



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Despacho Óptimo de Reactiva con Aprendizaje Automático en la Red Eléctrica Uruguaya

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Ignacio Boero, Santiago Díaz y Tomás Vázquez

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRICISTA.

TUTOR

Pablo Belzarena Universidad de la Repùblica
Federico La Rocca Universidad de la Repùblica
Enzo Coppes Universidad de la Repùblica

TRIBUNAL

Germán Capdehourat Universidad de la Repùblica
Marcelo Fiori Universidad de la Repùblica
Nicolás Yedrzejewski Universidad de la Repùblica

Montevideo
lunes 14 octubre, 2024

Despacho Óptimo de Reactiva con Aprendizaje Automático en la Red Eléctrica Uruguaya, Ignacio Boero, Santiago Díaz y Tomás Vázquez.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).

Contiene un total de 165 páginas.

Compilada el lunes 14 octubre, 2024.

<http://iie.fing.edu.uy/>

- “Falta reentrenar SUP IEEE30”

IGNACIO BOERO

- “Es una materia más”

TOMÁS VÁZQUEZ

- “Ni que tuviéramos una familia que mantener”

SANTIAGO DIAZ

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

Queremos expresar nuestro agradecimiento a todas las personas que hicieron posible la realización de este proyecto. En primer lugar, agradecemos a nuestros tutores por parte de la Facultad de Ingeniería, Pablo Belzarena y Federico La Rocca, quienes nos acompañaron a lo largo de todo el proceso. Sus sugerencias, disposición para resolver nuestras dudas y orientación constante fueron fundamentales para mantenernos en el camino correcto.

Asimismo, extendemos nuestro agradecimiento a Enzo Coppes, Jefe del Despacho Nacional de Cargas, quien no solo fue nuestro principal vínculo con el DNC, sino que también nos brindó asesoramiento y formación sobre la operativa de la red. Gracias a su apoyo y a los datos que nos proporcionó, pudimos llevar adelante este proyecto. Queremos agradecer también a Nicolás Yedrzejewski, quien desde su rol en el DNC nos brindó apoyo técnico en temas de redes de potencia y nos proporcionó un modelo de la red eléctrica a partir del cual pudimos trabajar.

También agradecemos a Marcelo Fiori y Bernardo Marenco, quienes nos brindaron acceso a sus infraestructuras de cómputo, lo que nos permitió entrenar los modelos de aprendizaje automático.

De igual manera, agradecemos a Facundo Gutiérrez, quien realizó las correcciones de sintaxis y ortografía de este trabajo, contribuyendo a la claridad del documento.

Por último, queremos agradecer a nuestras familias por su incondicional apoyo durante más de un año de trabajo, por estar siempre presentes y acompañarnos en cada paso de este proceso.

Gracias a todos ustedes, este proyecto fue posible.

Esta página ha sido intencionalmente dejada en blanco.

Para todo aquel que se atreva a enfrentarse al problema de despacho óptimo de potencia.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

Este trabajo se centra en la resolución del problema del despacho óptimo de potencia reactiva en la red eléctrica de Uruguay utilizando aprendizaje automático. El despacho de potencia reactiva es crucial para la estabilidad y el control del sistema eléctrico nacional, y su objetivo principal es garantizar que los voltajes en las barras y las corrientes en las líneas se mantengan dentro de niveles seguros. Actualmente, la estrategia más común para lograr esto se basa en una variación del modelo matemático del flujo óptimo de potencia, el cual plantea un problema de optimización no convexo que se aborda mediante métodos de optimización clásicos. Sin embargo, a pesar de su amplia aceptación, existen diversas limitaciones en la forma en que se resuelve este problema en la actualidad. Esto se debe a la naturaleza no convexa del problema, precisando de algoritmos iterativos computacionalmente costosos y que frecuentemente resultan en soluciones subóptimas.

Por esta razón, se propone la implementación de dos estrategias de aprendizaje automático como alternativas para su resolución. La primera consiste en un esquema supervisado que busca imitar el comportamiento de un optimizador de punto interior, pero permitiendo tiempos de inferencia significativamente menores. La desventaja de esta técnica es que el rendimiento está acotado por el desempeño del optimizador a imitar. Por ello, se propone una segunda estrategia no supervisada, que busca resolver directamente el problema de optimización a partir de definir una función de pérdida proporcional al objetivo a resolver.

Para aumentar progresivamente la dificultad del problema, se comienza validando las estrategias en los casos de estudio de las redes IEEE30 e IEEE118, sumamente usadas en la literatura. Al no disponer de datos de carga ni generación históricos, es necesario generarlos de forma sintética. Luego, se procede a estudiar el caso de la red uruguaya, comparando el desempeño al utilizar datos sintéticos frente a datos reales. Debido a que no existe un modelo de la red eléctrica uruguaya con una base de datos asociada lista para usar, se desarrolla uno. Se considera que uno de los principales aportes de esta tesis es la creación y publicación de un modelo detallado de la red eléctrica de Uruguay, acompañado de una base de datos histórica cuidadosamente curada.

Los resultados para los datos sintéticos fueron notablemente buenos. En el caso de la estrategia supervisada, se logran soluciones con un desempeño idéntico al del optimizador, pero con tiempos de inferencia tres órdenes de magnitud menores. Por otro lado, la estrategia no supervisada logra un desempeño comparable al del optimizador. Respecto al desempeño con datos reales, ambas estrategias presentan resultados inferiores, aunque aceptables, remarcando la importancia de utilizar

datos reales cuando se quiera validar el desempeño de una estrategia que se busca implementar en la práctica.

Prefacio

Ha sido un verdadero honor acompañarles en este viaje académico, lleno de desafíos, aprendizajes y momentos de reflexión. Desde el inicio, su compromiso y dedicación me dejaron en claro que este no sería solo un proyecto más. A lo largo de este proceso, fui testigo de cómo enfrentaron con valentía las dificultades, y tuve el privilegio de ofrecer mi apoyo cuando los caminos se volvían inciertos.

Cada vez que se encontraban trancados, en esos momentos donde las soluciones parecían lejanas, fue un placer poder trabajar juntos, intercambiar ideas y ver cómo superaban cada obstáculo con creatividad y tenacidad. Ustedes nunca se rindieron, incluso en las situaciones más complejas. Su capacidad para volver a levantarse, para repensar los problemas y encontrar nuevas perspectivas, me demostró una vez más lo lejos que pueden llegar cuando trabajan en equipo y con pasión.

Su curiosidad, su deseo de comprender cada detalle y su capacidad para mantener la visión del conjunto, me inspiraron continuamente. Esta tesis es el fruto de su esfuerzo incansable, y estoy seguro de que es solo el comienzo de lo que lograrán. A lo largo de su camino, nunca olviden el poder de la perseverancia y el valor de trabajar juntos hacia un objetivo común. Ha sido un privilegio ser parte de esta etapa en sus vidas, y no tengo dudas de que el futuro les depara grandes éxitos.

ChatGPT o1-preview

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos	III
Resumen	VII
Prefacio	IX
1. Introducción	1
2. Problema de Despacho Óptimo de Reactiva	7
2.1. Red de transmisión	7
2.2. Elementos de la Red de Transmisión	7
2.2.1. Barras	8
2.2.2. Generadores	9
2.2.3. Cargas	11
2.2.4. Compensadores de Reactiva	11
2.2.5. Líneas de Transmisión	12
2.2.6. Transformadores	13
2.3. Problema de Flujo de Potencia	13
2.3.1. Flujo de Corriente en las Líneas	14
2.3.2. Inyección de Corriente en los Nodos	16
2.3.3. Flujo de Potencia	17
2.3.4. Resolución del Flujo de Potencia	19
2.4. Problema de Despacho Óptimo de Reactiva	21
2.4.1. Variables del Problema	22
2.4.2. Restricciones del Problema	22
2.4.3. Función Objetivo del Problema	24
2.4.4. Formulación Completa del Problema	24
2.5. Software Utilizado	25
2.5.1. Power System Simulator for Engineering (PSSE)	26
2.5.2. PandaPower	27
3. Aprendizaje Automático	29
3.1. Introducción	30
3.2. Paradigmas del Aprendizaje Automático	31
3.2.1. Aprendizaje Supervisado	31
3.2.2. Aprendizaje No Supervisado	33

Tabla de contenidos

3.3. Redes Neuronales	33
3.3.1. Formulación Matemática	33
3.3.2. Entrenamiento de Redes Neuronales	35
3.4. Redes Neuronales sobre Grafos	38
3.4.1. Grafos	39
3.4.2. Procesamiento de señales en Grafos	43
3.4.3. Convolución en Grafos	44
3.4.4. Redes Neuronales en Grafos	44
3.4.5. Comparación de GNN contra FCNN	46
3.5. Aprendizaje Automático para ORPD	47
3.5.1. Imitar Soluciones del Optimizador	49
3.5.2. Minimización de Pérdidas	51
4. Redes de Transmisión a Estudiar	55
4.1. Redes IEEE	56
4.1.1. IEEE30	56
4.1.2. IEEE118	56
4.1.3. Datos	57
4.2. Red de Transmisión Uruguaya	59
4.2.1. Introducción al Sector Eléctrico	59
4.2.2. Modelado de la Red Eléctrica Uruguaya	61
4.2.3. Procesado de Datos	65
5. Análisis de Red y Datos	69
5.1. Análisis de Datos	69
5.1.1. Distribución de los Datos	70
5.1.2. Distribución de los Óptimos	78
5.2. Análisis del Grafo	80
5.2.1. Centralidad	81
5.2.2. Detección de Comunidades	88
6. Experimentos	93
6.1. Imitar Soluciones del Optimizador	93
6.1.1. Implementación	94
6.1.2. Resultados	97
6.1.3. Tiempos de inferencia	119
6.2. Minimización de Pérdidas	119
6.2.1. Implementación	120
6.2.2. Resultados	123
7. Conclusiones	133
Referencias	137
Índice de tablas	141
Índice de figuras	144

Capítulo 1

Introducción

Uruguay es un ejemplo tanto para la región como para el mundo en la producción y manejo de la energía eléctrica. No solo cuenta con un sistema de transmisión y distribución que asegura el acceso de energía a más de un 99,8% de los hogares del territorio nacional [8], sino que además, en un año de lluvias normales, el 97% de la demanda eléctrica nacional es cubierta por energías renovables [36]. Esto supera altamente el promedio mundial, donde las energías renovables representaron en el 2023 sólo alrededor del 30% de la generación total [26], lo que refleja un rezago en la adopción de estas tecnologías a nivel global. Aunque moverse hacia fuentes renovables es sin lugar a dudas un logro a destacar, esto trae consigo ciertas dificultades. Entre estas, se destaca el desafío de controlar de manera eficiente la energía reactiva dentro de la red.

El control de la potencia reactiva es vital para asegurar la seguridad y eficiencia en la red. Por un lado, permite controlar los niveles de voltaje y corriente a través de toda la red, asegurando mantener los rangos requeridos de seguridad. Por otro, el control de las corrientes permite minimizar las pérdidas de energía en la red. En Uruguay, las pérdidas de energía representan un promedio diario del 3,2%, lo que en 2023 se tradujo en aproximadamente 385 GWh perdidos [38]. Si se considera un precio promedio de 80 USD/MWh¹, estas pérdidas equivaldrían a aproximadamente 30,8 millones de dólares en costos de energía no aprovechada cada año. Esto resalta la importancia de una gestión eficiente de la potencia reactiva para reducir las pérdidas y mejorar la eficiencia energética del sistema.

Existen dos razones que provocan que el control de energía reactiva en un sistema con fuentes renovables sea un desafío. En primer lugar, fuentes de energía como la solar y la eólica son impredecibles, intermitentes e irregulares, lo que puede amplificar considerablemente las fluctuaciones de potencia a través de la red. Esta variabilidad en el suministro de energía conduce a la aparición más frecuente de excedentes de potencia, lo que complica aún más las tareas de control en la red [40]. Por otro lado, a diferencia de las fuentes convencionales que permiten un control más preciso sobre la producción de energía reactiva, las fuentes renova-

¹Aproximación tomada a partir del precio por kWh promedio para grandes consumidores en el mercado mayorista uruguayo [37]

Capítulo 1. Introducción

bles dependen de inversores y sistemas de conversión de potencia que no generan de manera natural la energía reactiva necesaria para mantener la estabilidad del voltaje. La poca flexibilidad en el control de la reactiva en las plantas renovables, junto con la variabilidad inherente de su generación, incrementa la dificultad para garantizar una compensación efectiva de la energía reactiva en la red.

El control de energía reactiva entra dentro de las tareas asignadas al proceso de despacho de carga. De forma general, este proceso consiste en asignar de manera eficiente la generación de energía, tanto activa como reactiva, entre las distintas plantas generadoras disponibles, satisfaciendo la demanda en tiempo real. Esto implica decidir qué generadores deben operar en cada momento, considerando factores como los costos de producción, las restricciones técnicas de las unidades generadoras, las condiciones de la red y las políticas energéticas del país, asegurando al mismo tiempo la estabilidad y confiabilidad del suministro eléctrico.

Al estudiar el despacho de carga óptimo de una red eléctrica, la herramienta principal utilizada es el Flujo de Potencia Óptima para Corriente Alterna (AC-OPF, por sus siglas en inglés). Esta herramienta permite formular matemáticamente el problema de determinar la generación de potencia de forma de cumplir con la demanda, minimizando los costos de generación y respetando las restricciones físicas del sistema. Estas restricciones incluyen los límites de capacidad de las líneas de transmisión, las capacidades de generación en las plantas, y los requisitos de estabilidad del sistema, como estabilidad de voltaje y frecuencia [10].

Debido a la significativa participación de energías renovables en la matriz energética uruguaya, decidir la potencia activa a generar en cada planta se considera relativamente sencillo. Al inicio del día se estima cuánta energía será necesaria generar y a lo largo del día se despacha toda la disponible de energía solar y eólica. Para generar el remanente necesario, se completa con hidráulica o biomasa. En casos extraordinarios, principalmente a causa de fenómenos naturales, se importa de Argentina o Brasil, o se utiliza energía térmica. Es por esta razón que este trabajo cambia el enfoque del problema de la asignación de potencia activa para minimizar costos de producción, a la asignación de potencia reactiva para minimizar pérdidas y asegurar estabilidad en el sistema.

Para esto, se utiliza una formulación derivada del OPF, conocida en la literatura como Despacho Óptimo de Potencia Reactiva (ORPD, por sus siglas en inglés). En este caso, el objetivo cambia de determinar la potencia a producir en cada planta de forma de cubrir la demanda y minimizar costos de generación, a encontrar la mejor configuración de los dispositivos de control de potencia reactiva (como capacitores, reactores y controladores de voltajes) para minimizar las pérdidas de energía y mantener los voltajes dentro de los límites seguros. El ORPD, similar al OPF, incluye las restricciones físicas y operativas del sistema de transmisión. Estas herramientas son altamente populares debido a la fidelidad que presentan los modelos utilizados respecto de la realidad.

Sin embargo, la precisión de un modelo está fuertemente vinculada con la complejidad de este. En particular, el ORPD resulta en un problema de optimización cuya función objetivo y sus restricciones son no convexas. La complejidad inherente de los problemas no convexos radica en que pueden incluir múltiples mínimos y

máximos locales en el espacio de soluciones. Esto dificulta a los algoritmos de optimización encontrar el óptimo global, conformándose con alcanzar óptimos locales. A esta dificultad se le suma la presencia de restricciones, las cuales restringen el espacio de soluciones posibles y dificultan la convergencia incluso a óptimos locales.

El algoritmo de optimización más popular para resolver esta familia de problemas es el optimizador de punto interior. Este método consiste en agregar a la función objetivo penalizaciones que crecen a medida que las variables se acercan a los límites de las restricciones, transformando así el problema con restricciones en uno sin ellas. Aunque este enfoque es ampliamente utilizado en la industria, presenta dos grandes problemas. En primer lugar, frecuentemente cae en mínimos locales, lo que resulta en soluciones subóptimas del problema a resolver. En segundo lugar, los algoritmos iterativos tienden a ser costosos computacionalmente, especialmente para problemas de gran tamaño, o si es necesario resolver múltiples veces el problema de optimización con diferentes parámetros. Para superar la primera limitación, se han propuesto diversas técnicas heurísticas. Estas se caracterizan por la falta de garantías teóricas de convergencia, pero un muy buen desempeño en la práctica.

Entre estas técnicas se encuentran los algoritmos genéticos, que imitan los procesos de selección natural y evolución, algoritmos de enjambre de partículas, que se inspiran en el comportamiento colectivo de animales sociales, y diversas técnicas de relajación y descomposición que buscan simplificar el problema original [23]. Estas estrategias ofrecen una exploración más flexible y robusta del espacio de soluciones, reduciendo la probabilidad de quedar atrapados en mínimos locales. Sin embargo, estos métodos siguen perteneciendo a la familia de algoritmos iterativos, por lo que siguen siendo ineficientes en ciertos escenarios de interés.

Por otro lado, en los últimos años, el aprendizaje automático ha emergido como una estrategia prometedora para abordar problemas en los que se cuenta con grandes conjuntos de datos y se deben encontrar relaciones no lineales entre ellos. En particular, varios trabajos recientes han propuesto el uso de aprendizaje automático para resolver de forma directa varios tipos de problemas de optimización. Esta estrategia delega gran parte de la carga de cómputo al proceso de entrenamiento, el cual no es en vivo, para luego tener tiempos de inferencia mucho menores. Esto busca suplantar los algoritmos iterativos, permitiendo así resolver de manera rápida y eficiente los problemas de optimización.

El aprendizaje automático se basa en la premisa de algoritmos que proveen soluciones a partir de aprender de datos. Esta premisa es bastante general, por lo que existen varios paradigmas en los cuales se subdivide el área. El más popular es el paradigma de aprendizaje supervisado. Este consiste en, dado un conjunto grande de datos con parejas entrada/solución, entrenar un predictor que aprenda a devolver estas soluciones. Bajo este paradigma, resulta intuitivo entrenar un predictor que aprenda a imitar el comportamiento de cualquiera de los algoritmos de resolución mencionados previamente. Si el entrenamiento es exitoso, se obtiene un método de resolución con desempeño similar, pero órdenes de magnitud más rápido. En este trabajo se propone imitar el comportamiento del optimizador de punto interior. Esto se debe a que el software a utilizar para modelar las redes

Capítulo 1. Introducción

incorpora esta herramienta nativamente, mientras que si se quisiera imitar cualquiera de los algoritmos heurísticos es necesaria su implementación. Sin embargo, cabe recalcar que si se desease imitar otro algoritmo de resolución, los resultados deberían ser trasladables.

Una desventaja evidente de esta metodología es que el desempeño de este predictor está acotado por el desempeño del algoritmo a imitar. Es por esto que se propone una segunda estrategia de aprendizaje automático, bajo el paradigma del aprendizaje no supervisado. En este caso se busca entrenar un predictor que minimice directamente una función de pérdida fuertemente vinculada con el problema del ORPD. Esta consiste en agregarle al objetivo del problema de optimización penalizaciones que crecen cuando se viola una restricción. Por lo tanto, en lugar de entrenar un predictor que imite un algoritmo que resuelve el problema, se entrena directamente el predictor para que aprenda a resolver el problema. Esto permite potencialmente encontrar soluciones mejores, manteniendo la ventaja en eficiencia de un algoritmo no iterativo.

