es crucial. Por lo tanto, investigaciones futuras deberían considerar la implementación de nuevas arquitecturas, como Crossformer [25] y N-Beats [26].

Este trabajo proporcionó una solución para la predicción de la demanda a nivel nacional. Sin embargo, estudios futuros podrían abordar este problema de manera más detallada, reutilizando la implementación existente pero centrándose en la predicción a nivel de subsistema.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

BIBLIOGRAFÍA

- [1] Administración Nacional de Electricidad, *Plan Maestro de Transmisión*, Febrero 2021, período: 2021 – 2030.
- [2] C. Kuster, Y. Rezgui, y M. Mourshed, “Electrical load forecasting models: A critical systematic review,” *Sustainable Cities and Society*, 2017. [En línea]. Disponible: <https://doi.org/10.1016/j.scs.2017.10.017>
- [3] O. A. Barboza, “Automatización de previsión de demanda horaria de potencia eléctrica en el sistema interconectado nacional,” *Revista Científica de la UCSA*, pp. <https://www.mdpi.com/1996-1073/11/7/1636>, 2014.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, y L. Kaiser, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017. [En línea]. Disponible: <https://arxiv.org/abs/1706.03762>
- [5] B. Lima, S. O. Arik, N. Loeff, y T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *Elsevier*, p. arXiv:1912.09363, 2019.
- [6] G. Parra y J. Verdún, “Metodología para estimación de demanda eléctrica a corto plazo de una subestación utilizando redes neuronales artificiales,” Trabajo Final de Grado, 2015.
- [7] M. Verdún y F. Morales, “Diseño de un modelo inteligente para la predicción horaria a corto plazo de la demanda eléctrica nacional basado en aprendizaje profundo y propuesta de implementación en el despacho de carga de la ande,” Trabajo Final de Grado, Facultad de Ingeniería - UNA, 2020.
- [8] C. D. B. I. A. A. Benitez y R. Rodriguez, “Aplicación de redes neuronales artificiales en la previsión del suministro energético de la central hidroeléctrica de itaipu al sistema eléctrico paraguayo,” Trabajo Final de Grado, 2014.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

- [9] Administración Nacional de Electricidad (ANDE), “Memoria anual 2022,” 2023.
- [10] —, “Mercado eléctrico nacional proyección 2019-2030: Informe ejecutivo,” 2019.
- [11] Mauricio, *Análisis de Series Temporales*, 1st ed. Universidad Complutense de Madrid, 2007.
- [12] B. Lim y S. Zohren, “Time series forecasting with deep learning: A survey,” *Department of Engineering Science, University of Oxford, Oxford, UK*, p. arXiv:2004.13408, 2020.
- [13] Y. Bengio, P. Simard, y P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [14] J. Kolen y S. Kremer, “Gradient flow in recurrent nets: The difficulty of learning long-term dependencies,” pp. 237–243, 2001.
- [15] I. Goodfellow, Y. Bengio, y A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [16] S. Hochreiter y J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, November 1997.
- [17] R. G. Steadman, “A universal scale of apparent temperature,” *Journal of Climate and Applied Meteorology*, vol. 23, no. 12, pp. 1674–1687, 1984.
- [18] Escala de beaufort. [En línea]. Disponible: <https://es.wikipedia.org/wiki/Escala-de-Beaufort> Accedido el XX de mes de XXXX.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

- [19] C. Fan, M. Chen, X. Wang, J. Wang, y B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Frontiers in Energy Research*, vol. 9, p. 652801, 2021, this article is part of the Research Topic "Future Energy Systems for Building Clusters and Districts". [En línea]. Disponible: <https://doi.org/10.3389/fenrg.2021.652801>
- [20] F. Ziel, "Modeling public holidays in load forecasting: a german case study," *Journal of Modern Power Systems and Clean Energy*, vol. 6, pp. 191–207, 2018, special Section on Forecasting in Modern Power Systems, Open access.
- [21] Y. Cheng, C. Xu, D. Mashima, V. L. Thing, y Y. Wu, "Powerlstm: Power demand forecasting using long short-term memory neural network," pp. 727–740, 2017.
- [22] J. Herzen, S. G. Piazzetta, T. Neuer, G. , y T. Van Pottelbergh, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, 2022.
- [23] T. Akiba, S. Sano, T. Yanase, T. Ohta, y M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," *arXiv preprint arXiv:1907.10902*, 2019.
- [24] Z. Al-Ali. Production forecasting using transfer learning of pretrained deep model. [En línea]. Disponible: <https://suetrid.stanford.edu/production-forecasting-using-transfer-learning-pretrained-deep-model> (2022)
- [25] Y. Zhang y J. Yan, "Crossformer: Transformer utilizing crossdimension dependency for multivariate time series forecasting," 2023, published as a conference paper at ICLR 2023.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