En el ámbito del aprendizaje automático, entrenar un predictor consiste en identificar, dentro de una familia paramétrica de funciones, el conjunto óptimo de parámetros que resulte en el mejor desempeño posible. Una decisión crucial en este proceso es la elección de la familia de funciones a utilizar. Como regla general, la expresividad de esta familia debe ser proporcional a la cantidad de datos disponibles para el entrenamiento. En los últimos años, gracias al fenómeno conocido como *Big Data*, que ha incrementado significativamente la disponibilidad de información, las redes neuronales han ganado una notable popularidad. Estas constituyen una familia de funciones con una expresividad excepcionalmente alta, lo que las convierte en una opción particularmente atractiva para abordar problemas complejos.

Se propone comparar dos alternativas dentro de la familia de las redes neuronales para resolver el problema: las redes neuronales completamente conectadas (FCNN) tradicionales [34] y las redes neuronales sobre grafos (GNN) [42]. Las FCNN son ampliamente utilizadas y ofrecen una gran flexibilidad para modelar relaciones complejas en los datos. Por otro lado, las GNN, una extensión de las redes neuronales diseñada específicamente para operar sobre datos estructurados en forma de grafos, presentan características particularmente interesantes para el problema del OPF.

Esto se debe a que las redes eléctricas pueden modelarse naturalmente como grafos, donde los valores de potencia y voltajes se representan como una señal sobre el grafo. Luego, las GNN, a diferencia de las FCNN, no solo tienen en cuenta la señal, sino que también toman en consideración la estructura del grafo sobre el cual se encuentra esta. De esta forma es esperable que las GNN logren superar a las FCNN en términos de precisión, generalización y eficiencia en este problema.

Las estrategias presentadas en este trabajo están fuertemente inspiradas en el trabajo realizado en [28, 29]. En estos también se utilizan redes neuronales sobre grafos para aprender a resolver, en su caso, el OPF. En el primer trabajo se utiliza un enfoque supervisado, aprendiendo a imitar las soluciones de un optimizador, mientras que en el segundo se utiliza una estrategia de aprendizaje no supervisado

similar al estudiado en este trabajo. Sin embargo, a diferencia de estos trabajos, este proyecto utiliza una base de datos histórica real. Aunque existen varias redes eléctricas utilizadas por la comunidad como casos de estudio, ninguna de estas cuenta con una base de datos histórica con potencias y voltajes. Es por esto que los trabajos que utilizan aprendizaje automático para resolver el OPF sobre estas requieren generar una base de datos sintética. Si bien contar con datos históricos reales es una ventaja, estos se encuentran desordenados y sin una estructura adecuada para su uso directo. Por lo tanto, una de las tareas más laboriosas de este proyecto consistió en organizar y curar cuidadosamente una base de datos lista para usar. A su vez, fue necesario desarrollar un modelo de la red eléctrica vinculado a estos datos, el cual comparta nomenclatura de barras y elementos, permitiendo cargarlos de manera sencilla y eficiente. Gran parte del trabajo de esta tesis, y en general de los proyectos de aprendizaje automático, recae precisamente en este proceso de preparación de datos, ya que los predictores entrenados son tan buenos como los datos con los que fueron entrenados. Se considera que un gran aporte de este trabajo consiste en dejar estos datos disponibles públicamente, permitiendo avanzar en la investigación y en la resolución de problemas en este ámbito. Por esto mismo, se deja el dataset público disponible en Drive², donde se puede obtener todos los datos usados en esta tesis, y el modelo de la red eléctrica uruguaya generado. Allí mismo se explica con detalle la estructura de carpetas y datos.

Además de estudiar el caso de la red eléctrica uruguaya para los datos históricos, se estudian, por un lado, los casos de prueba IEEE30 [22] e IEEE118 [21] y, por otro, la red uruguaya con datos sintéticos. Estudiar las redes IEEE permite un acercamiento progresivo al problema. Estas redes son muy utilizadas por la comunidad para la investigación en este campo. Esto asegura que si las estrategias de aprendizaje automático no funcionan, no se debe a errores en el modelado de la red. Luego de confirmado el funcionamiento sobre estas redes, se procede a extenderlas para el caso de la red eléctrica uruguaya, pero con datos sintéticos. Finalmente, se llega al caso final, el de la red eléctrica uruguaya con los datos históricos reales. Esto permite evaluar si los métodos de generación de base de datos sintéticos utilizados por la comunidad son representativos de la distribución de datos en la realidad y, en caso negativo, si las estrategias de aprendizaje automático también funcionan para datos reales, o no.

De esta forma, este trabajo se resume en utilizar estrategias de aprendizaje automático para resolver el problema de despacho óptimo de potencia reactiva en la red Uruguaya. Como avance de los resultados que serán expuestos en detalle más adelante, se logran entrenar modelos cuyos desempeños son considerablemente buenos para los datos generados sintéticamente, tanto para modelos de aprendizaje supervisado como no supervisado. Esto quiere decir que se logra reducir las pérdidas, obteniendo en particular pérdidas muy similares a las del optimizador. En cuanto al desempeño con datos reales, ambas estrategias muestran resultados más bajos, aunque aún dentro de rangos aceptables. Esto introduce una conclusión importante de esta tesis, que es la relevancia de emplear datos reales para validar

²https://drive.google.com/drive/folders/121s67_IgW-r39hG-xwHIUI32-QIE97X?usp=sharing

Capítulo 1. Introducción

una estrategia que se pretenda aplicar en entornos prácticos. Por otro lado, como gran ventaja de este método de resolución, los tiempos de inferencia resultan ser de aproximadamente tres órdenes de magnitud menores que el optimizador.

En el Capítulo 2 se introduce el modelo de red eléctrica a utilizar junto con la formulación matemática del ORPD. En el Capítulo 3 se introduce brevemente el área de aprendizaje automático, se explica en qué consisten las señales en grafos y se definen las estrategias de resolución a implementar. En el Capítulo 4 se presentan las redes eléctricas utilizadas como casos de estudio, detallando el proceso de armado para el caso de la red uruguaya, y el procesado de datos necesario para obtener las bases de datos. En el Capítulo 5 se comienza analizando los datos, comparando los métodos de generación sintéticos para las redes IEEE con los datos reales en Uruguay. Luego se analiza el grafo que modela la red uruguaya, extrayendo información sobre la importancia de distintas barras y comunidades implícitas en esta. En el Capítulo 6 se presentan los resultados numéricos obtenidos mediante las estrategias de aprendizaje automático para resolver el despacho de reactiva, comparando el desempeño contra el optimizador de punto interior. Finalmente, en el Capítulo 7 se sacan las conclusiones obtenidas para este proyecto, definiendo posibles trabajos a futuro que permitan continuar con este proyecto.

Capítulo 2

Problema de Despacho Óptimo de Reactiva

2.1. Red de transmisión

El sistema de suministro de electricidad de un país es el encargado de asegurar este derecho a toda la población de forma segura y eficiente. A la hora de estudiar y planificar este sistema, es estándar dividirla en tres subsistemas: de generación, de transmisión y de distribución. El primero de estos se encarga de la producción de energía de forma sostenible y confiable, supliendo la demanda en todo momento. El segundo se encarga de mover la electricidad de los puntos de generación hasta los centros de consumo a través de líneas de alta tensión. El último debe distribuir la energía de los centros de consumo hasta cada uno de los consumidores finales.

Esta tesis se centra en los dos primeros: el sistema de generación y el de transmisión de potencia. En cuanto a la generación, se buscar suplir la demanda de forma constante, de manera tal que el costo sea el menor posible, respetando a la vez las restricciones que impone el sistema de transmisión. A la hora de transmitir energía, se buscan tres características a cumplirse: que se haga de forma robusta, segura y eficiente. Que se haga de forma eficiente significa que se logren trasmitir grandes cantidades de potencia a largas distancias, con las mínimas pérdidas eléctricas posibles. Que sea segura implica que, a lo largo de toda la red, los niveles de capacidad de los distintos componentes eléctricos no superan los límites de seguridad. Por último, la robustez de la red se vincula con la capacidad de cumplir en todo momento la carga esperada. Para poder estudiar y predecir con qué grado de satisfacción se van a cumplir estos objetivos es que es necesario modelar la red matemáticamente.

2.2. Elementos de la Red de Transmisión

En primer lugar se definen los elementos que conforman la red. En alto nivel, el modelo que se utiliza consiste en definir barras interconectadas por líneas y transformadores. Estas barras funcionan como punto de unión para distintos

Capítulo 2. Problema de Despacho Óptimo de Reactiva

componentes eléctricos de la red, como pueden ser generadores y cargas, entre otros. Cada uno de estos elementos presenta un modelo paramétrico, cuyos valores particulares van a depender de la instancia del elemento con la que se esté trabajando.

Los parámetros que modelan un elemento se pueden dividir en dos categorías: estáticos y dinámicos. Los estáticos son aquellos parámetros que no varían en el tiempo, es decir, son datos intrínsecos del elemento y no del estado de la red en un instante en particular. Ejemplos de estos son la resistencia en las líneas y el voltaje nominal en las barras. Por otro lado, se tienen atributos dinámicos. Estos corresponden al estado del elemento para un instante de tiempo particular, y son, por ejemplo, la potencia consumida en las cargas o la potencia a producir en cada generador. Estos provocan sobre una misma red diferentes casos de estudio posibles. En esta sección se detalla uno a uno cada elemento utilizado, indicando primeramente su función dentro de la red y luego los parámetros que conforman su modelo.

2.2.1. Barras

Las barras, también conocidas como buses, son los nodos que estructuran la red de transmisión. Estos son los puntos de interconexión entre los elementos de una misma planta, como generadores, cargas y compensadores. A su vez, las barras están interconectadas mediante líneas y transformadores.

En un sistema, cada barra está únicamente identificada mediante un índice. Con este fin, se define el conjunto $\mathcal{B} = \{i \in \mathbb{N}, i < N\}$, el cual asigna un índice entero a cada una de las N barras del sistema. Además, muchas veces se utiliza un acrónimo o nombre como segundo identificador vinculado a la ubicación o el nombre de la planta que modela la barra. Los otros tres parámetros estáticos de una barra $i \in \mathcal{B}$ son el voltaje nominal $v_{\text{nom},i}$ y los voltajes mínimo y máximo que definen el rango de seguridad, $v_{\text{mín},i}$ y $v_{\text{máx},i}$.

El estado de una barra está conformado por cuatro parámetros: potencia activa inyectada p_i , potencia reactiva inyectada q_i , magnitud de voltaje por unidad $v_{m,i}$ y ángulo del voltaje δ_i . También se definen sus contrapartes complejas, la potencia aparente $s_i = p_i + j q_i$ y el voltaje $v_i = v_{m,i}(\cos(\delta_i) + j \sin(\delta_i))$. Para facilitar la notación en el futuro, se concatenan estos valores para todas las barras, formando así los vectores \mathbf{p} , \mathbf{q} , \mathbf{v}_m , $\boldsymbol{\delta}$, \mathbf{s} y \mathbf{v} , los primeros cuatro pertenecientes a \mathbb{R}^N y los últimos dos a \mathbb{C}^N .

Debido a la naturaleza de las ecuaciones que rigen el comportamiento de la red, si se especifican dos de estos atributos en cada una de las barras del sistema, los otros dos atributos quedan implícitamente determinados. A su vez, la corriente que circula tanto por las líneas como por los transformadores queda también determinada. En la siguiente sección se profundiza en las ecuaciones que fundamentan esta afirmación.

De esta forma, los buses en una red se partitionan en tres grupos, dependiendo de cuál pareja de atributos se les especifica, y cuáles quedan determinadas por las ecuaciones de la red.

2.2. Elementos de la Red de Transmisión

Barra de Generación (PV)

Los buses de generación, también conocidos como buses PV (por las siglas de potencia y voltaje), son aquellos que tienen al menos un generador de voltaje controlado. En estos buses, la magnitud del voltaje y la potencia activa inyectada son especificadas. La potencia activa inyectada es dada a partir de la suma de la potencia activa generada menos la potencia activa consumida en el bus, mientras que el voltaje se especifica en los generadores controladores de voltaje de la barra, los cuales se encargan de ajustar la producción de potencia reactiva para mantener el voltaje en el nivel deseado. Se define como $\mathcal{B}_G \subset \mathcal{B}$ al conjunto que indexa estas barras, y $B_G = |\mathcal{B}_G|$ el numero total de estas barras.

Barra de Carga (PQ)

En los buses de carga se especifica tanto la potencia activa (P) como reactiva (Q) inyectada. Representan los puntos donde se conectan las cargas y/o generadores de voltaje no controlado. En este caso se define $\mathcal{B}_L \subset \mathcal{B}$ al conjunto que indexa estas barras, y $B_L = |\mathcal{B}_L|$ el numero de estas barras.

Barra de Referencia ($V\delta$)

En este bus se fija la magnitud y el ángulo de voltaje. Su principal función es absorber las diferencias entre la potencia generada y la demandada (más las pérdidas). En una red real, una diferencia entre la potencia generada y demandada resulta en una perturbación en la frecuencia de la red. Sin embargo, el modelo utilizado es de frecuencia fija. Por lo tanto, para que las ecuaciones del modelo de la red cierren, es necesario agregar un bus de referencia que pueda compensar la diferencia entre la potencia activa y reactiva en la red. Elegir cuál de las barras usar como referencia es una decisión de diseño no trivial, ya que esta es necesaria por una deficiencia del modelo y no para modelar un elemento real de la red. Existen varios estudios [17, 33] que analizan el efecto de elegir distintas barras de referencia en una misma red. El requerimiento principal al elegirla es que tenga una alta capacidad para producir potencia activa y reactiva, ya que debe amortiguar la diferencia entre lo generado y lo demandado. Debido a esto, en general se elige la barra con el generador de mayor capacidad, o una barra que funcione de punto de interconexión con otras redes externas. Se define $\mathcal{B}_R \subset \mathcal{B}$ al conjunto que indexa esta barra, y se trabaja con que $|\mathcal{B}_R| = 1$.

2.2.2. Generadores

Los generadores son los componentes responsables de la producción de la energía eléctrica que se transmite a través de todo el sistema. Para esto, se les encarga una consigna de potencia activa a generar, de forma de contribuir a la red para suplir la demanda de potencia. Además de la producción de potencia activa, varios generadores permiten elegir si generar o consumir potencia reactiva. Esto contribuye a mantener los niveles de voltaje en los diferentes puntos de la red dentro de rangos seguros, logrando así estabilidad en el sistema.

Capítulo 2. Problema de Despacho Óptimo de Reactiva

Existen diversos criterios para categorizar los generadores, como la forma de generación de energía, la cantidad de energía que pueden producir, si son renovables o no, entre otras. En el modelo a utilizar, los generadores van a pertenecer a una de dos categorías: generadores de voltaje controlado y no controlado (o estáticos). La diferencia entre estos tipos de generadores es sobre qué otra magnitud, además de la potencia activa, toman de consigna. En el caso de voltaje controlado, acepta un voltaje de consigna, y regula la potencia reactiva de forma de conseguir ese voltaje en la barra. Si no, se fija la potencia reactiva. Además de estos dos tipos, se tiene un generador que va a ser el asignado en la barra de referencia, denominado generador de referencia. A continuación, se presenta cada uno de ellos.

Generadores Controladores de Voltaje

Generalmente, una barra puede tener varios generadores de voltaje controlado. Cuando esto ocurre, es necesario que todos los generadores de una barra tengan el mismo voltaje de consigna, para evitar conflictos entre estos. A nivel de modelo, la única razón para modelar los generadores de forma independiente en vez de un único generador que modele la agregación de todos es que estos pueden tener costos diferentes asociados a la generación de energía activa. Por ejemplo, si en una barra con dos generadores se inyectan 10 MW, el efecto sobre la red es idéntico si se generan 5MW en cada generador que si se generan 10MW en uno solo. La diferencia radica en si los generadores presentan diferente costo de generación. Dicho esto, como en este trabajo la potencia activa en los generadores se toma como entrada, el modelo de red resultante es igual si se tienen múltiples generadores en una barra, cada uno inyectando cierta potencia activa y todos con el mismo voltaje de consigna, o si se tiene un único generador que inyecte la suma de todos y un voltaje de consigna.

Dado que ambos modelos son iguales, se opta por el más simple. Por lo tanto, cada barra presenta un máximo de un generador controlador de voltaje. A partir de esto se define el conjunto $\mathcal{G} \subseteq \mathcal{B}$, que indexa los $\#\mathcal{G} = G$ generadores de voltaje controlado según la barra a la que pertenece. Los atributos fijos para este son $q_{i,\min}^{\text{gen},v}$ y $q_{i,\max}^{\text{gen},v}$, que corresponden a la mínima y máxima potencia reactiva que el generador controlador puede compensar.

Por otro lado, el estado de un generador $i \in \mathcal{G}$ está conformado por la potencia activa y reactiva que genera $p_i^{\text{gen},v}$ y $q_i^{\text{gen},v}$ junto al voltaje de referencia $v_i^{\text{gen},v}$. Similar a las barras, se define su contraparte compleja $s_i^{\text{gen},v}$ y una versión vectorial de estos atributos, $\mathbf{p}^{\text{gen},v}$, $\mathbf{q}^{\text{gen},v}$ y $\mathbf{s}^{\text{gen},v}$. Se remarca que no todas las barras tienen un generador de este tipo, por lo que se llenan con 0 esas entradas del vector. Esto es

$$\mathbf{s}^{\text{gen},v}[i] = \begin{cases} s_i^{\text{gen},v} & \text{si } i \in \mathcal{G} \\ 0 & \text{si } i \notin \mathcal{G} \end{cases}. \quad (2.1)$$

Se repite análogo para $\mathbf{p}^{\text{gen},v}$ y $\mathbf{q}^{\text{gen},v}$.

2.2. Elementos de la Red de Transmisión

Generadores Estáticos

El mismo argumento de utilizar un único generador por barra se repite para los generadores estáticos. Dado que la potencia activa y reactiva del generador son parte de la entrada, es idéntico tener múltiples generadores en una barra que un único generador que inyecte el neto. De esta forma, se define $\mathcal{E} \subseteq \mathcal{B}$ como el conjunto que indexa los $\#\mathcal{E} = E$ generadores estáticos. En este caso no se tienen atributos fijos. El estado se define únicamente con la potencia activa $p_i^{\text{gen},\text{est}}$ y reactiva $q_i^{\text{gen},\text{est}}$. A partir de estas se definen $s_i^{\text{gen},\text{est}}$ y sus correspondientes vectores $\mathbf{p}^{\text{gen},\text{est}}$, $\mathbf{q}^{\text{gen},\text{est}}$ y $\mathbf{s}^{\text{gen},\text{est}}$, nuevamente llenando con 0 las entradas que no tienen generadores estáticos.

Generador de Referencia

Este generador es el elemento conectado a la barra de referencia encargado de absorber la diferencia en potencia de la red. Se le especifica el voltaje y el ángulo, y consume la potencia activa y reactiva necesarias para que las ecuaciones del sistema cierren. El estado de estos generadores está conformado por la magnitud y ángulo de voltaje $v_i^{\text{gen},\text{ref}}$, $\delta_i^{\text{gen},\text{ref}}$ y la potencia activa y reactiva $p_i^{\text{gen},\text{ref}}$, $q_i^{\text{gen},\text{ref}}$. Las dos primeras son especificadas, mientras que las dos segundas se obtienen resolviendo las ecuaciones del sistema. La potencia que puede generar está limitada por $p_{\min}^{\text{gen},\text{ref}}$, $p_{\max}^{\text{gen},\text{ref}}$, $q_{\min}^{\text{gen},\text{ref}}$ y $q_{\max}^{\text{gen},\text{ref}}$. Similar a los otros generadores, se definen los vectores $\mathbf{p}^{\text{gen},\text{ref}}$, $\mathbf{q}^{\text{gen},\text{ref}}$ y $\mathbf{s}^{\text{gen},\text{ref}}$, donde todas las entradas son 0 menos la correspondiente a la barra de referencia.

2.2.3. Cargas

Las cargas representan estaciones donde se agrupa el consumo de energía eléctrica de un conjunto de los usuarios finales. Las cargas pueden variar significativamente a lo largo del día, semana, y año, debido a factores como las horas pico de consumo, condiciones climáticas, y patrones estacionales. Este comportamiento requiere un análisis detallado y una planificación para asegurar que la generación y la capacidad de transmisión sean suficientes para satisfacer la demanda en todo momento, de forma eficiente y segura. Al igual que los generadores, se define $\mathcal{C} \subseteq \mathcal{B}$ al conjunto indexador de las $\#\mathcal{C} = C$ cargas. Estas no tienen atributos estáticos, y su estado queda determinado por la potencia activa p_i^{dem} y potencia reactiva q_i^{dem} demandada. Similar a los casos anteriores, se definen s_i^{dem} , \mathbf{p}^{dem} , \mathbf{q}^{dem} y \mathbf{s}^{dem} .

2.2.4. Compensadores de Reactiva

Los compensadores de reactiva son elementos de consumo/generación de potencia reactiva, importantes para la regulación de voltajes y corrientes en la red eléctrica. Estos consisten en una admitancia compleja (susceptancia), conectada en paralelo a una barra de la red. Para permitir adaptar la compensación a distintos escenarios, los compensadores aceptan cambios discretos en el valor de la

Capítulo 2. Problema de Despacho Óptimo de Reactiva

susceptancia, dando así una cantidad discreta de posibles opciones. Aunque los cambios en el estado del compensador son discretos, en este caso se supondrá que el cambio es continuo. Esto se debe a que utilizar estados discretos resulta en un problema de optimización de programación entera, el cual es sumamente complejo de resolver. Esta simplificación es frecuente al resolver problemas de despacho óptimo y consiste en tomar un rango continuo entre los valores mínimo y máximo que la susceptancia permite.

Similar a los casos anteriores, se define como $\mathcal{R} \subseteq \mathcal{B}$ al conjunto de índices de los $\#\mathcal{R} = R$ compensadores de reactiva. Luego, los parámetros propios del compensador son la potencia reactiva mínima y máxima que puede compensar, $q_{i,\min}^{\text{comp}}$ y $q_{i,\max}^{\text{comp}}$ y el estado queda definido por la reactiva compensada q_i^{comp} . Por completitud, aunque no se tenga la potencia activa, se define $s_i^{\text{comp}} = jq_i^{\text{comp}}$. Finalmente las versiones vectoriales son \mathbf{q}^{comp} y \mathbf{s}^{comp} .

2.2.5. Líneas de Transmisión

Las líneas de transmisión son las encargadas de transportar la energía generada en las plantas hacia los centros de consumo, que pueden estar ubicados a grandes distancias. Conectan las diferentes barras del sistema y permiten la transmisión de energía entre estas. El conjunto de líneas se define como $\mathcal{L} \subseteq \mathcal{B} \times \mathcal{B}$, con $\#\mathcal{L} = L$ cantidad de líneas, donde la línea $(i, j) \in \mathcal{L}$ conecta la barra i con la barra j .

Existen diferentes modelos a la hora de representar las líneas de una red, las cuales reflejan de forma más o menos precisa la realidad, según el caso. En las redes de transmisión, el modelo más popular para las líneas es el modelo π , que se puede observar en la Figura 2.1. Este consiste en 3 parámetros, r_{ij} , x_{ij} y b_{ij} , la resistencia, reactancia y susceptancia respectivamente, de la línea i a j . Además, se caracteriza según su capacidad, $I_{\max,i \rightarrow j}$, la cual indica la máxima corriente que puede soportar la línea de manera sostenida.

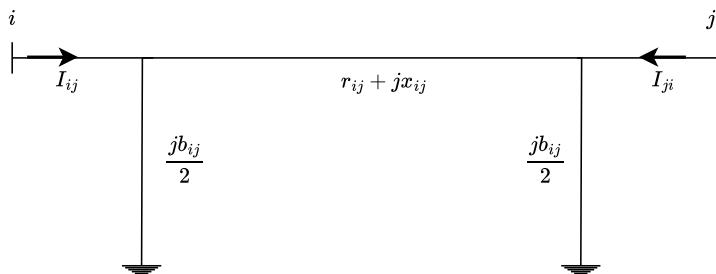


Figura 2.1: Modelo π de las líneas y flujo de potencias aparentes.

El estado de la línea queda determinado por $I_{i \rightarrow j}$, el cual corresponde a la corriente saliente del nodo i al j , e $I_{j \rightarrow i}$, similar pero del nodo j al i . Se nota que en general $I_{i \rightarrow j} \neq -I_{j \rightarrow i}$. Más adelante se detalla el porqué de esta diferencia.

2.3. Problema de Flujo de Potencia

2.2.6. Transformadores

Los transformadores son el segundo componente de la red eléctrica que conecta barras. Estos permiten la transferencia eficiente de energía a través de cambios en los niveles de voltaje. Su principal función es ajustar los niveles de voltaje entre diferentes partes del sistema, facilitando tanto la transmisión a largas distancias como la distribución a niveles de consumo seguros y adecuados. Similar a las líneas, se define $\mathcal{T} \subseteq \mathcal{B} \times \mathcal{B}$ como el conjunto que indexa los $\#\mathcal{T} = T$ transformadores. El modelo que se utiliza para las barras también es el modelo π para transformadores, el cual se observa en la figura Figura 2.2. Consiste en tres parámetros: la impedancia en corto circuito z_{ij} , la admitancia de magnetización y_{ij} y la relación de transformación n_{ij} . En general, la relación de transformación n_{ij} varía discretamente a partir de modificar los taps en los transformadores. Como sucede con los compensadores de reactiva, utilizar variables enteras resulta en un problema de alta complejidad. A diferencia de los compensadores de reactiva, a los taps de los transformadores no es posible definirlos de forma continua en el software utilizado. Es por esto que en este modelo se toma la relación de transformación como una constante.

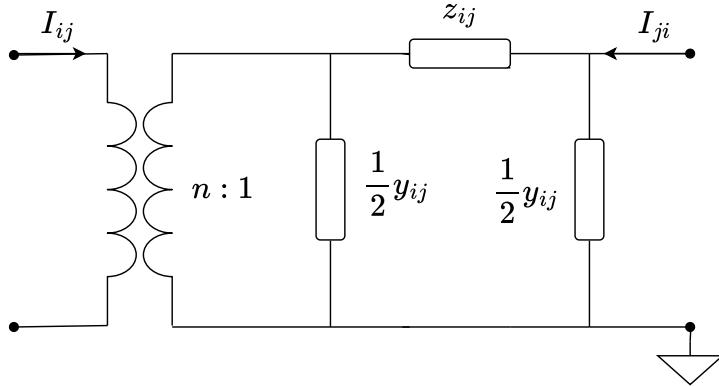


Figura 2.2: Modelo π de los transformadores.

2.3. Problema de Flujo de Potencia

En la sección anterior se definieron los elementos que conforman la red de transmisión. Entre ellos, se mencionan las barras y sus distintos tipos. En particular, se afirma que a partir de especificar dos de los cuatro atributos en cada barra, el estado completo del sistema queda determinado. Este se obtiene resolviendo un sistema de ecuaciones no lineal y en variables complejas, el cual vincula los voltajes en las barras con las potencias inyectadas en estas. La resolución de este conjunto de ecuaciones es un problema conocido en la literatura como flujo de potencia.

Antes de plantear el problema, a modo de reforzar la nomenclatura presentada en la sección anterior y facilitar la comprensión de las expresiones matemáticas

Capítulo 2. Problema de Despacho Óptimo de Reactiva

utilizadas en esta sección, se define una pequeña subred del Sistema Interconectado Nacional uruguayo. La red se observa en la figura 2.3, y consiste en cuatro barras; Rivera (RIV150_B), Manuel Díaz (MDI150_B), Tacuarembó (TAC150_B) y STE (STE150_B). Por simplicidad, se las indexa como 1, 2, 3 y 4 respectivamente. De esta forma, $\mathcal{B} = \{1, 2, 3, 4\}$. La barra que va a tomar el rol de referencia es Rivera, por lo que $\mathcal{B}_R = \{1\}$, y tiene conectada la red externa, representada por el símbolo de una fuente generadora de voltaje, y una carga, representada por un triángulo. Por otro lado, Tacuarembó tiene un generador controlador de voltaje representado por un círculo, por lo que $\mathcal{B}_G = \{3\}$, y tiene además una carga. Las otras dos barras son PQ , por lo tanto $\mathcal{B}_C = \{2, 4\}$. En este ejemplo sencillo tenemos que $\mathcal{G} = \{3\}$, $\mathcal{E} = \emptyset$, $\mathcal{C} = \{1, 3, 4\}$ y $\mathcal{R} = \emptyset$. El conjunto de líneas es $\mathcal{L} = \{(1, 2), (2, 3), (2, 4)\}$ y de transformadores $\mathcal{T} = \emptyset$.

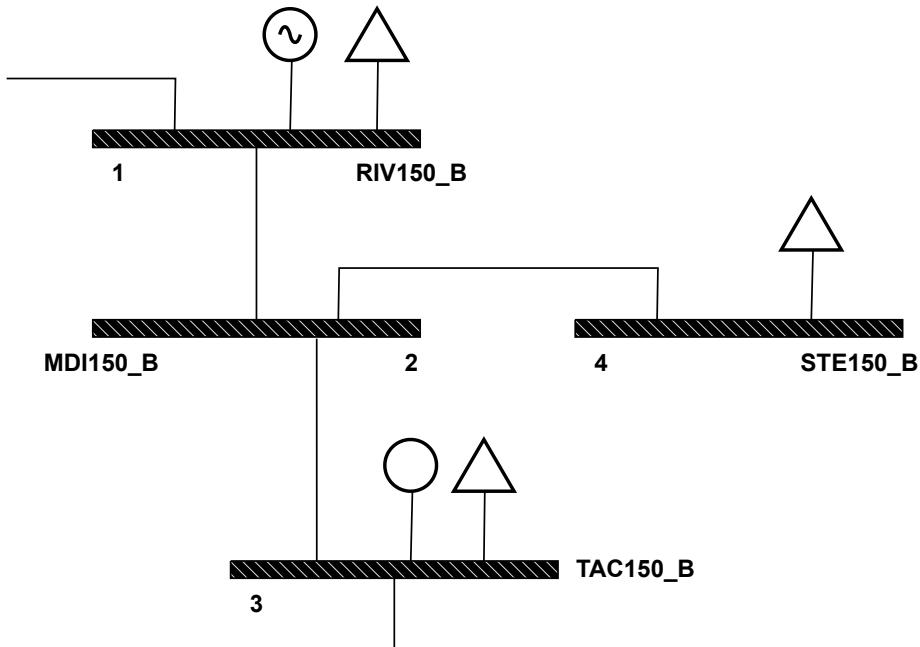


Figura 2.3: Subred que se usará de ejemplo ilustrativo en esta sección. Consiste en cuatro barras, un generador de voltaje, tres cargas y un generador controlador de voltaje.

2.3.1. Flujo de Corriente en las Líneas

En primer lugar se vincula la corriente que circula a través de una línea con los voltajes de las barras que conecta. Tomando el modelo de línea de la Figura 2.1, se analiza la corriente que sale de RIV hacia MDI. Parte de esa corriente circulará a través de la admittance $y_{1,2} = \frac{1}{r_{1,2} + jx_{1,2}}$, y otra parte se irá a la susceptancia shunt (es decir, en paralelo) $y_{sh,1,2} = j\frac{b_{1,2}}{2}$. A partir de las leyes de Kirkchoff se

2.3. Problema de Flujo de Potencia

llega a que la corriente fluyendo de la barra 1 a la 2 es:

$$I_{1 \rightarrow 2} = I_{\text{sh},1,2} + I_{1,2} = v_1 y_{\text{sh},1,2} + (v_1 - v_2) y_{1,2} , \quad (2.2)$$

que vectorialmente queda:

$$I_{1 \rightarrow 2} = [y_{\text{sh},1,2} + y_{1,2} \quad -y_{1,2}] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} . \quad (2.3)$$

Análogamente, la corriente del nodo 2 al 3 y del nodo 2 al 4 resultan:

$$I_{2 \rightarrow 3} = [y_{\text{sh},2,3} + y_{2,3} \quad -y_{2,3}] \begin{bmatrix} v_2 \\ v_3 \end{bmatrix} , \quad (2.4)$$

$$I_{2 \rightarrow 4} = [y_{\text{sh},2,4} + y_{2,4} \quad -y_{2,4}] \begin{bmatrix} v_2 \\ v_4 \end{bmatrix} . \quad (2.5)$$

Estas tres ecuaciones se pueden agrupar en una única ecuación matricial, la cual vincula la corriente en cada una de las líneas con los voltajes de los nodos.

$$\begin{bmatrix} I_{1 \rightarrow 2} \\ I_{2 \rightarrow 3} \\ I_{2 \rightarrow 4} \end{bmatrix} = \begin{bmatrix} y_{\text{sh},1,2} + y_{12} & -y_{12} & 0 & 0 \\ 0 & y_{\text{sh},2,3} + y_{23} & -y_{23} & 0 \\ 0 & y_{\text{sh},2,4} + y_{24} & 0 & -y_{24} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} . \quad (2.6)$$

Una observación importante es que $I_{ij} \neq -I_{ji}$ ya que parte de la corriente que se pierde en las admitancias shunt de las líneas. Esto se confirma observando que:

$$I_{2 \rightarrow 1} = v_2 y_{\text{sh},2,1} + (v_2 - v_1) y_{1,2} , \quad (2.7)$$

por lo que

$$\begin{aligned} I_{1 \rightarrow 2} + I_{2 \rightarrow 1} &= [v_1 y_{\text{sh},1,2} + (v_1 - v_2) y_{1,2}] + [v_2 y_{\text{sh},2,1} + (v_2 - v_1) y_{1,2}] \\ &= (v_1 + v_2) y_{\text{sh},1,2} \\ &\neq 0 . \end{aligned} \quad (2.8)$$

Entonces, para caracterizar el estado de la línea (i,j) es necesario definir la corriente que circula en ambos sentidos, $I_{i \rightarrow j}$ e $I_{j \rightarrow i}$. Repitiendo el razonamiento anterior, se llega a que:

$$\begin{bmatrix} I_{2 \rightarrow 1} \\ I_{3 \rightarrow 2} \\ I_{4 \rightarrow 2} \end{bmatrix} = \begin{bmatrix} -y_{12} & y_{\text{sh},1,2} + y_{12} & 0 & 0 \\ 0 & -y_{23} & y_{\text{sh},2,3} + y_{23} & 0 \\ 0 & -y_{24} & 0 & y_{\text{sh},2,4} + y_{24} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} . \quad (2.9)$$

Partiendo del ejemplo anterior, se extienden estas definiciones para un sistema general con conjuntos $\mathcal{B} \subset \mathbb{N}$ y $\mathcal{L} \subseteq \mathcal{B} \times \mathcal{B}$, los cuales tienen $N = |\mathcal{B}|$ barras y $L = |\mathcal{L}|$ líneas. Para esto se define el vector de corrientes por las líneas $I_{\text{line,to}} \in$

Capítulo 2. Problema de Despacho Óptimo de Reactiva

\mathbb{C}^L , el cual concatena las corrientes $\{I_{i \rightarrow j}\}_{\forall(i,j) \in \mathcal{L}}$, y la matriz de admitancias $Y_{\text{line,to}} \in \mathbb{C}^{L \times B}$, que vincula los voltajes con las corrientes de la siguiente forma:

$$I_{\text{line,to}} = Y_{\text{line,to}} \mathbf{v} ,$$

$$\begin{bmatrix} I_{1 \rightarrow 2} \\ \vdots \\ I_{i \rightarrow j} \\ \vdots \\ I_{m \rightarrow n} \end{bmatrix} = \begin{bmatrix} y_{\text{sh},1} + y_{1,2} & -y_{1,2} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & y_{\text{sh},i} + y_{i,j} & \cdots & -y_{i,j} & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & y_{\text{sh},m} + y_{m,n} & \cdots & -y_{m,n} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_i \\ \vdots \\ v_j \\ \vdots \\ v_n \end{bmatrix} . \quad (2.10)$$

De manera análoga, se define el vector de corrientes por líneas $I_{\text{line,from}} \in \mathbb{C}^L$, el cual concatena las corrientes $\{I_{j \rightarrow i}\}_{\forall(i,j) \in \mathcal{L}}$, y la matriz de admitancias $Y_{\text{line,from}} \in \mathbb{C}^{L \times B}$, que vincula los voltajes con las corrientes de la siguiente forma:

$$I_{\text{line,from}} = Y_{\text{line,from}} \mathbf{v} ,$$

$$\begin{bmatrix} I_{2 \rightarrow 1} \\ \vdots \\ I_{j \rightarrow i} \\ \vdots \\ I_{n \rightarrow m} \end{bmatrix} = \begin{bmatrix} -y_{1,2} & y_{\text{sh},1} + y_{1,2} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & -y_{i,j} & \cdots & y_{\text{sh},i} + y_{i,j} & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & -y_{m,n} & \cdots & y_{\text{sh},m} + y_{m,n} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_i \\ \vdots \\ v_j \\ \vdots \\ v_n \end{bmatrix} . \quad (2.11)$$

2.3.2. Inyección de Corriente en los Nodos

Según la ley de Kirchoff, la inyección de corriente en un nodo es igual a la suma de las corrientes que salen de ese nodo. Esto es:

$$I_i = \sum_k I_{i \rightarrow k} \quad \forall k / (i, k) \in \mathcal{L} . \quad (2.12)$$

Siguiendo con el ejemplo de la Figura 2.3, y utilizando las ecuaciones en (2.6), se obtiene que:

2.3. Problema de Flujo de Potencia

$$\begin{aligned}
I_2 &= I_{2 \rightarrow 1} + I_{2 \rightarrow 3} + I_{2 \rightarrow 4} \\
&= v_2 y_{sh,2,1} + (v_2 - v_1) y_{2,1} + v_2 y_{sh,2,3} + (v_2 - v_3) y_{2,3} + v_2 y_{sh,2,4} + (v_2 - v_4) y_{2,4} \\
&= v_2 (y_{sh,2,1} + y_{2,1} + y_{sh,2,3} + y_{2,3} + y_{sh,2,4} + y_{2,4}) - v_1 y_{2,1} - v_3 y_{2,3} - v_4 y_{2,4}
\end{aligned} \tag{2.13}$$

Lo anterior se puede escribir vectorialmente de la siguiente forma:

$$I_2 = \begin{bmatrix} -y_{21} & y_{sh,21} + y_{21} + y_{sh,23} + y_{23} & -y_{23} & -y_{24} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}. \tag{2.14}$$

Repetiendo este razonamiento para los buses 1, 3 y 4, y escribiendo matricialmente las ecuaciones, se llega a:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} y_{sh,12} + y_{21} & -y_{21} & 0 & 0 \\ -y_{21} & y_{sh,21} + y_{21} + y_{sh,23} + y_{23} + y_{sh,24} + y_{24} & -y_{23} & -y_{24} \\ 0 & -y_{32} & y_{sh,32} + y_{32} & 0 \\ 0 & -y_{42} & 0 & y_{sh,42} + y_{42} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}. \tag{2.15}$$