- [26] B. N. Oreshkin, D. Carpov, N. Chapados, y Y. Bengio, “N-beats: Neural basis expansion analysis for interpretable time series forecasting,” in *International Conference on Learning Representations (ICLR)*. Element AI and Mila, 2020, published as a conference paper at ICLR 2020.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Anexo A: Definiciones complementaria

Grid-Search y Random-Search

Un procedimiento de optimización implica la definición de un espacio de búsqueda. Éste puede pensarse geométricamente como un volumen de n dimensiones, donde cada hiperparámetro representa una dimensión diferente y la escala de la dimensión son los valores que puede asumir el hiperparámetro, como valor real, valor entero o categórico.

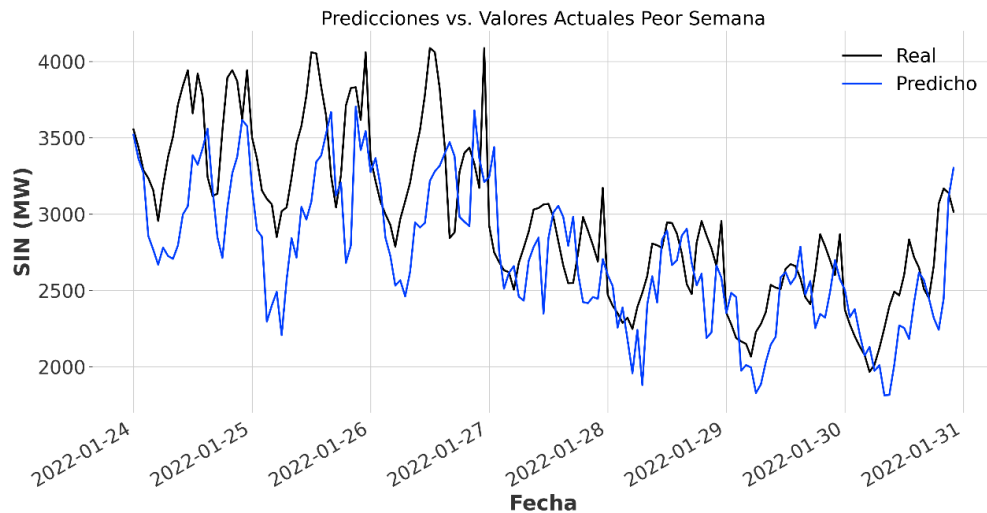
- **Espacio de búsqueda:** Volumen a buscar donde cada dimensión representa un hiperparámetro y cada punto representa una configuración de modelo.

Un punto en el espacio de búsqueda es un vector con un valor específico para cada valor de hiperparámetro. El objetivo del procedimiento de optimización es encontrar un vector que dé como resultado el mejor rendimiento del modelo después de su aprendizaje, como una máxima precisión o un mínimo error. Se puede utilizar una gama de diferentes algoritmos de optimización, aunque dos de los métodos más simples y comunes son la búsqueda aleatoria (Random Search) y la búsqueda en cuadrículas (Grid Search).

- **Búsqueda aleatoria:** Definir un espacio de búsqueda como un dominio delimitado de valores de hiperparámetros y puntos de muestra aleatorios en ese dominio.
- **Búsqueda en la red:** Definir un espacio de búsqueda como una cuadrícula de valores hiperparamétricos y evaluar cada posición en la cuadrícula.

La búsqueda de cuadrículas es genial para comprobar combinaciones que se sabe que funcionan bien en general. La búsqueda aleatoria es genial para descubrir y obtener combinaciones de hiperparámetros que no habrías adivinado intuitivamente, aunque a menudo requiere más tiempo para ejecutarse.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL



Representación gráfica de búsqueda en la red y búsqueda aleatoria

Fuente: Elaboración propia.