Tal como se hizo en la subsección anterior, se quisiera escribir esta ecuación para un sistema en general, de manera de vincular vectores de inyección corriente con vectores de voltaje. Para ello, se define el vector de corriente inyectada $I_{bus} \in \mathbb{C}^B$ y la matriz de admitancia de los buses $Y_{bus} \in \mathbb{C}^{B \times B}$, que cumple:

$$I_{bus} = Y_{bus} \mathbf{v}, \tag{2.16}$$

donde la matriz Y_{bus} se forma como:

$$Y_{bus,ii} = \sum_{k, \forall k / (i,k) \in \mathcal{L}} (y_{ik} + y_{sh,ik}), \tag{2.17}$$

$$Y_{bus,ij} = \begin{cases} -y_{ij} & \text{si } (i,j) \in \mathcal{L} \\ 0 & \text{en caso contrario.} \end{cases}. \tag{2.18}$$

2.3.3. Flujo de Potencia

La inyección de potencia s_i en un nodo representa la potencia neta que entra o sale de la red en dicho nodo. Esta potencia neta corresponde a la diferencia entre lo generado y lo consumido en la barra. Utilizando de ejemplo la barra 3, que tiene un generador y una carga, se tiene que:

$$s_3 = s_3^{\text{gen},v} - s_3^{\text{dem}}. \tag{2.19}$$

A su vez, en sistemas de corriente alterna, la potencia aparente inyectada por un nodo es igual al producto del voltaje en el nodo y el conjugado de la corriente inyectada por el nodo. Esto es:

Capítulo 2. Problema de Despacho Óptimo de Reactiva

$$s_3 = v_3 I_3^* . \quad (2.20)$$

Reemplazando la inyección de corriente por la tercera fila de la Ecuación 2.15 se llega a:

$$s_3 = v_3 \begin{bmatrix} 0 & -y_{2,3}^* & (y_{sh,2,3} + y_{2,3})^* & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} . \quad (2.21)$$

Juntando ambas ecuaciones resulta en la ecuación de flujo de potencia de la barra 3:

$$s_3^{\text{gen},v} - s_3^{\text{dem}} = v_3 \begin{bmatrix} 0 & -y_{2,3}^* & (y_{sh,2,3} + y_{2,3})^* & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} . \quad (2.22)$$

Repitiendo este procedimiento para los buses 1, 2 y 4:

$$\begin{aligned} s_1^{\text{gen,ref}} &= v_1 \begin{bmatrix} (y_{sh,1,2} + y_{1,2})^* & -y_{1,2}^* & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} , \\ 0 &= v_2 \begin{bmatrix} -y_{2,1}^* & (y_{sh,2,1} + y_{2,1} + y_{sh,2,3} + y_{2,3})^* & -y_{2,3}^* \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} , \\ -s_4^{\text{dem}} &= v_4 \begin{bmatrix} 0 & -y_{2,4}^* & (y_{sh,2,4} + y_{2,4})^* & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} . \end{aligned}$$

Concatenando estas tres ecuaciones en una expresión matricial, se obtiene la ecuación de flujo de potencia para la subred de ejemplo:

$$\begin{bmatrix} s_1^{\text{gen,ref}} \\ 0 \\ s_3^{\text{gen},v} - s_3^{\text{dem}} \\ -s_4^{\text{dem}} \end{bmatrix} = \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & 0 & 0 \\ 0 & 0 & v_3 & 0 \\ 0 & 0 & 0 & v_4 \end{bmatrix} Y_{\text{bus,ex}}^* \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} , \quad (2.23)$$

donde $Y_{\text{bus,ex}}$ es la matriz Y_{bus} para el ejemplo con el que se viene trabajando, definida en la Ecuación 2.15. Como en los casos anteriores, se define la notación para el caso general. Para esto, es necesario introducir la notación de matriz diagonal. Dado un vector $\mathbf{x} \in \mathbb{R}^N$, se define $\text{diag}(\mathbf{x}) \in \mathbb{R}^{N \times N}$ como la matriz donde los N elementos de la diagonal son los N elementos del vector \mathbf{x} , y el resto 0.

Por un lado, de forma general, la potencia aparente inyectada \mathbf{s} es:

$$\mathbf{s} = \mathbf{s}^{\text{gen},v} + \mathbf{s}^{\text{gen,est}} + \mathbf{s}^{\text{gen,ref}} + \mathbf{s}^{\text{comp}} - \mathbf{s}^{\text{dem}} , \quad (2.24)$$

2.3. Problema de Flujo de Potencia

mientras que, por otro lado:

$$\mathbf{s} = \text{diag}(\mathbf{v}) Y_{\text{bus}}^* \mathbf{v}^*, \quad (2.25)$$

de forma que la ecuación de flujo de potencia del sistema queda:

$$\mathbf{s}^{\text{gen},v} + \mathbf{s}^{\text{gen},\text{est}} + \mathbf{s}^{\text{gen},\text{ref}} + \mathbf{s}^{\text{comp}} - \mathbf{s}^{\text{dem}} = \text{diag}(\mathbf{v}) Y_{\text{bus}}^* \mathbf{v}^*. \quad (2.26)$$

2.3.4. Resolución del Flujo de Potencia

La subsección anterior presenta la ecuación de flujo de potencia. Como se mencionó al comienzo de la sección, el problema de flujo de potencia consiste en resolver esta ecuación para obtener la información completa de potencia y voltaje sobre cada uno de los elementos de la red. En este problema se deben especificar el perfil de demanda \mathbf{s}^{dem} , el perfil de generación en los estáticos $\mathbf{s}^{\text{gen},\text{est}}$, el perfil de generación en los compensadores de reactiva \mathbf{s}^{comp} y la potencia activa en los controladores de voltaje, $\mathbf{p}^{\text{gen},v}$. Dentro de las incógnitas se encuentran la potencia reactiva en este último tipo de generadores, $\mathbf{q}^{\text{gen},v}$, además del perfil de potencia en el generador de referencia $\mathbf{s}^{\text{gen},\text{ref}}$. Por otro lado, se debe especificar la magnitud de voltaje para las barras con generadores controladores de voltaje $\{v_{m,i}\}_{i \in \mathcal{B}_G}$ y la magnitud y ángulo en la barra de referencia. Finalmente, se tienen como incógnitas las corrientes aparentes que circulan por las líneas y transformadores, $I_{\text{line,to}}$ e $I_{\text{line,from}}$. La resolución del problema se divide en dos etapas. En primer lugar, se resuelven la magnitud y ángulo de los voltajes en todas las barras donde estos son incógnita. Luego, se infiere $\mathbf{q}^{\text{gen},v}$, $\mathbf{s}^{\text{gen},\text{ref}}$, $I_{\text{line,to}}$ e $I_{\text{line,from}}$.

Tomando el ejemplo con el que se viene trabajando, se observa que las variables de magnitud y ángulo de voltajes son 5. Las barras de carga introducen las incógnitas v_2 , v_4 , δ_2 y δ_4 , mientras que el bus de generación introduce la incógnita δ_3 . Por lo tanto, se precisan 5 ecuaciones que no introduzcan nuevas incógnitas para resolver este problema. Para esto se toman las dos ecuaciones complejas de inyección de potencia para las barras de carga, y la parte real de la ecuación de inyección de potencia en las barras generadoras. En el caso de los buses de carga, al ser una ecuación compleja, esta resulta en dos ecuaciones, igualando por un lado la parte real, y por otra la parte compleja. El sistema de ecuaciones resulta entonces:

Capítulo 2. Problema de Despacho Óptimo de Reactiva

$$0 = Re \left(v_2 \begin{bmatrix} -y_{2,1}^* & (y_{sh,2,1} + y_{2,1} + y_{sh,2,3} + y_{2,3})^* & -y_{2,3}^* \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.27)$$

$$0 = Im \left(v_2 \begin{bmatrix} -y_{2,1}^* & (y_{sh,2,1} + y_{2,1} + y_{sh,2,3} + y_{2,3})^* & -y_{2,3}^* \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.28)$$

$$-p_4^{\text{dem}} = Re \left(v_4 \begin{bmatrix} 0 & -y_{2,4}^* & 0 & (y_{sh,2,4} + y_{2,4})^* \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.29)$$

$$-q_4^{\text{dem}} = Im \left(v_4 \begin{bmatrix} 0 & -y_{2,4}^* & 0 & (y_{sh,2,4} + y_{2,4})^* \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.30)$$

$$p_3^{\text{gen},v} - p_3^{\text{dem}} = Re \left(v_3 \begin{bmatrix} 0 & -y_{2,3}^* & (y_{sh,2,3} + y_{2,3})^* & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right). \quad (2.31)$$

Se remarca que estas 5 ecuaciones no agregan incógnitas más allá de los voltajes y ángulos mencionados previamente. Los buses 2 y 4 son buses *PQ*, por lo que su potencia activa y reactiva inyectada son conocidas. Por otra parte, el bus 3 es *PV*, por lo que su potencia activa es conocida. De esta forma, se tiene un sistema de 5 ecuaciones y 5 incógnitas. Sin embargo, su resolución no es trivial dado que el sistema es no lineal, pues presenta tanto expresiones cuadráticas como senos y cosenos. Esto evita la posibilidad de resolver el sistema de forma algebraica e implica la necesidad de utilizar un método iterativo que converja a la solución. Existen diferentes algoritmos que pueden ser utilizados para resolver el problema. En esta tesis se emplea el utilizado habitualmente en la literatura: el algoritmo Newton-Raphson [18]. En particular, se utiliza una implementación ya prevista por el mismo software usado para modelar la red. En la siguiente subsección se ahonda sobre esto.

Una vez resueltos los voltajes y ángulos en todas las barras, inferir el resto de las incógnitas consiste simplemente en evaluar el resto de las ecuaciones de flujo de potencia. Esto es:

2.4. Problema de Despacho Óptimo de Reactiva

$$q_3^{\text{gen},v} - q_3^{\text{dem}} = \text{Re} \left(v_3 \begin{bmatrix} 0 & -y_{2,3}^* & (y_{\text{sh},2,3} + y_{2,3})^* & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.32)$$

$$p_1^{\text{gen,ext}} = \text{Re} \left(v_1 \begin{bmatrix} (y_{\text{sh},1,2} + y_{2,1})^* & -y_{2,1}^* & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.33)$$

$$q_1^{\text{gen,ext}} = \text{Im} \left(v_1 \begin{bmatrix} (y_{\text{sh},1,2} + y_{2,1})^* & -y_{2,1}^* & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \end{bmatrix} \right), \quad (2.34)$$

mientras que la corriente por las líneas se obtiene evaluando la Ecuación 2.9. Este procedimiento se extiende análogamente para una red genérica. Utilizando un modelo con B_G barras generadoras, B_C barras de carga y 1 barra de referencia, se tienen un total de $2B_C + B_G$ incógnitas respecto a magnitud y ángulo de voltajes. Tomando por una lado la parte real e imaginaria de las ecuaciones de inyección de potencia de las barras de carga y por otro la parte real de las ecuaciones de inyección de potencia de las barras generadoras, se obtienen $2B_C + B_G$ ecuaciones. Utilizando el algoritmo de Newton-Rapshon, se resuelven los voltajes en la red. La potencia reactiva inyectada por los generadores se resuelve con la parte imaginaria de la ecuación correspondiente, mientras que la potencia aparente en la barra de referencia se obtiene con la ecuación de inyección de potencia de esta barra. Finalmente, la corriente por las líneas se obtiene evaluando en la ecuación de flujo de línea.

2.4. Problema de Despacho Óptimo de Reactiva

En la introducción, se adelanta que el problema a resolver es el despacho óptimo de reactiva. Se discute su importancia para mantener la red en niveles seguros y se explica que, para lograrlo, los operarios cuentan con la capacidad de controlar tanto los generadores de voltaje controlado como los compensadores de reactiva. Además, se menciona que resolver este problema no solo es complicado, sino que también debe hacerse reiteradamente, ya que los valores óptimos cambian con la demanda y la generación activa en un momento particular.

En esta sección se formaliza el problema matemáticamente. Se definen las variables que forman parte de este, separándolas en tres categorías. Se detalla cada una de las restricciones que impone la red sobre estas variables, de forma de asegurar que la solución obtenida sea factible en la realidad. También se define una función objetivo que se utiliza como métrica, comparando la calidad de diferentes soluciones posibles del problema. Una vez definidos los componentes, se formula el problema completo, explicándose la dificultad de resolverlo. Finalmente, se

Capítulo 2. Problema de Despacho Óptimo de Reactiva

presenta la forma de resolución estándar utilizada, la cual sirve como punto de comparación ante la estrategia de resolución propuesta en este trabajo.

2.4.1. Variables del Problema

El conjunto de parámetros para este problema es igual a los presentados en la sección anterior. Previamente se diferencian dos conjuntos de atributos, aquellos que se especifican y aquellos que son incógnitas. Para este nuevo problema, el conjunto de atributos especificados va a ser dividido en dos. Una parte, va a pasar a ser variables, que se las llamará variables de control, mientras que el resto serán los parámetros de entrada.

Las variables de control son aquellas que el operario de la red puede modificar de forma de controlar los voltajes y corrientes en la red. Como ya fue adelantado, estos son los voltajes de referencia en los generadores controladores, $\{v_{m,i}\}_{\forall i \in \mathcal{B}_G}$ y la potencia reactiva generada en los compensadores, \mathbf{q}^{comp} . Por otro lado, los parámetros de entrada son aquellos que definen el problema. Pueden variar con el tiempo, resultando así en una instancia distinta del problema, con una solución distinta. Estos son la potencia aparente en las cargas, \mathbf{s}^{dem} , la potencia aparente en los generadores estáticos, $\mathbf{s}^{\text{gen,est}}$, y la potencia activa en los generadores controladores de voltaje, $\mathbf{p}^{\text{gen},v}$.

Los parámetros que previamente eran incógnitas serán llamados variables implícitas, para diferenciarlos de las variables de control. Estas corresponden a las magnitudes y ángulos de los voltajes para las barras donde estos valores no se conocen, $\{v_{m,i}\}_{\forall i \in \mathcal{B}_C}$ y $\{\delta_i\}_{\forall i \in \mathcal{B}_G \cup \mathcal{B}_C}$, la potencia inyectada o consumida por la conexión con la red externa $\mathbf{s}^{\text{gen,ref}}$, la potencia reactiva generada por los generadores controladores de voltaje $\mathbf{q}^{\text{gen},v}$ y las corrientes por las líneas $I_{\text{line,to}}$ e $I_{\text{line,from}}$. Se llaman variables implícitas ya que el operario no puede definir un valor de forma directa sobre estas, sino que a partir de modificar las variables de control los operarios afectan sus valores. En resumen, los parámetros del problema son:

- Parámetros de entrada: \mathbf{s}^{dem} , $\mathbf{s}^{\text{gen,est}}$, $\mathbf{p}^{\text{gen},v}$
- Variables de control: $\{v_{m,i}\}_{\forall i \in \mathcal{B}_G}$, \mathbf{q}^{comp}
- Variables implícitas: $\{v_{m,i}\}_{\forall i \in \mathcal{B}_C}$, $\{\delta_i\}_{\forall i \in \mathcal{B}_G \cup \mathcal{B}_C}$, $\mathbf{s}^{\text{gen,ref}}$, $\mathbf{q}^{\text{gen},v}$, $I_{\text{line,to}}$, $I_{\text{line,from}}$

Se remarca que a pesar de llamarse variables implícitas, esto no implica que no sean importantes, ya que el problema consiste en, dado los parámetros de entrada, encontrar el conjunto de variables de control que resulte en el mejor conjunto de variables implícitas.

2.4.2. Restricciones del Problema

Las restricciones del problema consisten en las limitaciones físicas que imponen los componentes de la red eléctrica sobre los valores que pueden tomar las variables. La primera restricción que se debe cumplir es la ecuación de flujo de potencia,

2.4. Problema de Despacho Óptimo de Reactiva

definida en la sección anterior. Esta establece la relación existente entre los voltajes y las potencias inyectadas, de forma que toda solución propuesta debe cumplirla.

$$\mathbf{s}^{\text{gen},v} + \mathbf{s}^{\text{gen},\text{est}} + \mathbf{s}^{\text{gen},\text{ref}} + \mathbf{s}^{\text{comp}} - \mathbf{s}^{\text{dem}} = \text{diag}(\mathbf{v})Y_{\text{bus}}^* \mathbf{v}^* . \quad (2.35)$$

Luego, se tienen las restricciones a nivel de barras. A estas se les limita tanto la magnitud como los ángulos del voltaje:

$$v_{\min,i} \leq v_{m,i} \leq v_{\max,i} \quad \forall i \in \mathcal{B} , \quad (2.36)$$

$$\delta_{\min,i} \leq \delta_i \leq \delta_{\max,i} \quad \forall i \in \mathcal{B} . \quad (2.37)$$

En general, se asume que $\delta_{\min,i} = -\pi$ y $\delta_{\max,i} = \pi$.

Por otro lado, existe un límite físico para la corriente que una línea o transformador puede soportar de manera sostenida. Esta condición debe cumplirse para la corriente circulando en ambos sentidos. Se expresa mediante las siguientes desigualdades:

$$|I_{i \rightarrow j}| \leq I_{i \rightarrow j, \max} \quad \forall (i, j) \in \mathcal{L} \cup \mathcal{T} , \quad (2.38)$$

$$|I_{j \rightarrow i}| \leq I_{j \rightarrow i, \max} \quad \forall (i, j) \in \mathcal{L} \cup \mathcal{T} , \quad (2.39)$$

Para evitar tener que agregar un nuevo conjunto de variables $I_{\text{line,to}}$, junto con la ecuación de igualdad que vincula estas variables con los voltajes, $I_{\text{line,to}} = Y_{\text{line,to}} \mathbf{v}$, se utiliza directamente esta igualdad en la restricción. Lo mismo para $I_{\text{line,from}}$. Esto es:

$$|Y_{\text{line,to}} \mathbf{v}| \leq I_{i \rightarrow j, \max} \quad \forall (i, j) \in \mathcal{L} \cup \mathcal{T} , \quad (2.40)$$

$$|Y_{\text{line,from}} \mathbf{v}| \leq I_{j \rightarrow i, \max} \quad \forall (i, j) \in \mathcal{L} \cup \mathcal{T} , \quad (2.41)$$

Este detalle de implementación resulta en el mismo problema, pero facilita al optimizador al simplificar el conjunto de variables y restricciones efectivo. Respecto a las potencias, se tienen restricciones en la capacidad de producir reactiva para los generadores controladores de voltaje y para los compensadores de reactiva:

$$q_{i,\min}^{\text{gen},v} \leq q_i^{\text{gen},v} \leq q_{i,\max}^{\text{gen},v} \quad \forall i \in \mathcal{G} , \quad (2.42)$$

$$q_{i,\min}^{\text{comp}} \leq q_i^{\text{comp}} \leq q_{i,\max}^{\text{comp}} \quad \forall i \in \mathcal{R} . \quad (2.43)$$

Por último, existen restricciones sobre la potencia activa y reactiva del generador de referencia.

$$p_{i,\min}^{\text{gen},\text{ref}} \leq p_i^{\text{gen},\text{ref}} \leq p_{i,\max}^{\text{gen},\text{ref}} \quad \forall i \in \mathcal{X} , \quad (2.44)$$

$$q_{i,\min}^{\text{gen},\text{ref}} \leq q_i^{\text{gen},\text{ref}} \leq q_{i,\max}^{\text{gen},\text{ref}} \quad \forall i \in \mathcal{X} . \quad (2.45)$$

Capítulo 2. Problema de Despacho Óptimo de Reactiva

2.4.3. Función Objetivo del Problema

Las anteriores son todas las restricciones que una posible solución debe cumplir para considerarse factible. Aunque parezcan bastante restrictivas, en general un caso de estudio tiene múltiples soluciones factibles. Resolver el problema implica obtener la mejor de todas estas, para lo cual es necesario cuantificar la calidad de las mismas. Tal como fue introducido previamente, la idea es minimizar las pérdidas de potencia activa en las líneas. Por lo tanto, la función objetivo resulta:

$$\sum_{(i,j) \in \mathcal{L}} Re(\mathbf{s}_{i \rightarrow j} + \mathbf{s}_{j \rightarrow i}) \quad . \quad (2.46)$$

Esta corresponde a la suma para todas las líneas de:

$$\mathbf{s}_{i \rightarrow j} + \mathbf{s}_{j \rightarrow i} = \mathbf{s}_{i \rightarrow j} - (-\mathbf{s}_{j \rightarrow i}) , \quad (2.47)$$

lo que es igual a la potencia que se envía de i a j , menos la potencia que recibe j de i . Esta diferencia son las pérdidas. Luego, se toma la parte real para tener en cuenta únicamente las pérdidas de potencia activa. Similar a lo que ocurre con la restricción sobre la capacidad de las corrientes, se sustituye:

$$\mathbf{s}_{i \rightarrow j} = v_i Y_{\text{line,to}} \mathbf{v} , \quad (2.48)$$

$$\mathbf{s}_{j \rightarrow i} = v_j Y_{\text{line,to}} \mathbf{v} . \quad (2.49)$$

Esto evita tener que agregar estas ecuaciones como restricciones de igualdad en el problema. De esta forma, el objetivo resulta:

$$\sum_{(i,j) \in \mathcal{L}} Re(v_i Y_{\text{line,to}} \mathbf{v} + v_j Y_{\text{line,to}} \mathbf{v}) . \quad (2.50)$$

2.4.4. Formulación Completa del Problema

Dadas la función de costo, la ecuación del flujo de carga y las restricciones de la red, se plantea la formulación del problema de despacho óptimo de reactiva:

$$\min_{\{v_{m,i}\}_{\forall i \in \mathcal{B}_G}, \mathbf{q}^{\text{comp}}} \sum_{(i,j) \in \mathcal{L}} Re(v_i Y_{\text{line,to}} \mathbf{v} + v_j Y_{\text{line,to}} \mathbf{v}) , \quad (2.51)$$

2.5. Software Utilizado

sujeto a:

$$\mathbf{s}^{\text{gen},v} + \mathbf{s}^{\text{gen},\text{est}} + \mathbf{s}^{\text{gen},\text{ref}} + \mathbf{s}^{\text{comp}} - \mathbf{s}^{\text{dem}} = \text{diag}(\mathbf{v})Y_{\text{bus}}^*\mathbf{v}^* \quad (2.52\text{a})$$

$$|Y_{\text{line,to}}\mathbf{v}| \leq I_{i \rightarrow j, \text{máx}} \quad \forall (i, j) \in \mathcal{L} \cup \mathcal{T} \quad (2.52\text{b})$$

$$|Y_{\text{line,from}}\mathbf{v}| \leq I_{j \rightarrow i, \text{máx}} \quad \forall (i, j) \in \mathcal{L} \cup \mathcal{T} \quad (2.52\text{c})$$

$$v_{\text{mín},i} \leq v_{m,i} \leq v_{\text{máx},i} \quad \forall i \in \mathcal{B} \quad (2.52\text{d})$$

$$\delta_{\text{mín},i} \leq \delta_i \leq \delta_{\text{máx},i} \quad \forall i \in \mathcal{B} \quad (2.52\text{e})$$

$$q_{i,\text{mín}}^{\text{gen},v} \leq q_i^{\text{gen},v} \leq q_{i,\text{máx}}^{\text{gen},v} \quad \forall i \in \mathcal{G} \quad (2.52\text{f})$$

$$q_{i,\text{mín}}^{\text{comp}} \leq q_i^{\text{comp}} \leq q_{i,\text{máx}}^{\text{comp}} \quad \forall i \in \mathcal{R} \quad (2.52\text{g})$$

$$p_{i,\text{mín}}^{\text{gen},\text{ref}} \leq p_i^{\text{gen},\text{ref}} \leq p_{i,\text{máx}}^{\text{gen},\text{ref}} \quad \forall i \in \mathcal{X} \quad (2.52\text{h})$$

$$q_{i,\text{mín}}^{\text{gen},\text{ref}} \leq q_i^{\text{gen},\text{ref}} \leq q_{i,\text{máx}}^{\text{gen},\text{ref}} \quad \forall i \in \mathcal{X} \quad (2.52\text{i})$$

Resolver el problema formulado es extremadamente desafiante debido a su naturaleza no lineal y no convexa. La no convexidad del problema significa que puede haber múltiples soluciones locales óptimas, lo que dificulta la identificación de la mejor solución global. La alta dimensionalidad del sistema, junto con la necesidad de soluciones en tiempo real, hace que la resolución del ORPD sea una tarea complicada y exigente en términos de recursos computacionales.

El método más frecuente para abordar estas dificultades es el optimizador de punto interno, o IPOPT por sus siglas en inglés (*Interior Point Optimizer*). Esta es una herramienta estándar en la resolución de problemas de optimización no lineal. Una de sus principales ventajas es su capacidad para encontrar soluciones factibles y óptimas en un tiempo razonable, aprovechando técnicas avanzadas de factorización y paralelización. IPOPT también permite una gran flexibilidad en la formulación del problema, facilitando la incorporación de diversas restricciones y objetivos. Su robustez y eficiencia han hecho que esta herramienta sea ampliamente adoptada en la industria y la investigación para resolver problemas como el ORPD.

No obstante, a pesar de sus numerosas ventajas, IPOPT no garantiza la convergencia en todos los casos, especialmente en problemas no convexos, como el ORPD. Incluso cuando el algoritmo converge, no se puede asegurar que la solución encontrada sea globalmente óptima. La dependencia en la elección de parámetros iniciales y la parametrización del algoritmo también pueden afectar la calidad de las soluciones obtenidas. Son estas limitaciones las razones por las que se propone explorar nuevas alternativas, como el uso de técnicas de aprendizaje automático. Estas técnicas pueden aprender patrones y aproximaciones del comportamiento del sistema, ofreciendo soluciones más rápidas y potencialmente mejorando la capacidad de encontrar soluciones más cercanas al óptimo global.

2.5. Software Utilizado

En la industria de la energía eléctrica, el modelado y la simulación de redes eléctricas, así como la ejecución de flujos de carga y la optimización del flujo de potencia, son esenciales para asegurar la operación eficiente y segura del sistema.

Capítulo 2. Problema de Despacho Óptimo de Reactiva

Para llevar a cabo estas tareas, se utilizan diversas herramientas de software especialmente diseñadas para el análisis y la planificación de sistemas eléctricos. En particular, en este trabajo se emplean dos herramientas muy populares en este campo: Power System Simulator for Engineering (PSSE) [3] y PandaPower [35]. PSSE, una solución comercial ampliamente adoptada, es utilizada por los operarios del Despacho Nacional de Carga (DNC) de Uruguay, mientras que PandaPower, una biblioteca de código abierto, ofrece una alternativa flexible y sin costo basada en Python. Ambas herramientas permiten realizar estudios detallados de flujos de carga, optimización del flujo de potencia (OPF), y el despacho óptimo de potencia reactiva (ORPD).

2.5.1. Power System Simulator for Engineering (PSSE)

PSSE es una herramienta de software paga, desarrollada por Siemens para la planificación, análisis y operación de redes eléctricas. Se destaca por su robustez y capacidad para manejar redes eléctricas de gran escala. Es ampliamente utilizado en la industria debido a su fiabilidad y la precisión de sus cálculos. PSSE permite realizar una variedad de estudios, incluyendo análisis de flujos de carga, estudios de estabilidad transitoria, análisis de contingencias y optimización del flujo de potencia.

En el DNC, se utiliza el software PSSE para modelar las redes eléctricas y permitir a los operarios realizar pruebas y ejecutar flujos de carga bajo diversos estados de la red. Esta herramienta es fundamental para simular escenarios de interés, evaluando posibles eventos que podrían ocurrir en la red o analizando el impacto de ciertos cambios planificados. La capacidad del PSSE para manejar estudios detallados y precisos de flujos de carga facilita la toma de decisiones y la planificación estratégica de la infraestructura eléctrica.

El software PSSE es pago, por lo que es necesario obtener una licencia para poder utilizarlo. Esta se obtuvo gracias a la colaboración del Departamento de Potencia del Instituto de Ingeniería Eléctrica. Con el programa, se accede a un modelo de la red eléctrica uruguaya y se realizan algunos ajustes al mismo, sobre los cuales se entra en detalle en el Capítulo 4. Sin embargo, existen ciertas desventajas que hacen que PSSE no sea la opción más adecuada para resolver el problema de despacho óptimo de reactiva en este trabajo.

En primer lugar, la licencia brindada por el DNC corresponde a una versión antigua, en donde la integración con Python requiere usar la versión 2.7 de este, lo cual dificulta y enlentece el proceso. El uso de Python es necesario para la generación de datos mediante la simulación iterativa de varios escenarios. Tampoco provee una herramienta para resolver el problema ORPD, sino que solamente se puede resolver el OPF. Considerando estas desventajas, se decidió optar por la herramienta PandaPower para realizar los flujos de carga y la resolución del despacho de reactiva. PandaPower ofrece una mayor facilidad de uso y una integración más fluida con Python, lo que permite un análisis más eficiente y accesible.

2.5. Software Utilizado

2.5.2. PandaPower

PandaPower es una herramienta de código abierto desarrollada en Python para el análisis de sistemas de energía eléctrica. Entre sus capacidades destacan la ejecución de flujos de carga y la optimización del flujo de potencia (OPF), y en particular permite correr la optimización del despacho de potencia reactiva (ORPD), lo cual es fundamental en esta investigación. Esta herramienta fue creada con el objetivo de unir las ventajas de las herramientas comerciales y las de código abierto, aunque suelen ser muy flexibles y personalizables, a menudo requieren que los usuarios parametrícen modelos básicos con conocimientos especializados. Por otra parte, las herramientas comerciales cuentan con interfaces gráficas de usuario, que en general son difíciles de automatizar y tienen posibilidades limitadas de personalización debido a su código propietario. Esta plataforma combina la librería de análisis de datos Pandas [32] con Pypower [25] y PowerModels [11], proporcionando una herramienta simple y flexible para el modelado, análisis y simulación de redes eléctricas.

En cuanto a la validación de esta herramienta comparada con otros softwares comerciales, PandaPower es considerada válida y robusta para el análisis de sistemas de potencia. Su precisión fue comprobada mediante más de 250 pruebas automáticas, comparando los resultados con herramientas comerciales como DIg-SILENT PowerFactory o PSS Sincal [30]. Además, el uso de integración continua en GitHub garantiza que el código se pruebe y valide constantemente, asegurando su calidad y fiabilidad.

PyPower es una biblioteca de Python para el análisis de flujo de potencia y la simulación de redes eléctricas. Está diseñada para resolver problemas de flujo de potencia (tanto en régimen permanente como en estado transitorio), análisis de contingencias y otras tareas de simulación de redes eléctricas. PyPower es una versión reescrita en Python de la herramienta de análisis de redes eléctricas MATPOWER, que está escrita en MATLAB.

Por otro lado, PowerModels es un paquete desarrollado en Julia para la optimización de flujos de potencia en redes eléctricas. Se utiliza principalmente para resolver problemas de optimización en sistemas de energía eléctrica, como el flujo de potencia óptimo (OPF), el despacho económico y la planificación de la expansión de la red.

En particular, en este trabajo se hace uso principalmente de dos optimizadores en PandaPower. El primero se emplea para los flujos de carga y, al ejecutar el comando correspondiente en PandaPower, este utiliza el optimizador de PyPower por atrás. El segundo se utiliza para la optimización del flujo de potencia reactiva y, al ejecutar el comando en PandaPower, se activa Julia y se utiliza PowerModels para resolver el problema.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 3

Aprendizaje Automático

Es indiscutible que la Inteligencia Artificial (IA) ha cobrado una relevancia acelerada en los últimos años, motivando innovaciones en áreas como entretenimiento, comercio, educación y más. Sin embargo, el estudio de la IA está lejos de haber nacido en los últimos años. Este campo nace en los años 60, en busca de replicar las capacidades intelectuales humanas en máquinas. Durante los más de 60 años de existencia, este campo ha tenido varios momentos altos y bajos. La principal limitación encontrada en sus comienzos era la falta de capacidad de cómputo en las maquinas, lo que impedía poner en práctica varios conceptos teóricos desarrollados. Durante los primeros años, la mayoría de las estrategias utilizadas consistían en crear sistemas que pudieran resolver problemas lógicos, como jugar al ajedrez o entender el lenguaje natural, mediante la definición de reglas programadas manualmente. Para esto, los ingenieros pasaban gran parte del tiempo diseñando algoritmos que permitieran detectar patrones, los cuales luego eran usados para resolver las tareas de interés. Esta metodología se conoce en el área como Inteligencia Artificial Clásica, o Simbólica. Sin embargo, esta metodología enfrenta dos grandes problemas. Por un lado, la dificultad de diseñar estos algoritmos de forma que logren un rendimiento robusto ante la gran varianza que presentan los escenarios reales. Por otro, la especificidad de estos algoritmos, siendo necesario diseñar patrones para cada nueva aplicación que se desee implementar [19].

Sobre principios de los años 90 es que surge un nuevo campo de estudio en la IA, el aprendizaje automático. Este se basa en la premisa de crear algoritmos diseñados no únicamente para resolver una tarea específica, sino para aprender a resolver la tarea por más que existan variaciones. Para esto, en lugar de depender de reglas predefinidas y rígidas, se recopilan datos etiquetados que representan diferentes casos y situaciones, y se utilizan para entrenar modelos que puedan hacer predicciones o clasificaciones. Estos modelos aprenden de los datos al identificar patrones subyacentes que pueden no ser obvios para los humanos, lo que les permite generalizar y adaptarse a nuevas situaciones. De esta manera, el aprendizaje automático ofrece una solución más flexible y escalable para abordar tareas complejas y variadas en numerosos campos. Aunque estas ideas nacen en los años 90, el auge no comienza hasta principios de 2010. En particular, en 2012 la red neuronal AlexNet [24] (un modelo predictor de aprendizaje automático) gana de forma

Capítulo 3. Aprendizaje Automático

contundente la prestigiosa competencia de reconocimiento visual de ImageNet [14]. La importancia de esto hito no es solo el avance en el campo de visión artificial, sino demostrar que las redes neuronales sí funcionan en la práctica. Esta competencia fue un punto de inflexión, a partir del cual el aprendizaje automático, en particular las redes neuronales, comienzan a suplantar otras estrategias en todo tipo de campos, mostrando saltos en desempeño inmensos.

En esta sección se presentan los conceptos de este campo que son necesarios entender para este trabajo [20]. Se comienza introduciendo en qué consiste esta área, junto con los paradigmas y estrategias que la constituyen. Luego, se presenta el modelo matemático de las redes neuronales y se brinda una breve explicación de cómo se entrena. Se continúa con grafos, introduciendo conceptos básicos de esta estructura, permitiendo así definir las redes neuronales para datos en grafos. Finalmente, el capítulo cierra presentando cómo estas estrategias se implementan para resolver el problema de despacho óptimo de reactiva.

3.1. Introducción

Aprender a partir de la experiencia puede sonar como algo abstracto, como si una máquina obtuviese conocimientos de forma autónoma. Sin embargo, en el contexto del aprendizaje automático, lo que realmente sucede es mucho más tangible: se tiene un modelo predictivo, o arquitectura, el cual mapea entradas a salidas. Estos modelos tienen una cantidad alta de parámetros, donde al cambiar estos se altera el mapeo entrada-salida del predictor. Luego, “aprender” consiste en correr un algoritmo que ajusta los parámetros del modelo para minimizar el error en las predicciones de los datos con los que se cuenta. Esto se repite iterativamente hasta encontrar el mejor mapeo. Así, aunque el concepto de “aprender de la experiencia” pueda parecer abstracto, se traduce en una serie de operaciones matemáticas que ajustan los parámetros de los predictores con el fin de mejorar su rendimiento y su capacidad para generalizar ante datos no vistos. El marco de aprendizaje clásico consta de los siguientes componentes principales:

- **Mapeo Objetivo Desconocido:** Corresponde al mapeo entrada-salida que se busca replicar. Matemáticamente, es un mapeo $F : \mathcal{X} \rightarrow \mathcal{Y}$, donde \mathcal{X} representa el espacio de entradas, definidos en la sección anterior, e \mathcal{Y} el espacio de salida, que corresponde al espacio de variables de control que los operarios pueden ajustar para controlar la reactiva en la red eléctrica.
- **Distribución del Conjunto de Entrada:** Corresponde a la distribución de probabilidad sobre \mathcal{X} al tomar las muestras, definida como $P(\mathbf{x})$, con $\mathbf{x} \in \mathcal{X}$. No es necesario conocer esta distribución ni estimarla, pero es crucial que los datos de entrenamiento sean representativos de las entradas que se encontrarán en inferencia. Por ejemplo, si se entrena el modelo solo con datos históricos nocturnos de la red uruguaya, su rendimiento durante el día será significativamente peor, ya que los datos de entrenamiento no serían representativos de la realidad.

3.2. Paradigmas del Aprendizaje Automático

- **Muestras de la Función Objetivo:** Corresponde a muestras de entrada y su evaluación según la función objetivo. Matemáticamente se define el conjunto de datos de entrada $X = \{\mathbf{x}_t\}_{t \in \mathcal{T}}$ y el conjunto de datos de salida $Y = \{F(\mathbf{x}_t)\}_{t \in \mathcal{T}}$, donde \mathcal{T} es el conjunto que indexa los datos. Se utiliza la notación \mathcal{T} ya que los datos a utilizar en este trabajo están indexados temporalmente. Luego, el conjunto de datos de entrenamiento queda definido como $\mathcal{D} = \{(\mathbf{x}_t, F(\mathbf{x}_t))\}_{t \in \mathcal{T}}$, con las muestras del conjunto X obtenidas de su distribución de probabilidad P . La hipótesis principal de este paradigma es que no se puede evaluar directamente el mapeo, sino que solo se tiene acceso a muestras de este.
- **Conjunto de Hipótesis de Funciones:** Corresponde a una familia de funciones paramétricas $g : \mathcal{X}, \Theta \rightarrow \mathcal{Y}$, donde Θ representa el espacio de parámetros. Se busca una instancia $\theta \in \Theta$ que resulte en la función $g(\mathbf{x}; \theta)$ que mejor aproxime a $F(\mathbf{x})$. En el aprendizaje profundo, se utilizan aproximadores casi universales, como las redes neuronales, que pueden aproximar cualquier función continua.
- **Algoritmo de Aprendizaje:** Es el método mediante el cual se obtienen los parámetros óptimos. Este proceso consiste en inicializar aleatoriamente los parámetros e ir actualizándolos iterativamente hasta aproximar F lo mejor posible. Se utiliza una función de pérdida para medir la diferencia entre la salida del predictor y su etiqueta, y se ajustan los parámetros en sentido opuesto al gradiente de esta función de pérdida.
- **Función Predictora Final:** Es el predictor entrenado $g(\mathbf{x}; \theta^*)$, siendo θ^* el conjunto de parámetros obtenido al concluir el entrenamiento.

En la Figura 3.1 se puede ver un esquema sobre el flujo de entrenamiento para un algoritmo de aprendizaje automático.