Métricas de Evaluación de Rendimiento

La elección de una métrica de evaluación del rendimiento a menudo depende del dominio del problema. Las principales métricas de evaluación de predicciones que se pueden usar para crear y probar modelos predictivos de valores numéricos son:

Error medio (ME – Mean Error)

El error medio (ME) es el promedio de todos los errores de un conjunto de observaciones.

Esta es una métrica muy simple. Desafortunadamente está sesgada debido al efecto de compensación de errores de predicción positivos y negativos, que pueden ocultar la imprecisión de la predicción para observaciones concretas. Debido a eso, el ME no es muy útil para la evaluación del modelo. Sin embargo, es muy fácil de entender (esto no siempre es una ventaja debido a las limitaciones descritas). El ME puede mostrar rápidamente la simetría de la distribución

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

de errores, lo que puede ser útil para evaluar un modelo específico.

Error absoluto medio (MAE – Mean Absolute Error)

La solución más sencilla para la inexactitud del error medio es el uso del error absoluto medio (MAE).

El MAE utiliza los valores absolutos de los errores en los cálculos, lo que arregla el problema de la cancelación de errores con signos opuestos. Nos da un promedio de la magnitud absoluta de todos los valores de los errores, sin importar si eran positivos o negativos.

Error cuadrático medio (MSE – Mean Square Error)

Al igual que el MAE, el error cuadrático medio (MSE) también arregla el problema de la cancelación de errores positivos y negativos. Sin embargo, otorga una mayor penalización en los errores de predicción grandes que el MAE.

La raíz del error cuadrático medio

La raíz del error cuadrático medio (RMSE) es lo que en estadística se conoce como desviación estándar de los errores.

El RMSE comparte las ventajas del MSE y se utiliza comúnmente en la predicción y análisis de las regresiones para verificar resultados experimentales. Además, tiene la ventaja de tener las mismas unidad

Error porcentual medio (MPE – Mean Percentage Error)

El error porcentual medio (MPE) es el promedio de errores porcentuales por los que cada previsión difiere de sus correspondientes valores reales observados. Esta métrica es fácil de entender porque proporciona el error en términos de porcentajes. Al igual que en el ME, los errores de previsión positivos y negativos

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

pueden compensarse entre sí, por lo que se puede utilizar para medir el sesgo en las previsiones. La desventaja de esta estadística es que no es adecuada para conjuntos de datos que contienen valores observados iguales a cero (pues estos están en el denominador de todos los elementos del sumatorio y un valor nulo implicaría un MPE indefinido arbitrariamente grande).

Error porcentual absoluto medio (MAPE – Mean Absolute Percentage Error)

El error porcentual absoluto medio (MAPE) arregla el problema con la compensación de errores (tal y como lo hacía el MAE) y funciona mejor si no hay extremos en los datos (y no hay ceros).

Métricas o coeficientes de precisión U de Theil

Hay cierta confusión sobre el coeficiente de precisión de Theil, causado probablemente por el propio Theil, pues propuso dos fórmulas diferentes en diferentes momentos con el mismo nombre. En el primer coeficiente U1 los valores están en el rango (0, 1).

Cuanto mayor sea la precisión de la predicción, menor será el valor del coeficiente. En el segundo coeficiente U2 se indica cuánto más (o menos) preciso es un modelo en relación con una predicción trivial.

Al igual que U1, U2 tiene un límite inferior de 0 (que indica una predicción perfecta), sin embargo no tiene un límite superior. Cuando su valor supera 1, significa que la predicción es incluso peor que la predicción trivial.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Anexo B: Documentación de la API DARTS

```
class darts.models.forecasting.tft_model.TFTModel (input_chunk_length,  
  
    output_chunk_length, hidden_size=16, lstm_layers=1,  
  
    num_attention_heads=4, full_attention=False,  
  
    feed_forward='GatedResidualNetwork', dropout=0.1,  
  
    hidden_continuous_size=8,  
  
    categorical_embedding_sizes=None,  
  
    add_relative_index=False, loss_fn=None, likelihood=None,  
  
    norm_type='LayerNorm', use_static_covariates=True, **kwargs)
```

Parámetros:

input_chunk_length (int): Número de pasos de tiempo en el pasado que se toman como entrada del modelo (por fragmento). Se aplica a la serie objetivo, y a covariables pasadas y/o futuras (si el modelo lo soporta). También llamado: longitud del codificador.