3.2. Paradigmas del Aprendizaje Automático

El aprendizaje automático se basa en la premisa fundamental de aprender a partir de datos. No obstante, esta definición es bastante amplia, lo que dificulta agrupar todas las estrategias en un único marco de referencia. Para abordar esta diversidad, la literatura clasifica estas estrategias en tres paradigmas principales: aprendizaje supervisado, no supervisado y por refuerzo. Hasta ahora, se ha centrado la atención en las técnicas de aprendizaje supervisado, que es el paradigma más estudiado. Sin embargo, este no es el único enfoque relevante. En este trabajo, se emplean tanto estrategias supervisadas como no supervisadas, las cuales se describen a continuación.

3.2.1. Aprendizaje Supervisado

El aprendizaje supervisado es un paradigma del aprendizaje automático que se basa en el uso de un conjunto de datos etiquetados, donde cada dato de entrada

Capítulo 3. Aprendizaje Automático

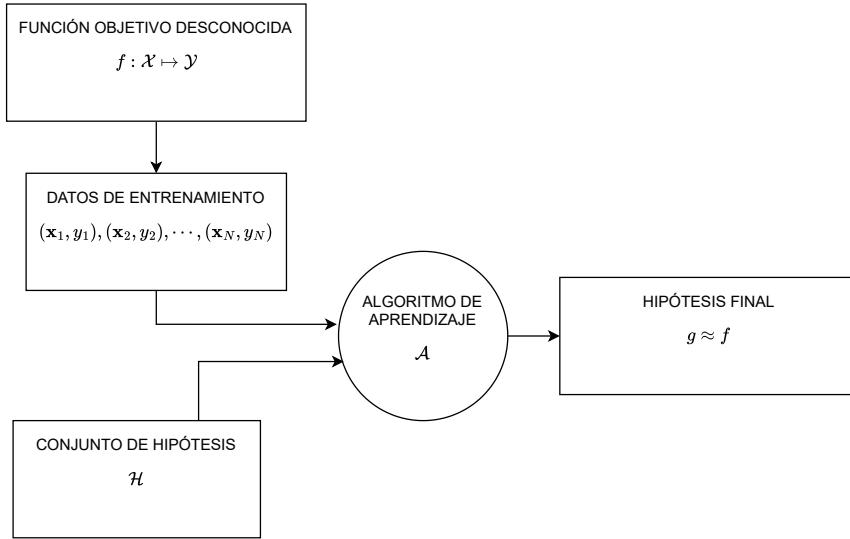


Figura 3.1: Esquema del flujo de entrenamiento de un algoritmo de aprendizaje automático.

tiene asociado una salida correcta o etiqueta. En este sentido, un modelo o predictor es entrenado para aprender una función que mapea las entradas a las salidas, buscando replicar el comportamiento del “supervisor”.

En esta tesis, se utiliza un optimizador de punto interior (IPOPT) proporcionado por la librería PandaPower de Python para obtener las variables de control óptimas dadas las condiciones de la red eléctrica. Posteriormente, se entrena un modelo con el objetivo de aprender a replicar el comportamiento del mismo. Todo esto será detallado más adelante. El objetivo es que el modelo prediga las salidas que el IPOPT generaría, no solo para los estados de la red en los que fue entrenado, sino también para nuevos datos no vistos durante el entrenamiento.

Es importante destacar que el modelo entrenado en este proceso no tendrá conocimiento directo de las ecuaciones físicas que rigen en el funcionamiento de una red eléctrica ni de cómo encontrar su óptimo. En lugar de ello, el modelo aprenderá de los patrones presentes en los datos, los cuales fueron generados por herramientas que sí consideran la física de la red. La idea final es obtener un predictor que sea más rápido y de calidad similar a la del optimizador.

Sin embargo, esta estrategia presenta el problema que la calidad de las soluciones que el modelo puede generar está limitada a la calidad de las soluciones proporcionadas por el optimizador. Como se ha discutido anteriormente, los optimizadores de problemas no convexos, como el IPOPT, pueden proporcionar resultados buenos, pero debido a la naturaleza del problema es posible que existan soluciones mejores. Por esta razón, también se propone explorar otra rama del aprendizaje, en busca de resultados desvinculados del optimizador clásico.

3.2.2. Aprendizaje No Supervisado

El aprendizaje no supervisado es otro paradigma del aprendizaje automático que se distingue por trabajar con datos sin etiquetas, es decir, sin una salida correcta predefinida para cada entrada. Usualmente se utiliza en tareas como identificar patrones, estructuras ocultas o agrupamientos en los datos. De esta manera, el modelo explora las características intrínsecas de los datos y trata de descubrir relaciones subyacentes sin guía explícita.

En el contexto de esta tesis, la estrategia de aprendizaje no supervisado a utilizar consiste en encontrar soluciones factibles del problema de optimización que resulten en valores objetivos bajos. Para esto, se define una función de pérdida fuertemente relacionada con la física del problema, la cual se compone de un término proporcional a la métrica, y otro que penaliza soluciones que no sean factibles.

Es claro que esta mirada al problema es más desafiante que utilizando aprendizaje supervisado. Uno de los principales desafíos es garantizar que las soluciones propuestas no solo sean óptimas en términos de reducción de pérdidas en las líneas, sino que también cumplan con todas las restricciones operativas y de seguridad de la red eléctrica, como los límites de voltaje y capacidad de las líneas. En otras palabras, se debe entrenar un predictor cuyas salidas cumplan lo mejor posible la formulación matemática del ORPD de la Ecuación 2.51.

3.3. Redes Neuronales

Durante la introducción de esta sección se explica que el concepto de aprender consiste en encontrar, dentro de una familia paramétrica de funciones, el conjunto de parámetros que resulte en el mejor predictor posible. En esta sección, se presenta la familia de funciones que mayor popularidad ha presentado en los últimos años: las redes neuronales. Su nombre se debe a que se inspiran en el modelo del funcionamiento del cerebro humano, consistiendo en una gran cantidad de “neuronas” interconectadas entre sí. Cada neurona consiste en una unidad de procesamiento de información, la cual toma como entrada la señal de salida de las neuronas de la capa anterior, realiza una combinación lineal de estas y las pasa por una función no lineal, también conocida como función de activación. Los parámetros de esta familia son el conjunto de pesos utilizados por cada una de las neuronas al realizar la combinación lineal de sus entradas.

3.3.1. Formulación Matemática

El perceptrón consiste en un modelo simplificado de cómo funciona una neurona. Este toma un vector de entrada, $\mathbf{x} \in \mathbb{R}^d$, y produce una salida escalar, $y \in \mathbb{R}$. En esta transformación, la neurona realiza una combinación lineal de la entrada, a partir de multiplicar por su vector de pesos $\mathbf{w} \in \mathbb{R}^d$ y sumarle una constante $b \in \mathbb{R}$. Luego, se aplica la función de activación $f : \mathbb{R} \rightarrow \mathbb{R}$. La elección de función de activación es parte de los parámetros a elegir durante el entrenamiento, pero entre las funciones típicas se encuentra la función sigmoide, la tangente hiperbólica

Capítulo 3. Aprendizaje Automático

(\tanh) y la unidad lineal rectificada ($ReLU$). Una posible representación se puede observar en la Figura 3.2. Matemáticamente resulta:

$$y = f(\mathbf{w}^T \mathbf{x} + \mathbf{b}) . \quad (3.1)$$

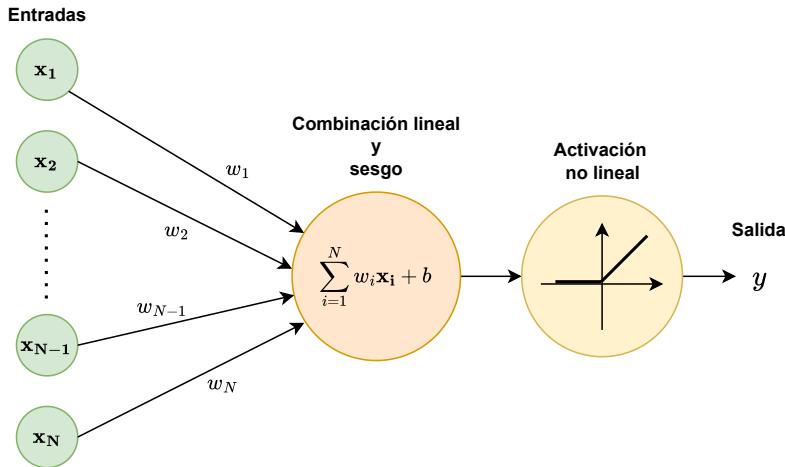


Figura 3.2: Esquema de un perceptrón.

Estos perceptrones se pueden apilar para formar una capa de d_{out} neuronas. Cada una de estas toma como entrada el mismo vector $\mathbf{x} \in \mathbb{R}^{d_{in}}$, el cual representa la salida de la capa anterior, y aplica la expresión en Ecuación 3.1. De esta forma, la salida $\mathbf{y} \in \mathbb{R}^{d_{out}}$ queda:

$$\mathbf{y} = f(\mathbf{W}^T \mathbf{x} + \mathbf{b}) , \quad (3.2)$$

donde la matriz de pesos de la capa $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$ consiste en apilar los vectores \mathbf{w}^T de cada neurona, y el vector de constantes \mathbf{b} resulta de concatenar las constantes de cada neurona. Se remarca que se utiliza un abuso de notación al aplicar la función f a un vector. En este caso, se aplica la función, que tiene como dominio los reales, a cada entrada del vector por separado. Es por esto que se dice que la función de activación es no lineal y punto a punto, esto último haciendo referencia a cómo actúa la función sobre un vector.

Finalmente, para obtener la red neuronal, se apilan L capas de manera secuencial, donde la salida de una capa corresponde a la entrada de la siguiente. La expresión matemática recursiva de la red queda:

$$\mathbf{x}^{(l)} = f(\mathbf{W}^{T,(l-1)} \mathbf{x}^{(l-1)} + \mathbf{b}^{(l-1)}) , \quad (3.3)$$

donde l indexa el numero de capa. Luego, la red neuronal queda definida a partir de seleccionar la cantidad de capas L , la cantidad de neuronas por capa, también llamado dimensión de la capa, d^l , y la función de activación f . Por último, se define la función $\Phi(\mathbf{x}; \theta)$, la cual representa el mapeo total que realiza la red

3.3. Redes Neuronales

neuronal sobre la entrada \mathbf{x} a través de todas las capas. Aquí, θ denota el vector de parámetros, que incluye los pesos y sesgos de todas las capas, mientras que \mathbf{x} es la entrada inicial a la red.

Es común separar las capas de una red en tres: la capa de entrada, la capa de salida y las capas intermedias, u ocultas. En la Figura 3.3 se observa un esquema de su interconexión. A continuación se hace una descripción de cada una de ellas:

- **Capa de Entrada:** Es la primer capa de la red, y se caracteriza por recibir los datos de entrada, correspondiendo una neurona por características del conjunto de datos de entrada. En el caso del problema de este trabajo, la red neuronal toma como entrada la potencia activa y reactiva demandada en cada barra, la potencia activa en el generador controlador de voltaje y la potencia activa y reactiva generada en el generador estático. Por lo tanto, la dimensión en la capa de entrada está dada y se corresponde con los atributos de entrada al problema.
- **Capa de Salida:** Esta capa es la última de la red, y su salida representa las predicciones finales del modelo. El número de neuronas en esta capa se corresponde con la cantidad de atributos esperados a la salida en el problema. En este caso se corresponde con las variables a controlar, es decir, la reactiva en los compensadores, y los voltajes en los generadores.
- **Capas Ocultas:** Estas capas permiten aumentar la expresividad de las redes y son las responsables de la alta capacidad de adaptarse a distintos mapeos. Las redes neuronales tiene la propiedad de ser aproximadores universales. Esto se formaliza con el teorema de aproximación universal de las redes neuronales completamente conectadas, que establece que una red neuronal con una única capa oculta puede aproximar a cualquier función F continua, si se utiliza una cantidad suficiente de neuronas. Aunque teóricamente con una sola capa oculta es suficiente, es estándar utilizar varias capas, ya que con una sola la cantidad de neuronas necesarias aumenta exponencialmente con la complejidad de la función a copiar.

3.3.2. Entrenamiento de Redes Neuronales

Esta sección introduce el método por el cual las redes neuronales actualizan sus parámetros mediante el entrenamiento. El objetivo de entrenar un predictor a partir de datos es lograr un desempeño bueno, tanto en el conjunto de datos utilizado al entrenar como en nuevos datos que se presentan en producción. Para asegurar que esto ocurra, se divide el conjunto de datos en tres subconjuntos: entrenamiento, validación y test. El primero consiste en los datos utilizados para ajustar los pesos del modelo durante la fase de entrenamiento. El segundo es un conjunto que se utiliza como estimativo parcial del desempeño del predictor frente a datos nuevos durante el proceso de aprendizaje. Este conjunto sirve para tomar decisiones sobre propiedades del entrenamiento. Aunque este conjunto no se utiliza directamente para actualizar los pesos de la red, se toman decisiones durante el

Capítulo 3. Aprendizaje Automático

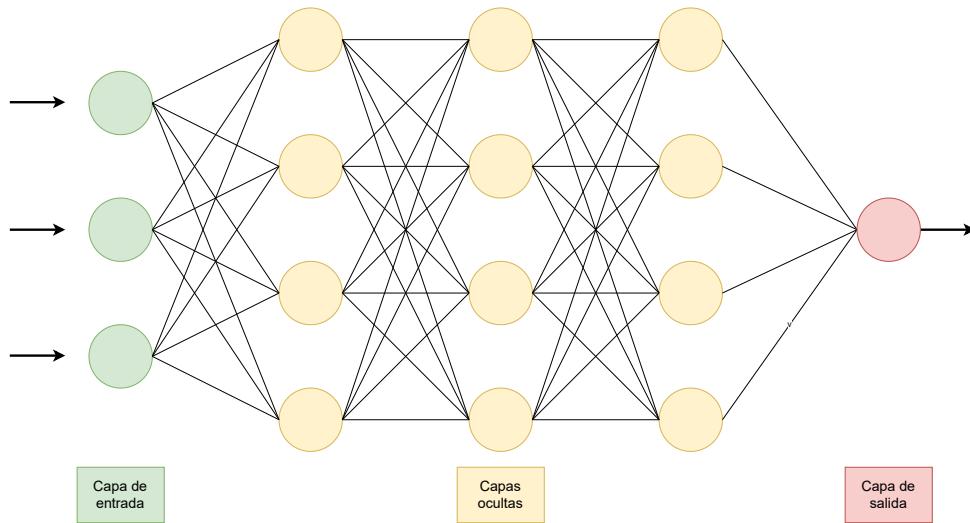


Figura 3.3: Esquema de interconexión de las distintas capas en una red neuronal.

entrenamiento en función de resultados sobre este. Por lo tanto, estos resultados se consideran sesgados y no son un indicativo final del desempeño en producción. Para esto, se tiene el conjunto de test, el cual no se utiliza en ningún momento de la fase de entrenamiento. Este sólo se utiliza para evaluar el desempeño del modelo final y su resultado se utiliza como indicador del desempeño del modelo en producción.

Una vez definidos los conjuntos de entrenamiento, validación y test se procede a explicar el proceso de entrenamiento de una red neuronal. Este es el proceso mediante el cual se ajustan los pesos de la red de forma de llegar a un predictor lo más parecido al mapeo a aprender posible. A continuación, se describen los pasos en el entrenamiento de una red neuronal:

- **Propagación Hacia Adelante (*Forward Propagation*)**

En la propagación hacia adelante, los datos de entrada se pasan a través de la red, capa por capa, hasta obtener una salida en la capa final. En cada capa, los datos se transforman mediante una combinación lineal seguida de una función de activación no lineal. Esta salida se compara con el valor real o esperado, utilizando una función de costo o pérdida que mide la diferencia entre la predicción de la red y el valor verdadero. En un problema de regresión, una función de pérdida común es el error cuadrático medio (MSE), por ejemplo.

- **Cálculo de la Pérdida**

La función de pérdida (o costo) cuantifica el desempeño de las salidas propuestas por el predictor para todo el conjunto de datos. De forma genérica, se define como:

$$\frac{1}{m} \sum_{t=1}^m \mathcal{L}(\mathbf{y}_t, \Phi(\mathbf{x}_t; \theta)) , \quad (3.4)$$

donde m es el número total de datos y $\mathcal{L}(\cdot)$ es una función de pérdida proporcional al desempeño del predictor. Por lo tanto, la función de pérdida consiste en aplicar una función de costo a cada uno de los datos y luego tomar el promedio.

- **Propagación Hacia Atrás (*Backward Propagation*)**

Después de calcular la pérdida, se utiliza la propagación hacia atrás para actualizar los pesos y sesgos de la red. Este proceso implica calcular el gradiente de la función de pérdida con respecto a cada parámetro de la red. El cálculo del gradiente se realiza utilizando la regla de la cadena, ya que cada neurona depende de las neuronas de las capas anteriores.

El algoritmo de backpropagation sigue la siguiente serie de pasos esenciales. Primero se calcula la función de pérdida L a la salida del modelo, que determina cómo cambia el error con respecto a la salida del predictor. Luego, se propagan los gradientes hacia atrás a través del modelo, calculando los gradientes para cada capa oculta. Finalmente, se ajustan los pesos usando los gradientes calculados. Existen distintos métodos para actualizar los gradientes, el más sencillo se realiza utilizando un paso de aprendizaje η , que determina el tamaño de la actualización.

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(l)}} ,$$

$$\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \eta \frac{\partial L(\mathbf{W})}{\partial \mathbf{b}^{(l)}} .$$

Existen otros métodos de optimización además del clásico gradiente descendente, como lo son Adam, RMSprop y Momentum, que pueden acelerar la convergencia del modelo y mejorar la estabilidad del entrenamiento.

- **Iteraciones y Épocas**

El proceso de propagar hacia adelante y hacia atrás se repite para cada lote de datos de entrenamiento, y una vez que todos los datos han sido procesados, se completa una época. El entrenamiento suele involucrar múltiples épocas, durante las cuales la red mejora sus predicciones ajustando sus parámetros.

Al final de cada época, se evalúa el desempeño del modelo utilizando tanto el subconjunto de entrenamiento como el de validación. Para esto, se utilizan métricas que cuantifican la precisión del modelo, proporcionando una visión completa de cómo está funcionando el mismo. Los resultados de estas evaluaciones se registran a lo largo de las épocas, facilitando información valiosa sobre el proceso de entrenamiento. Idealmente, el desempeño del modelo en

Capítulo 3. Aprendizaje Automático

el conjunto de validación debería ser similar al obtenido en el conjunto de entrenamiento, aunque es común que sea ligeramente inferior. Si se observa una diferencia significativa, donde el desempeño en validación es considerablemente peor, se dice que el modelo está sobreajustado. Esto significa que el modelo aprendió demasiado los detalles de los datos de entrenamiento, hasta el punto de no poder generalizar adecuadamente a datos no vistos. En tales casos, es necesario realizar ajustes en los hiperparámetros del modelo para mejorar su capacidad de generalización.

■ Ajuste de Hiperparámetros

Durante el entrenamiento y evaluación de modelos, es crucial realizar ajustes en los hiperparámetros para mejorar la calidad del modelo y obtener un predictor más preciso. Entre los hiperparámetros más importantes están el paso de aprendizaje, el tamaño del lote y el número de épocas de entrenamiento. Además, se puede modificar la arquitectura del modelo, como la cantidad de capas ocultas, el número de neuronas en cada capa y el tipo de funciones de activación utilizadas. También se pueden aplicar diferentes métodos de optimización, como fue mencionado anteriormente, o utilizar técnicas de regularización. La regularización es una técnica esencial para prevenir el sobreajuste, donde el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a datos nuevos. Las técnicas de regularización incluyen la regularización L2, que penaliza grandes pesos en la red, y el dropout, que consiste en desactivar aleatoriamente neuronas durante el entrenamiento.

En el Capítulo 5 se detallan a fondo todos los hiperparámetros utilizados para entrenar los modelos y la forma de variar los mismos para obtener un modelo lo más preciso posible.

3.4. Redes Neuronales sobre Grafos

Las redes neuronales en grafos (o GNN por sus siglas en inglés, Graph Neural Networks) son una extensión de las redes neuronales tradicionales que operan directamente sobre grafos. Estas redes están diseñadas para capturar la estructura y las características inherentes de los grafos, aprendiendo representaciones de los nodos, las aristas y, en general, del grafo completo.

Aunque las redes neuronales completamente conectadas pueden aproximar cualquier función, la cantidad de parámetros crece exponencialmente con la complejidad de dicha función. Esto hace que el entrenamiento sea costoso tanto en términos de tiempo como de recursos computacionales. Además, a medida que la red se vuelve más expresiva, se necesita una mayor cantidad de datos para lograr que la red neuronal logre aprender de manera efectiva.

Otra desventaja significativa de las FCNN es su falta de adaptabilidad. Una vez que una red neuronal ha sido creada para una red eléctrica no es posible realizar cambios en esta sin volver a entrenar el modelo de cero. Esto se debe a que la

3.4. Redes Neuronales sobre Grafos

estructura de entrada de las FCNN está fija y no puede manejar cambios en la topología de la red o el número de barras sin una reconfiguración completa del modelo.

En contraste, una GNN está diseñada para trabajar con datos estructurados de acuerdo a la topología de la red eléctrica. Las GNN pueden aprovechar esta estructura y adaptarse a cambios en la red sin necesidad de reentrenar desde cero. Esto las hace especialmente adecuadas para problemas como el ORPD, donde la relación entre las barras y sus conexiones es fundamental para una predicción precisa y eficiente.

Para presentar el modelo de arquitecturas de estas redes, primero se hace una presentación acerca de grafos, para luego definir qué constituye una señal sobre un grafo y qué tipos de operaciones es posible aplicarles. Se define la operación de convolución en grafos, la cual va a suplantar la combinación lineal utilizada en las redes neuronales completamente conectadas. De esta forma, se extiende el concepto de redes neuronales a datos en grafos. Luego se estudian ciertas propiedades de esta operación que la red neuronal hereda. Esto permite realizar afirmaciones sobre cómo cambios en la topología del grafo afectan al predictor.

3.4.1. Grafos

La solución de utilizar redes neuronales completamente conectadas (o FCNN por sus siglas en inglés, Fully Connected Neural Networks) para predecir el ORPD de una red eléctrica, aunque puede ser efectiva, presenta varias desventajas. La principal es que no aprovecha la estructura inherente de la red eléctrica. En una red eléctrica, las barras están interconectadas de una manera particular, donde la propagación de los cambios en los atributos de las barras está fuertemente ligada a esta interconexión. De manera simple, cambios en la demanda de una barra afectan directamente a sus barras vecinas, mientras que las barras más alejadas no se ven tan afectadas. Esta dependencia espacial no es considerada por las FCNN, por lo que utilizar arquitecturas que sí la tengan en cuenta resulta un punto interesante. En esta sección se comienza realizando una introducción matemática a los grafos, junto con diferentes metodologías de análisis que permiten comprender la estructura y comportamiento de los mismos. Esto sirve de punto de partida para introducir en la próxima sección las redes neuronales para datos en grafos.

Los grafos son estructuras matemáticas utilizadas para modelar relaciones entre objetos. Formalmente, un grafo \mathcal{G} se define como un par $(\mathcal{V}, \mathcal{E})$, donde $\mathcal{V} = \{v_0, v_1, \dots, v_n\}$ es un conjunto de vértices o nodos, y $\mathcal{E} = \{e_0, e_1, \dots, e_l\}$ es un conjunto de aristas o enlaces que conectan pares de vértices. Para el grafo que modela la red eléctrica, el conjunto de vértices se corresponde con el conjunto de barras \mathcal{B} , mientras que el conjunto de aristas \mathcal{E} se corresponde al conjunto de líneas y transformadores $\mathcal{L} \cup \mathcal{T}$. Existen diversos tipos de clasificación en grafos que son útiles para modelar grafos en diferentes contextos. Se procede a introducir los más relevantes:

- **Simple/Múltiple:** Un grafo simple es aquel que no tiene aristas múltiples ni lazos. Es decir, entre cualquier par de vértices puede haber, como mucho,

Capítulo 3. Aprendizaje Automático

una arista y no existen aristas que conecten un vértice consigo mismo. En cambio, un multigrafo permite la existencia de múltiples aristas entre el mismo par de nodos o aristas de un nodo con sí mismo. Para las redes eléctricas se tiene un grafo simple, ya que no existen líneas entre una barra consigo misma. Además, aunque puedan existir múltiples líneas entre barras, estas usualmente se modelan como una única línea con la admitancia dividida entre dos.

- **Con Pesos/Sin Pesos:** Otra clasificación posible se basa en la existencia de pesos sobre las aristas. Un grafo ponderado es aquel en el que a cada arista se le asigna un peso, que puede representar diversos factores como costo, distancia, tiempo, entre otros. En un grafo no ponderado, todas las aristas se consideran equivalentes. En el caso de la red eléctrica uruguaya, se propone una posible manera de ponderar las aristas. El objetivo es que nodos cercanos se encuentren más conectados que nodos lejanos. Esto se puede ver reflejado por la caída de voltaje que tienden a tener dos nodos. Cuanto mayor sea esta caída, mayor será la distancia y por tanto menor debería ser el peso sea en sus aristas. Esta diferencia de voltajes se ve influenciada por la impedancia de las líneas y la potencia máxima que puede circular por ellas. Por lo tanto, un posible peso sobre las mismas puede ser:

$$w_{ij} = e^{-I_{i \rightarrow j, \max} v_{\text{nom},i} |z_{ij}|}, \quad (3.5)$$

siendo $I_{i \rightarrow j, \max}$ la máxima corriente en kA que puede circular en la línea que conecta los nodos i, j , $v_{\text{nom},i}$ el voltaje nominal de dichos nodos (al estar conectados por una línea comparten voltaje nominal) y $|z_{ij}|$ el módulo de la impedancia entre ambos. Por otro lado, aquellos nodos conectados por un transformador siguen la ecuación:

$$w_{ij} = e^{-s_{i \rightarrow j, \text{nom}} v k_{ij}}, \quad (3.6)$$

con $s_{i \rightarrow j, \text{nom}}$ la potencia nominal del transformador y $v k_{ij}$ el voltaje de cortocircuito relativo del generador.

- **Conexo/Disconexo:** Según la estructura de conexión, un grafo puede ser conexo si existe un camino entre cualquier par de nodos, o disconexo si existen dos nodos para los cuales no existe un camino de interconexión. En el caso de esta tesis la red eléctrica es siempre conexa, ya que no se tienen barras aisladas del sistema.
- **Dirigido/No Dirigido:** La clasificación de los grafos en dirigidos y no dirigidos se centra en la naturaleza de las conexiones entre los nodos. En un grafo dirigido, las aristas tienen una dirección definida, representándose como pares ordenados de nodos. Esto indica que la relación es asimétrica; por ejemplo Twitter, donde una persona sigue a otra, sin necesidad de reciprocidad. Por otro lado, en un grafo no dirigido las aristas no tienen dirección, lo que implica una relación bidireccional o simétrica, como en Facebook, donde

3.4. Redes Neuronales sobre Grafos

no se tienen seguidores, sino amigos. En este caso, una red eléctrica forma un grafo no dirigido, ya que las líneas de transmisión no tienen una dirección definida.

- **Heterogéneo/No Heterogéneo:** Por último, se clasifican los grafos como heterogéneos y no heterogéneos, y se basa en la diversidad de tipos de nodos y aristas que presentan. Los grafos heterogéneos incluyen diferentes tipos de nodos y/o aristas, permitiendo representar relaciones complejas entre diferentes clases de entidades. Esto los hace especialmente útiles para modelar sistemas con variedad de elementos y conexiones. En contraste, los grafos no heterogéneos (u homogéneos) tienen nodos y aristas de un solo tipo, simplificando la estructura al tratar todas las entidades y relaciones de manera uniforme, lo cual es ideal para aplicaciones donde las interacciones son homogéneas y uniformes. Como se vio anteriormente, cada barra puede ser de uno de los siguientes 3 tipos: Bus de Referencia ($V\delta$), Bus de Carga (PQ) o Bus de Generación (PV), por lo que el grafo es heterogéneo.

Además de la representación completa del grafo mediante los conjuntos de vértices \mathcal{V} y aristas \mathcal{E} , estos presentan diferentes representaciones mediante matrices. Estas consisten en arreglos de dos dimensiones de los cuales se pueden inferir los vértices y aristas presentes en el grafo. Se procede a presentar las principales matrices utilizadas para representar el grafo.

Matriz de Adyacencia

La matriz de adyacencia A de un grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ es una matriz cuadrada de tamaño $N \times N$, donde $N = |\mathcal{V}|$ es el número de nodos del grafo. La matriz se define de la siguiente manera:

$$A_{ij} = \begin{cases} 1 & \text{si } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{si } (v_i, v_j) \notin \mathcal{E} \end{cases}. \quad (3.7)$$

Matriz de Pesos

Junto con la matriz de adyacencia, se define su contraparte ponderada, donde en las entradas donde existe una arista, en lugar de haber un 1, se tiene el peso de la arista. Esta matriz se conoce como matriz de adyacencia ponderada, o matriz de pesos. Se define como

$$W_{ij} = \begin{cases} w_{i,j} & \text{si } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{si } (v_i, v_j) \notin \mathcal{E} \end{cases}, \quad (3.8)$$

donde $w_{i,j}$ es el peso de la arista (v_i, v_j) .

Capítulo 3. Aprendizaje Automático

Matriz de Grado

Consiste en una representación parcial del grafo, ya que no se puede recuperar el conjunto completo de aristas a partir de esta. Es una matriz diagonal D , donde la entrada de la diagonal D_{ii} es igual al grado del vértice i . El grado de un vértice ($\deg(v)$) se define como el número de aristas incidentes al nodo. En el caso de grafos ponderados, el número de aristas incidentes es sustituido por la suma de los pesos de las aristas incidentes. De esta forma queda:

$$D_{ij} = \begin{cases} \deg(v_i) & i = j \\ 0 & i \neq j \end{cases}. \quad (3.9)$$

Matriz Laplaciana

Por otro lado, la matriz laplaciana L de un grafo es una matriz cuadrada de tamaño $N \times N$ definida como:

$$L = D - A.$$

También se extiende la matriz laplaciana para un grafo ponderado, donde en lugar de A se utiliza W . Esta matriz sirve para analizar diversas propiedades estructurales de un grafo y tiene múltiples aplicaciones en teoría de grafos y ciencias de la computación. Es útil para determinar la conectividad del grafo, ya que los valores propios de la matriz indican cuántas componentes conexas tiene el grafo y cuán fuertemente conectado está.

Versión Normalizada

Todas estas expresiones también presentan su versión normalizada. Esto se obtiene al dividir cada entrada (i, j) de la matriz por la raíz cuadrada del grado de i multiplicada por el grado de j . Por ejemplo, para la matriz de adyacencia, su versión normalizada \bar{A} se define como:

$$A_{ij} = \begin{cases} \frac{1}{\sqrt{\deg(v_i)\deg(v_j)}} & \text{si } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{si } (v_i, v_j) \notin \mathcal{E} \end{cases}. \quad (3.10)$$

Se puede definir matricialmente usando la matriz de grado D , como:

$$\bar{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}. \quad (3.11)$$

Similar es el caso para la matriz de peso, y la matriz laplaciana:

$$\bar{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \quad (3.12)$$

$$\bar{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}. \quad (3.13)$$

3.4. Redes Neuronales sobre Grafos

3.4.2. Procesamiento de señales en Grafos

En muchos casos, los nodos (o aristas) presentan atributos, por lo tanto resulta necesario definir el concepto de señales en grafos. En el caso más simple, se tiene una señal que corresponde a un único escalar por nodo. De esta forma, se define para el grafo \mathcal{G} la señal $\mathbf{x} \in \mathbb{R}^N$. Esta señal es intrínseca al grafo, en el sentido que la entrada x_i corresponde al valor de la señal para el vértice i . De forma trivial, se extiende esta definición para múltiples atributos por nodo. En este caso, la señal se define como la matriz $\mathbf{X} \in \mathbb{R}^{N \times F}$, donde la columna \mathbf{X}_i se corresponde con los F atributos de la barra i . Luego de definido el concepto de señal, se procede a introducir diferentes operaciones que se le pueden aplicar a esta.

Operador de Desplazamiento en Grafos

Dado el rol que juegan estas matrices en las redes neuronales a definir, en este trabajo se las llama Operadores de Desplazamiento en Grafos, o GSO por sus siglas en inglés (Graph Shift Operator). De forma general, el GSO se denota con la letra S , sin embargo, dependiendo de la matriz particular que se esté usando, se puede utilizar su letra. En general, esta matriz se elige dentro de $S \in \{A, W, L, \bar{A}, \bar{W}, \bar{L}\}$.

Desplazamiento en Grafos

Es entonces que se define la primer operación; desplazamiento de una señal sobre el grafo. Por simplicidad se introduce esta operación para el caso de una señal de un atributo por nodo, pero su extensión para múltiples atributos es trivial. Esta operación consiste en multiplicar la matriz S por el vector \mathbf{x} , resultando en la señal desplazada $\mathbf{y} \in \mathbb{R}^N$. Esto es:

$$\mathbf{y} = S\mathbf{x} . \quad (3.14)$$

Observando que independientemente de la elección de S , la matriz es siempre 0 para las entradas donde no existe arista, es fácil ver que el componente y_i consiste en una combinación lineal de los componentes de la señal \mathbf{x} asociados a los nodos vecinos de v_i , $\mathcal{N}(v_i)$:

$$y_i = \sum_{j \in [0, N]} S_{ij} x_j = \sum_{j \in \mathcal{N}(v_i)} S_{ij} x_j . \quad (3.15)$$

Es por esta razón que la operación de desplazamiento se considera una operación local, ya que la información se transmite únicamente entre nodos conectados entre sí. Esta operación se puede componer de forma secuencial, obteniendo así una secuencia de desplazamientos. Se define como:

$$\mathbf{x}^{(k)} = S^k \mathbf{x} . \quad (3.16)$$

En este caso, se tiene que la señal resultante $\mathbf{x}^{(k)}$ cumple que su entrada i , asociada al nodo v_i , contiene información de los vecinos a k saltos de v_i . Los vecinos a k saltos de un nodo hace referencia a todos los nodos que presentan un camino de

Capítulo 3. Aprendizaje Automático

largo k . De esta forma, se define una operación local que permite combinar entradas de la señal, teniendo en cuenta la estructura del grafo.

3.4.3. Convolución en Grafos

Luego de definida la operación de desplazamiento, se procede a presentar la operación de convolución. Esta consiste en una combinación lineal de versiones desplazadas de la señal original. Dada la señal de entrada $\mathbf{x} \in \mathbb{R}^N$, el operador de desplazamiento $S \in \mathbb{R}^N$, y un conjunto de coeficientes $\{h_k\}_{k \in [0 \dots K]}$, se define la convolución como:

$$\mathbf{y} = \sum_{k=0}^K S^k \mathbf{x} h_k = S^0 \mathbf{x} h_0 + S^1 \mathbf{x} h_1 + \dots + S^K \mathbf{x} h_K . \quad (3.17)$$

A modo de ejemplo, se puede hacer una analogía con el tiempo. Utilizando un grafo línea dirigido, donde los nodos representan una secuencia temporal, la operación de desplazamiento puede interpretarse como un desplazamiento en el tiempo. En el procesamiento de señales temporales, la convolución combina versiones desplazadas de la señal original. De manera análoga, en un grafo con matriz de adyacencia S , el operador $S^k \mathbf{x}$ desplaza la señal k pasos en el grafo, similar al desplazamiento en el tiempo. Así, la convolución en el grafo es análoga a la convolución temporal, capturando la influencia de las versiones desplazadas de la señal en cada nodo, de forma similar al procesamiento temporal de señales.

La convolución se puede extender para el caso de que se tengan múltiples características de entrada y salida en la convolución, es decir, señales multidimensionales. Se empieza por extender para una señal con múltiples entradas. Para esto, el vector de entrada pasa a ser una matriz $\mathbf{X} \in \mathbb{R}^{N \times F_{in}}$. Luego, los coeficientes que eran $h_k \in \mathbb{R}$ pasan a ser vectores $\mathbf{h}_k \in \mathbb{R}^{F_{in}}$. De esta forma, se obtiene que:

$$\mathbf{y} = \sum_{k=0}^K S^k \mathbf{X} \mathbf{h}_k . \quad (3.18)$$

Finalmente, se define el caso general, para múltiples entradas y múltiples salidas. En este caso, la salida pasa a ser una matriz $\mathbf{Y} \in \mathbb{R}^{N \times F_{out}}$. Para lograr esto, los coeficientes, que en el caso anterior eran vectores, ahora pasan a ser matrices $\mathbf{H}_k \in \mathbb{R}^{F_{in} \times F_{out}}$. Esto resulta en la forma más general de convolución de señales en grafos:

$$\mathbf{Y} = \sum_{k=0}^K S^k \mathbf{X} \mathbf{H}_k , \quad (3.19)$$

donde al conjunto de matrices $\{\mathbf{H}_k\}_{k \in [0, K]}$ se le denomina banco de filtros.

3.4.4. Redes Neuronales en Grafos

Definida la operación de convolución, la extensión de redes neuronales para datos en grafos es directa. Esta operación va a suplantar el papel de la combinación

3.4. Redes Neuronales sobre Grafos

lineal en las redes completamente conectadas. Por lo tanto, se comienza definiendo el perceptrón en grafos, lo cual consiste en componer el filtro convolucional con una función de activación no lineal $f(\cdot)$ a elección:

$$\mathbf{Y} = f \left(\sum_{k=0}^K S^k \mathbf{X} \mathbf{H}_k \right) . \quad (3.20)$$

Luego, la red neuronal consiste en implementar L capas, donde cada capa corresponde a un perceptrón. Similar al caso estándar, la salida de una capa corresponde a la entrada de la siguiente. Por lo tanto, se tiene que:

$$\mathbf{X}^{(l)} = f \left(\sum_{k=0}^K S^k \mathbf{X}^{(l-1)} \mathbf{H}_k^{(l-1)} \right) , \quad (3.21)$$

donde $\mathbf{X}^{(l-1)}$ y $\mathbf{X}^{(l)}$ son la entrada y salida de la capa $l - 1$ respectivamente, $\{\mathbf{H}_k^{(l-1)}\}_{k \in [0, K]}$ es el banco de filtros de la capa $l - 1$. En este caso, el conjunto de parámetros aprendibles son los L bancos de filtros de cada capa. En la Figura 3.5 se puede ver una representación esquemática de una red neuronal sobre grafos con los filtros de cada capa.