output_chunk_length (int): Número de pasos de tiempo que se predicen de una vez (por fragmento) por el modelo interno. También es el número de valores futuros de covariables futuras que se usan como entrada del modelo (si el modelo soporta covariables futuras). No es lo mismo que el horizonte de pronóstico n utilizado en `predict()`, que es el número deseado de puntos de predicción generados usando un pronóstico de una sola vez o autoregresivo. Establecer $n \leq output_chunk_length$ previene la autoregresión. Esto es útil cuando las covariables no se extienden lo suficientemente hacia el futuro, o para prohibir que el modelo use valores futuros de covariables pasadas y/o

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

futuras para la predicción (dependiendo del soporte de covariables del modelo). También llamado: longitud del decodificador.

hidden_size (Union[int, List[int]]): Tamaño del estado oculto del TFT. Es el hiperparámetro principal y común en toda la arquitectura interna del TFT.

lstm_layers (int): Número de capas para el codificador y decodificador de Memoria a Corto y Largo Plazo (LSTM) (1 es un buen valor por defecto).

num_attention_heads (int): Número de cabezas de atención (4 es un buen valor por defecto).

full_attention (bool): Si es Falso, solo atiende a pasos de tiempo anteriores en el decodificador. Si es Verdadero, atiende a pasos de tiempo anteriores, actuales y futuros. Por defecto es Falso.

feed_forward (str): Una red feedforward es una capa completamente conectada con una activación. TFT puede ser una de las variantes de la Red de FeedForward (FFN)[2]. Las variantes de la Red de FeedForward son una serie de FFNs diseñadas para funcionar mejor con modelos basados en Transformer. Por defecto es "GatedResidualNetwork". ["GLU", "Bilinear", "ReLU", "GEGLU", "SwiGLU", "ReLU", "GELU"] o la Red de FeedForward original de TFT ["GatedResidualNetwork"].

dropout (float): Fracción de neuronas afectadas por el dropout. Esto es compatible con el dropout de Monte Carlo en el tiempo de inferencia para la estimación de incertidumbre del modelo (habilitado con `mc_dropout=True` en el tiempo de predicción).

hidden_continuous_size (int): Por defecto para el tamaño oculto al procesar variables continuas.

categorical_embedding_sizes (Optional[Dict[str, Union[int, Tuple[int,

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

int]]]]): Un diccionario utilizado para construir incrustaciones para co-variables estáticas categóricas. Las claves son los nombres de las columnas de las covariables estáticas categóricas. Cada valor es un solo entero o un par de enteros. Para un solo entero, da el número de categorías únicas (n) de la variable correspondiente. Por ejemplo, `{.alguna_columna": 64}`. El tamaño de la incrustación se determinará automáticamente por $\min(\text{round}(1,6 \times n^{0,56}), 100)$. Para un par de enteros, da $(\text{número de categorías únicas}, \text{tamaño de la incrustación})$. Por ejemplo, `{.alguna_columna": (64, 8)}`. Note que los `TorchForecastingModels` solo soportan datos numéricos. Considere transformar/codificar sus datos con transformers de `darts.dataprocessing` como

`StaticCovariatesTransformer` para covariables estáticas.

add_relative_index (bool): Si se debe agregar valores posicionales a co-variables futuras. Por defecto es Falso. Esto permite usar el modelo TFT sin tener que pasar covariables futuras a `fit()` y `train()`. Da un valor a la posición de cada paso de entrada y salida del fragmento relativo al punto de predicción. Los valores se normalizan con `input_chunk_length`.

loss_fn (nn.Module): Función de pérdida de PyTorch utilizada para el entrenamiento. Por defecto, el modelo TFT es probabilístico y usa una probabilidad en su lugar (`QuantileRegression`). Para hacer que el modelo sea determinista, puede establecer el `likelihood` a `None` y proporcionar un argumento `loss_fn`.

likelihood (Optional[Likelihood]): El modelo de probabilidad que se utilizará para pronósticos probabilísticos. Por defecto, el TFT utiliza una probabilidad de `QuantileRegression`

norm_type (str | nn.Module): El tipo de variante de `LayerNorm` a usar.

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

Por defecto: `LayerNorm`. Las opciones disponibles son `["LayerNorm", "RMSNorm", "LayerNormNoBias"]`, o proporcione un `nn.Module` personalizado.

`use_static_covariates (bool)`: Si el modelo debe usar información de covariables estáticas en caso de que las series de entrada pasadas a `fit()` contengan covariables estáticas. Si es Verdadero, y las covariables estáticas están disponibles en el momento del ajuste, se exigirá que todas las series objetivo tengan la misma dimensionalidad de covariables estáticas en `fit()` y `predict()`.

`kwargs`:** Argumentos opcionales para inicializar

`pytorch_lightning.Module`,
`pytorch_lightning.Trainer` y el
`TorchForecastingModel` de Darts.

`torch_metrics`: Una métrica de torch o una `MetricCollection` utilizada para la evaluación. Una lista completa de métricas disponibles se puede encontrar en <https://torchmetrics.readthedocs.io/en/latest/>. Por defecto: Ninguno.

`optimizer_cls`: La clase de optimizador de PyTorch que se usará. Por defecto: `torch.optim.Adam`.

`optimizer_kwargs`: Opcionalmente, algunos argumentos de palabras clave para el optimizador de PyTorch (por ejemplo, `{'lr': 1e-3}` para especificar una tasa de aprendizaje). De lo contrario, se utilizarán los valores predeterminados del `optimizer_cls` seleccionado. Por defecto: Ninguno.

`lr_scheduler_cls`: Opcionalmente, la clase de programador de tasa de aprendizaje de PyTorch que se usará. Especificar `None` corresponde a usar una tasa de aprendizaje constante. Por defecto: Ninguno.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

lr_scheduler_kwargs: Opcionalmente, algunos argumentos de palabras clave para el programador de tasa de aprendizaje de PyTorch. Por defecto: Ninguno.

use_reversible_instance_norm: Si se debe usar la normalización de instancia reversible `RINorm` contra el desplazamiento de distribución como se muestra en [3]. Solo se aplica a las características de la serie objetivo y no a las covariables.

batch_size: Número de series de tiempo (secuencias de entrada y salida) utilizadas en cada paso de entrenamiento. Por defecto: 32.

n_epochs: Número de épocas sobre las cuales entrenar el modelo. Por defecto: 100.

norm_type (str | nn.Module): El tipo de variante de LayerNorm a usar. Por defecto: LayerNorm. Las opciones disponibles son ["LayerNorm", "RMSNorm", "LayerNormNoBias"], o proporcione un nn.Module personalizado.

use_static_covariates (bool): Si el modelo debe usar información de covariables estáticas en caso de que las series de entrada pasadas a `fit()` contengan covariables estáticas. Si es Verdadero, y las covariables estáticas están disponibles en el momento del ajuste, se exigirá que todas las series objetivo tengan la misma dimensionalidad de covariables estáticas en `fit()` y `predict()`.

****kwargs:** Argumentos opcionales para inicializar `pytorch_lightning.Module`, `pytorch_lightning.Trainer` y el `TorchForecastingModel` de Darts.

torch_metrics: Una métrica de torch o una `MetricCollection` utilizada para la evaluación. Una lista completa de métricas disponibles se puede encontrar en <https://torchmetrics.readthedocs.io/en/latest/>. Por defecto: Ninguno.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

optimizer_cls: La clase de optimizador de PyTorch que se usará. Por defecto: torch.optim.Adam.

optimizer_kwargs: Opcionalmente, algunos argumentos de palabras clave para el optimizador de PyTorch (por ejemplo, 'lr': 1e-3 para especificar una tasa de aprendizaje). De lo contrario, se utilizarán los valores predeterminados del optimizer_cls seleccionado. Por defecto: Ninguno.

lr_scheduler_cls: Opcionalmente, la clase de programador de tasa de aprendizaje de PyTorch que se usará. Especificar None corresponde a usar una tasa de aprendizaje constante. Por defecto: Ninguno.

lr_scheduler_kwargs: Opcionalmente, algunos argumentos de palabras clave para el programador de tasa de aprendizaje de PyTorch. Por defecto: Ninguno.

use_reversible_instance_norm: Si se debe usar la normalización de instancia reversible RINorm contra el desplazamiento de distribución como se muestra en [3]. Solo se aplica a las características de la serie objetivo y no a las covariables.

batch_size: Número de series de tiempo (secuencias de entrada y salida) utilizadas en cada paso de entrenamiento. Por defecto: 32.

n_epochs: Número de épocas sobre las cuales entrenar el modelo. Por defecto: 100.

model_name: Nombre del modelo. Se utiliza para crear puntos de control y guardar datos de tensorboard. Si no se especifica, por defecto es la siguiente cadena

"YYYY-mm-dd_HH_MM_SS_torch_model_run_PID", donde la parte inicial del nombre está formateada con la fecha y hora local, mientras que PID es el ID del proceso (evitando que modelos iniciados al mismo

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

tiempo por diferentes procesos compartan el mismo `model_name`).

Por ejemplo, "2021-06-14_09_53_32_torch_model_run_44607".

work_dir: Ruta del directorio de trabajo, donde guardar los puntos de control y resúmenes de Tensorboard. Por defecto: el directorio de trabajo actual.

log_tensorboard: Si se establece, usar Tensorboard para registrar los diferentes parámetros. Los registros se ubicarán en:

`_dir@"work_dir/darts_logs/model_name/logs/"`. Por defecto: Falso.

nr_epochs_val_period: Número de épocas a esperar antes de evaluar la pérdida de validación (si se pasa una `TimeSeries` de validación al método `fit()`). Por defecto: 1.

force_reset: Si se establece en Verdadero, cualquier modelo previamente existente con el mismo nombre será reiniciado (todos los puntos de control serán descartados). Por defecto: Falso.

save_checkpoints: Si se deben guardar automáticamente el modelo sin entrenar y los puntos de control del entrenamiento. Para cargar el modelo desde el punto de control, llame a `MyModelClass.load_from_checkpoint()`, donde `MyModelClass` es la clase `TorchForecastingModel` que se utilizó (como `TFTModel`, `NBEATS-Model`, etc.). Si se establece en Falso, el modelo aún se puede guardar manualmente usando `save()` y cargado usando `load()`. Por defecto: Falso.

add_encoders: Un gran número de covariables pasadas y futuras pueden generarse automáticamente con `add_encoders`. Esto se puede hacer agregando múltiples codificadores de índice predefinidos y/o funciones hechas por el usuario que se usarán como codificadores de índice. Además, se puede agregar un transformador como el `Scaler` de

DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO DE LA DEMANDA ELÉCTRICA NACIONAL

Darts para transformar las covariables generadas. Esto sucede todo bajo un mismo techo y solo necesita especificarse en la creación del modelo. Lea `SequentialEncoder` para obtener más información sobre `add_encoders`. Por defecto: Ninguno.

random_state: Controla la aleatoriedad de la inicialización de los pesos. Consulte este enlace para más detalles. Por defecto: Ninguno.

pl_trainer_kwargs: Por defecto, `TorchForecastingModel` crea un Entrenador de PyTorch Lightning con varios ajustes preestablecidos útiles que realizan los procesos de entrenamiento, validación y predicción. Estos ajustes incluyen el checkpointing automático, el registro en tensorboard, la configuración del dispositivo torch y más. Con `pl_trainer_kwargs`, puedes agregar argumentos adicionales para instanciar el objeto del entrenador de PyTorch Lightning. Consulte la documentación del Entrenador de PL para obtener más información sobre los kwargs admitidos. Por defecto: Ninguno. También es posible ejecutar en GPU(s) utilizando `pl_trainer_kwargs` especificando las claves `"accelerator"`, `"devices"` y `"auto_select_gpus"`.

Metodos

- **backtest(series[, past_covariates, ...]):** Calcula los valores de error que el modelo habría producido cuando se utiliza en (potencialmente múltiples) series.
- **fit(series[, past_covariates, ...]):** Ajusta/entrena el modelo en una o múltiples series.
- **fit_from_dataset(train_dataset[, ...]):** Entrena el modelo con una instancia específica de `darts.utils.data.TrainingDataset`.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

- **generate_fit_encodings(series[, ...]):** Genera las codificaciones de covariables que se utilizaron/generaron para ajustar el modelo y devuelve una tupla de series de covariables pasadas y futuras con las covariables originales y codificadas juntas.
- **generate_fit_predict_encodings(n, series[, ...]):** Genera codificaciones de covariables para el entrenamiento y la inferencia/predicción y devuelve una tupla de series de covariables pasadas y futuras con las covariables originales y codificadas juntas.
- **generate_predict_encodings(n, series[, ...]):** Genera codificaciones de covariables para el conjunto de inferencia/predicción y devuelve una tupla de series de covariables pasadas y futuras con las covariables originales y codificadas juntas.
- **gridsearch(parameters, series[, ...]):** Encuentra los mejores hiperparámetros entre un conjunto dado utilizando una búsqueda en cuadrícula.
- **historical_forecasts(series[, ...]):** Calcula las predicciones históricas que habrían sido obtenidas por este modelo en (potencialmente múltiples) series.
- **load(path, **kwargs):** Carga un modelo desde una ruta de archivo dada.
- **load_from_checkpoint(model_name[, work_dir, ...]):** Carga el modelo desde puntos de control guardados automáticamente bajo
`work_dir\darts_logs/{model_name}/checkpoints/`.
- **load_weights(path[, load_encoders, skip_checks]):** Carga los pesos de un modelo guardado manualmente (guardado con `save()`).
- **load_weights_from_checkpoint([model_name, ...]):** Carga solo los pesos desde puntos de control guardados automáticamente bajo

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

`work_dirdarts_logs/{model_name}/checkpoints/`.

- **lr_find(series[, past_covariates, ...]):** Un envoltorio alrededor de `Tuner.lr_find()` de PyTorch Lightning.
- **predict(n, *args, **kwargs):** Predice los n pasos de tiempo siguientes al final de la serie de entrenamiento, o de la serie especificada.
- **predict_from_dataset(n, input_series_dataset):** Este método permite predecir con una instancia específica de `darts.utils.data.InferenceDataset`.
- **reset_model():** Reinicia el objeto del modelo y elimina todos los datos almacenados: modelo, puntos de control, registradores e historial de entrenamiento.
- **residuals(series[, past_covariates, ...]):** Calcula los residuos producidos por este modelo en una (o secuencia de) serie de tiempo univariada.
- **save([path]):** Guarda el modelo bajo una ruta dada.
- **to_cpu():** Actualiza los parámetros del Entrenador de PyTorch Lightning para mover el modelo a la CPU la próxima vez que se llame a `fit()` o `predict()`.

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

Anexo C: Código ejemplo para entrenamiento TFT en lenguaje

Python

```
import torch

import numpy as np

import pandas as pd

import shutil

from sklearn.preprocessing import MinMaxScaler

import matplotlib.pyplot as plt

from darts import TimeSeries

from darts.dataprocessing.transformers import Scaler

from darts.models import TFTModel

from darts.metrics import mape

from darts.utils.timeseries_generation

import datetime_attribute_timeseries

from pytorch_lightning.callbacks import EarlyStopping

from torchmetrics import MeanAbsolutePercentageError

from sklearn.metrics import r2_score,

mean_absolute_error, mean_squared_error,

mean_absolute_percentage_error, max_error

import warnings

warnings.filterwarnings("ignore")

import logging

logging.disable(logging.CRITICAL)


file_path = 'processed_datasetNew.csv'

data = pd.read_csv(file_path)

data = data.set_index('Date')
```

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

```
data.index = pd.to_datetime(data.index)

split_date = pd.to_datetime('2008-12-31 23:59')

data = data[data.index >= split_date]


data["Month_sin"] = np.sin(2 * np.pi * data.index.month / 12)
data["Month_cos"] = np.cos(2 * np.pi * data.index.month / 12)
data["Year"] = data.index.year
data["weekday_cos"] = np.sin(2 * np.pi * (data.index.weekday+1) / 7)
data["weekday_sin"] = np.cos(2 * np.pi * (data.index.weekday+1) / 7)


features_options = [
    ["Temperature", "Humidity", "Year", "weekday_cos",
    "weekday_sin", "Month_cos", "Month_sin"],
]

data_sin = data[['SIN']]
data_cov = data[features_options[0]]


series = TimeSeries.from_dataframe(data_sin, value_cols=["SIN"])
covariates = TimeSeries.from_dataframe(data_cov,
value_cols=features_options[8])
series = series.astype(np.float32)
covariates = covariates.astype(np.float32)
covariates = covariates.add_holidays('PY')


training_cutoff = pd.Timestamp("20191231T230000")
train, val = series.split_after(training_cutoff)
transformer = Scaler()
train_transformed = transformer.fit_transform(train)
```

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

```
val_transformed = transformer.transform(val)

series_transformed = transformer.transform(series)

covariates = covariates.stack(
    TimeSeries.from_times_and_values(
        times=series.time_index,
        values=np.arange(len(series)),
        columns=["linear_increase"],
    )
)

covariates = covariates.astype(np.float32)
cov_train, cov_val = covariates.split_after(training_cutoff)
scaler_covs = Scaler()
covariates_transformed_train = scaler_covs.fit_transform(cov_train)
covariates_transformed_val = scaler_covs.transform(cov_val)
covariates_transformed = scaler_covs.transform(covariates)

torch_metrics = MeanAbsolutePercentageError()
torch.manual_seed(42)

my_stopper = EarlyStopping(
    monitor="val_MeanAbsolutePercentageError", # "train_loss"
    patience=2,
    min_delta=0.0001,
    mode='min',
)

pl_trainer_kwargs = {
    "accelerator": "gpu",
```

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

```
    "devices": [0],  
    "callbacks": [my_stopper]  
}
```

```
tft = TFTModel(  
    input_chunk_length=19*24,  
    output_chunk_length=1*24,  
    hidden_size=46,  
    lstm_layers=1,  
    num_attention_heads=3,  
    dropout=0.3505077873908772,  
    batch_size=24,  
    n_epochs=6,  
    add_relative_index=False,  
    add_encoders=None,  
    # loss_fn=MSELoss(),  
    random_state=42,  
    force_reset=True,  
    save_checkpoints=True,  
    torch_metrics=torch_metrics,  
    model_name="tft_best_model_07",  
    pl_trainer_kwargs=pl_trainer_kwargs,  
)
```

```
tft.fit(series=train_transformed, val_series=val_transformed,  
future_covariates=covariates_transformed,  
val_future_covariates=covariates_transformed_val,  
num_loader_workers=2, verbose=True)
```

**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

```
tft = TFTModel.load_from_checkpoint("tft_best_model_07")

tft_model_path = "tft_best_model_07.pt"

tft.save(tft_model_path)


n = len(val)

tft_pred_val = tft.predict(n=n,
future_covariates=covariates_transformed)

tft_pred_val = transformer.inverse_transform(tft_pred_val)


validation_range = ('2020-01-01', '2022-12-31')
dtin = pd.date_range(start=validation_range[0],
end=validation_range[1], freq='H')
tft_pred_val_arr = tft_pred_val.values()
val_arr = val.values()


ixmax = np.argmax(val_arr.reshape(-1,24), axis=1)
hmaxerr = ((tft_pred_val_arr.reshape(-1,24)[
np.arange(tft_pred_val_arr.reshape(-1,24).shape[0]),ixmax]
-np.max(val_arr.reshape(-1,24),axis=1))
/np.max(val_arr.reshape(-1,24),axis=1))
hsup = np.percentile(hmaxerr,99)
hinf = np.percentile(hmaxerr,1)
mse = mean_squared_error( val_arr, tft_pred_val_arr)
rmse = np.sqrt(mse)
error = np.abs(val_arr-tft_pred_val_arr)
Error_95_ = np.percentile(error,95)
maxError_ = max_error(val_arr, tft_pred_val_arr)
```


**DISEÑO DE UN MODELO DE DEEP LEARNING BASADO EN LA ARQUITECTURA
TEMPORAL FUSION TRANSFORMER PARA LA PREDICCIÓN HORARIA A CORTO PLAZO
DE LA DEMANDA ELÉCTRICA NACIONAL**

```
r2_ = r2_score(val_arr, tft_pred_val_arr)

print( "MSE: " + str(mse)+" RMSE:" + str(rmse  ))

print( "MaxError: " + str(maxError_)+" R2:" + str(r2_  )
+" Error95:" + str(Error_95_))

print( "hsup: " + str(hsup)+" hinf:" + str(hinf))

print( "hmaxerr5: " + str(np.percentile(error,5)))

print( "hmaxerr50: " + str(np.percentile(error,50)))

print( "hmaxerr95: " + str(np.percentile(error,95)))
```