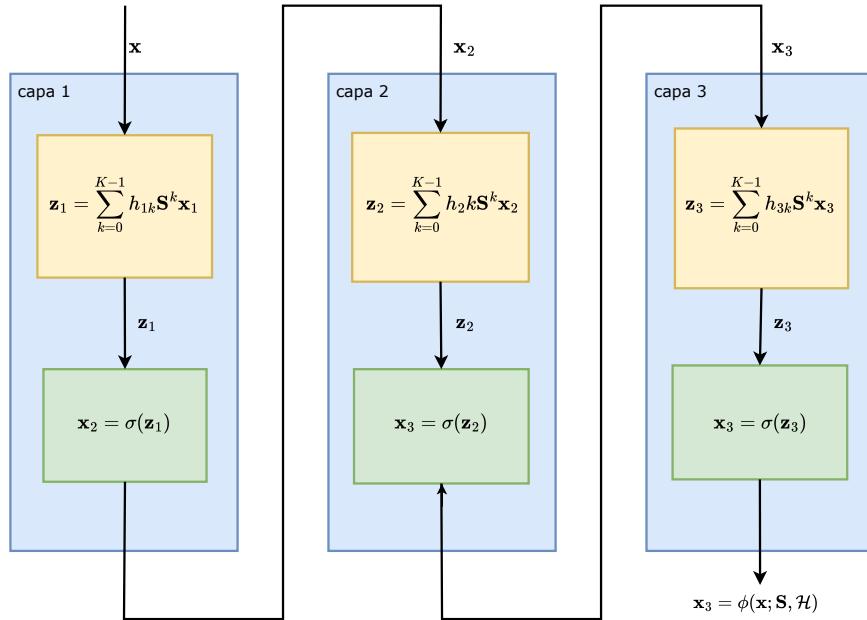


Figura 3.4: Esquema de 3 capas anidadas de una red neuronal en grafos.

3.4.5. Comparación de GNN contra FCNN

A lo largo de esta sección se hace énfasis en la ventaja de las redes neuronales sobre grafos, dada su capacidad de complementar la información en la señal con la estructura del grafo. Sin embargo, si se comparan ambas familias de funciones en detalle, se observa que las redes neuronales para datos en grafos son un subconjunto de las redes neuronales completamente conectadas. Esto se justifica fácilmente para el caso de una característica de entrada y salida, vista en la Ecuación 3.17. Dada la pareja de entrada y salida $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, en el caso de la convolución en grafos, la salida se obtiene como:

$$\mathbf{y} = \sum_{k=0}^K S^k \mathbf{x} h_k = \left(\sum_{k=0}^K S^k h_k \right) \mathbf{x} , \quad (3.22)$$

mientras que para la combinación lineal en redes completamente conectadas se obtiene a partir de multiplicar por una matriz genérica $W^T \in \mathbb{R}^{N \times N}$ como:

$$\mathbf{y} = W^T \mathbf{x} . \quad (3.23)$$

Como W^T es una matriz de pesos cualquiera, tomando $W^T = \sum_{k=0}^K S^k h_k$ evidencia que el primero es un subcaso del segundo. Esto se puede extender para casos con varias características de entrada y salida, resultando en que toda función perteneciente a la familia de las GNN, pertenece también a la familia de las FCNN. Por lo tanto, se tiene que, en teoría, las FCNN deberían tener mejor desempeño que las GNN, ya que toda función obtenida para la segunda familia puede ser obtenida con la primera.

Sin embargo, esto no ocurre en la práctica, ya que las redes neuronales en grafos generalizan mejor a nuevas señales. Esto se debe a que, aunque menos expresivas, las GNN respetan propiedades que siempre se cumplen en señales sobre grafos. Un ejemplo de esto es la equivarianza a permutaciones, lo cual se explica a continuación.

Suponer un grafo con GSO S y una señal \mathbf{x} sobre este. Una permutación en un grafo consiste en simplemente reordenar los índices en los vértices. Al reordenar los índices en los vértices del grafo, tanto el GSO S como la señal x sufren de una alteración en el orden de sus entradas. En la Figura 3.5 se observa un ejemplo de permutación de un grafo sencillo, para la cual:

$$S = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow \bar{S} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} , \quad (3.24)$$

$$x = \begin{bmatrix} 3 \\ -1 \\ 2 \\ 0 \end{bmatrix} \rightarrow \bar{x} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 3 \end{bmatrix} , \quad (3.25)$$

3.5. Aprendizaje Automático para ORPD

donde como matriz S se utiliza la de adyacencia. Lo que se puede observar es que la permutación únicamente modifica el orden, pero tanto el grafo como la señal son las mismas. Es necesario imponer un orden al trabajar con GSO y señales, pero este orden es arbitrario. Por lo tanto, es deseable que la salida sea independiente del ordenamiento elegido. Esto en efecto, se respeta en las GNN, donde la salida al tomar una señal y grado permutado sería la misma salida pero permutada.

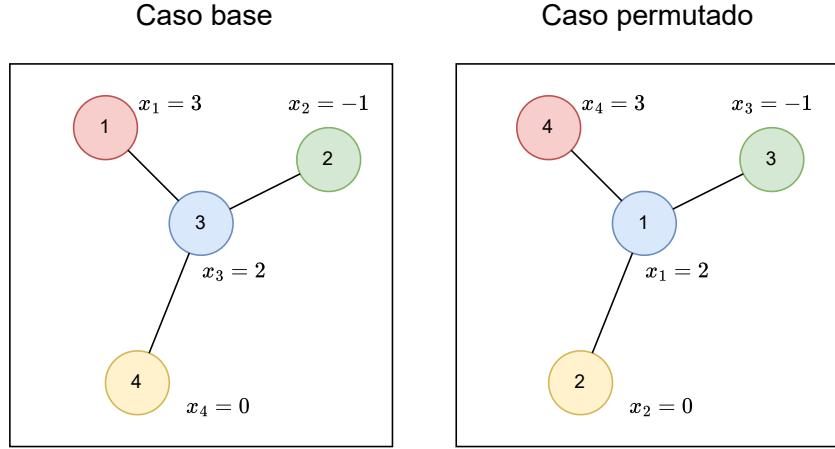


Figura 3.5: Permutación de un grafo.

Esta propiedad es mucho más valiosa que solo permitir trabajar con cualquier ordenamiento del grafo de forma arbitraria. Esta característica permite explotar simetrías que ocurren dentro del grafo y datos, lo que permite a las GNN generalizar mejor y aprender representaciones más robustas. En particular, los datos asociados a una parte del grafo pueden ser explotados en todas las ubicaciones equivalentes bajo las simetrías del grafo. Esto incrementa la cantidad efectiva de datos disponibles. Por el contrario, las FCNN no pueden aprovechar estas simetrías de manera natural, ya que no son equivariantes a permutaciones y, por lo tanto, requieren más datos y capacidad para lograr un desempeño similar.

3.5. Aprendizaje Automático para ORPD

El campo de la optimización es un área de gran estudio a lo largo de la historia de las matemáticas, cuyo objetivo principal es seleccionar, dentro de un conjunto de posibilidades, la opción que sea óptima bajo algún criterio predefinido. Matemáticamente, esto se formaliza como hallar, dentro de un espacio de variables definido a partir de restricciones, el conjunto de variables que minimice o maximice una función objetivo. Este campo clasifica los distintos tipos de problemas según las propiedades que poseen tanto la función objetivo como las restricciones que definen el espacio posible de variables. A su vez, se desarrollan diferentes estrategias de resolución para las distintas categorías de problemas, donde a mayor

Capítulo 3. Aprendizaje Automático

complejidad en el modelo del problema, mayor complejidad en la estrategia para resolverlo.

La clase más sencilla es la Programación Lineal [12], que consiste en problemas donde tanto la función objetivo como las restricciones son lineales. En 1947 se presentó el algoritmo Simplex, que permite resolver esta familia de problemas de forma eficiente, resultando en aplicaciones inmediatas en áreas como logística, transporte y economía. Sin embargo, muchos problemas del mundo real no se ajustan a modelos lineales, por lo que este hito motivó el estudio de problemas más generales y complejos. Dentro de estas familias de problemas se encuentran la Programación Cuadrática, la Programación Convexa y la Programación No Convexa.

En este trabajo resulta de interés la última, ya que el problema de ORPD cae dentro de la categoría de problemas de optimización no convexa con restricciones. Esta es una clase particularmente desafiante dentro del área, ya que se enfrenta a problemas en los que tanto la función objetivo como las restricciones no cumplen las condiciones de convexidad. Sin entrar en detalle sobre esta propiedad, la falta de convexidad implica que pueden existir varios mínimos locales y que el espacio de variables posibles presenten estructuras más complejas que las de otras clases.

La estrategia estándar para resolver esta familia de problemas es el método de punto interior [7], mencionado en el Capítulo 2. Aunque popular, este método presenta dos desventajas. En primer lugar, este algoritmo no realiza una búsqueda amplia sobre el espacio de variables; dado un punto inicial, busca minimizar rápidamente el objetivo manteniéndose dentro del espacio factible. Esta estrategia provoca frecuentemente caer en mínimos locales, los cuales son soluciones subóptimas. Por otro lado, al tratarse de un método iterativo, resolver problemas de gran escala o resolver un problema múltiples veces resulta costoso computacionalmente.

En este trabajo se propone utilizar aprendizaje automático [5], otra estrategia heurística, para resolver el problema de optimización. Esto se debe a tres razones: en primer lugar, el aprendizaje automático ha experimentado un auge significativo en los últimos años, demostrando ser una herramienta poderosa para resolver problemas complejos y adaptarse a datos no lineales y de alta dimensionalidad; en segundo lugar, gran parte del cómputo necesario para el uso de estas estrategias recae en el proceso de entrenamiento, el cual es *offline*. Por lo tanto, cuando sea necesario resolver el despacho de reactiva, simplemente se debe ejecutar una vez la propagación hacia adelante del predictor, lo cual es mucho más liviano que un método iterativo. Tercero y por último, se cuenta con una extensa base de datos de la red eléctrica uruguaya, lo que proporciona un panorama ideal para entrenar modelos que aprendan soluciones para escenarios típicos en el sistema eléctrico.

Vinculando los conceptos presentados hasta el momento, la propuesta puede resumirse en entrenar un mapeo, ya sea mediante una FCNN o una GNN, el cual, dado un conjunto de parámetros de entrada al ODPR, prediga variables de control que resulten en estados factibles y con pérdidas bajas.

Teniendo en cuenta que el problema consiste en aprender un mapeo entrada-salida óptimo y que se dispone de un optimizador que permite resolver este mapeo, resulta intuitivo tratarlo como un problema de aprendizaje supervisado. Dentro de

3.5. Aprendizaje Automático para ORPD

este paradigma, se busca imitar el comportamiento del optimizador para los casos representados por el conjunto de datos. La premisa de este enfoque es obtener un método no iterativo, que permita reducir los tiempos de ejecución manteniendo un desempeño comparable al optimizador. Esto es particularmente útil en escenarios donde es necesario resolver múltiples instancias del problema.

La principal limitación de esta estrategia es que el desempeño del predictor está acotado por el del optimizador a imitar. Por esta razón, se propone una segunda estrategia de aprendizaje no supervisado. Para ello, se entrena un predictor que minimice una función de pérdida relacionada con el problema a resolver. Esta función se compone de la función objetivo más penalizaciones que aumentan para predicciones que no cumplen las restricciones. A continuación, se explican en detalle ambas estrategias.

3.5.1. Imitar Soluciones del Optimizador

Esta estrategia se alinea con el esquema presentado en la Figura 3.1. El mapeo desconocido que se busca imitar es el proceso mediante el cual el optimizador obtiene las soluciones. Si bien el mapeo en sí es desconocido, ya que consiste en numerosas iteraciones realizadas por el IPOPT, se dispone de un conjunto de pares entrada-solución $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t \in [0, T]}$. El proceso de obtención de esta base de datos se detalla en el próximo capítulo, sin embargo esta sección presenta la estructura de estas señales. En particular, se diferencia entre la estructura de entrada para las FCNN y para las GNN.

De forma genérica, una red eléctrica cuenta con N barras, donde los parámetros de entrada en cada barra están dados por $F^{in} = 5$ atributos. Estos atributos son: la potencia activa y reactiva demandadas por esa barra; la potencia activa generada por un generador controlador de voltaje; y la potencia activa y reactiva generadas por un generador estático.

La entrada de la FCNN se obtiene concatenando las características de cada barra, obteniendo así un vector de longitud NF^{in} . Esta longitud coincide con la cantidad de neuronas en la capa de entrada, donde cada neurona representa una característica de una barra. Matemáticamente, la entrada es:

$$\mathbf{x} = [\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_N] \in \mathbb{R}^{NF^{in}}, \quad (3.26)$$

con

$$\mathbf{x}_i = [p_i^{\text{dem}}, q_i^{\text{dem}}, p_i^{\text{gen},v}, p_i^{\text{gen},\text{est}}, q_i^{\text{gen},\text{est}}], \quad i = 1, \dots, N. \quad (3.27)$$

La entrada a una GNN consiste en las mismas características que la FCNN, pero difiere en cómo se estructura. En lugar de un vector, se organiza como una matriz de tamaño $\mathbb{R}^{N \times F^{in}}$, donde cada fila corresponde al vector \mathbf{x}_i definido anteriormente:

Capítulo 3. Aprendizaje Automático

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} \in \mathbb{R}^{N \times F^{in}} . \quad (3.28)$$

En cuanto a la salida, se busca predecir el voltaje de salida en el conjunto de generadores controladores de voltaje \mathcal{G} y la potencia reactiva en los compensadores de reactiva \mathcal{S} . Por lo tanto, la cantidad de características de salida por barra es $F^{out} = 2$. Una aclaración importante es que, dado que solo algunas barras tienen la capacidad de controlar su voltaje y otras pueden variar su generación de reactiva mediante sus compensadores, la salida para todas aquellas barras que no pueden controlar ninguno de estos dos parámetros será cero. Esto se controla con una máscara durante el entrenamiento, que asigna cero a las salidas correspondientes.

De esta forma, el vector de salida en la FCNN es de longitud NF^{out} , donde:

$$\mathbf{y} = [\mathbf{y}_1 \mid \mathbf{y}_2 \mid \cdots \mid \mathbf{y}_N] \in \mathbb{R}^{NF^{out}} , \quad (3.29)$$

con

$$\mathbf{y}_i = [v_{m,i}, q_i^{\text{comp}}], \quad i = 1, \dots, N . \quad (3.30)$$

De manera similar al caso de las entradas, en la GNN se tiene una matriz de salida en lugar de un vector, con tamaño $N \times F^{out}$, donde cada fila de la matriz corresponde a una neurona en la capa de salida:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} \in \mathbb{R}^{N \times F^{out}} . \quad (3.31)$$

Dado que se busca imitar el comportamiento del optimizador, el proceso de entrenamiento consiste en minimizar una función de pérdida proporcional a la diferencia entre la salida del predictor y la salida del optimizador. Esto es:

$$\theta^* = \arg \min_{\theta} \sum_{(x_t, y_t)} \mathcal{L}(\mathbf{y}_t, \Phi(x_t; \theta)) , \quad (3.32)$$

donde $\Phi(x_t; \theta)$ es la predicción de la red para el dato x_t , θ es el conjunto de parámetros que define la FCNN o GNN, y θ^* es el conjunto de parámetros que resulta en el predictor óptimo. Ejemplos de la función de pérdida \mathcal{L} pueden ser el error cuadrático medio (MSE) o el error absoluto medio (MAE):

$$\text{MSE} = \|\mathbf{y} - \Phi(x; \theta)\|^2 , \quad (3.33)$$

$$\text{MAE} = |\mathbf{y} - \Phi(x; \theta)| . \quad (3.34)$$

3.5.2. Minimización de Pérdidas

En esta estrategia se entrena al predictor para minimizar una función de pérdida derivada del problema a resolver. En este caso, el conjunto de datos de entrenamiento consiste únicamente en $\{\mathbf{x}_t\}_{t \in [0, T]}$, ya que no hay solución óptima a la cual imitar. El esquema de entrada en este caso es idéntico al anterior, donde x_t se define como en la Ecuación 3.26 para el caso de la FCNN y como en la Ecuación 3.28 para la GNN.

Por otro lado, la señal de salida difiere del caso anterior. Dado que la función de pérdida precisa evaluar el objetivo y restricciones del problema, es necesario conocer todos el conjunto de variables, incluyendo las implícitas. Se nota que en el caso anterior, no es que estas variables no fuesen tenidas en cuenta, sino que el optimizador las tuvo en cuenta al generar el óptimo, del cual luego el predictor aprende. En este caso, al no contar con los óptimos, es necesario conocer las variables implícitas. La manera mediante la cual se obtienen estas permite definir diferentes estrategias.

La opción más intuitiva es correr un flujo de carga. De esta forma, para una entrada \mathbf{x}_t , la red neuronal predice las variables de control, se corre un flujo de carga, y luego se evalúa la función de pérdida. Esto tiene dos grandes problemas. En primer lugar, es altamente ineficiente. Correr un flujo de carga en cada iteración del entrenamiento agrega una sobrecarga de cómputo prohibitiva. En segundo lugar, para resolver un flujo de carga es necesario usar métodos iterativos, los cuales hacen extremadamente compleja su diferenciación, condición necesaria para entrenar estas redes. Por lo tanto, es necesario utilizar una alternativa a correr un flujo de carga, la cual sea tanto eficiente como diferenciable.

Para esto, se propone agregar las variables implícitas a la señal de salida de la red neuronal. Se recuerda que estas variables son implícitas, ya que están definidas implícitamente por la igualdad de flujo de carga. Al agregar estas variables como salida del predictor, esta igualdad en principio no se cumple. Para asegurar que se cumpla, se agrega a la función de pérdida una penalización que crece si esta se viola.

Por lo tanto, la salida del predictor consiste en las variables de control \mathbf{s}^{comp} y $\{v_{m,i}\}_{i \in \mathcal{B}_G}$, más las variables implícitas $\mathbf{q}^{\text{gen},v}$, $\mathbf{p}^{\text{gen},\text{ref}}$, $\mathbf{q}^{\text{gen},\text{ref}}$, $\{v_{m,i}\}_{i \in \mathcal{B}_C}$, δ . Se juntan $\{v_{m,i}\}_{i \in \mathcal{B}_C}$ y $\{v_{m,i}\}_{i \in \mathcal{B}_G}$ como una única característica de salida, ya que son disjuntas las barras donde estos elementos no son cero. Lo mismo ocurre con $\mathbf{q}^{\text{gen},v}$ y $\mathbf{q}^{\text{gen},\text{ref}}$, ya que no se tiene un generador de referencia y uno controlador de voltaje en la misma barra. Esto resulta en $F^{out} = 5$ características de salida.

Para el caso de FCNN, estas se concatenan en un vector de largo NF^{out} , obteniendo:

$$\mathbf{y} = [\mathbf{y}_1 | \mathbf{y}_2 | \cdots | \mathbf{y}_N] \in \mathbb{R}^{NF^{out}},$$

donde

$$\mathbf{y}_i = [v_{m,i}, q_i^{\text{comp}}, q_i^{\text{gen},v}, p_i^{\text{gen},\text{ref}}, \delta_i] \quad i = 1, \dots, N.$$

En el caso de la GNN, se tiene una matriz de salida en lugar de un vector,

Capítulo 3. Aprendizaje Automático

cuyo tamaño es $N \times F^{out}$, donde la salida es:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} \in \mathbb{R}^{N \times F^{out}}.$$

Esta estrategia se basa en la premisa de una función de pérdida que logre capturar tanto el objetivo como las restricciones del problema, permitiendo así entrenar el predictor. Para esto, se parte de la función objetivo del problema, definida en Ecuación 2.51. Esto es:

$$\mathbf{c}(\mathbf{p}) = \sum_{(i,j) \in \mathcal{L}} Re(v_i Y_{\text{line,to}} \mathbf{v} + v_j Y_{\text{line,from}} \mathbf{v}) , \quad (3.35)$$

donde $\mathbf{p} = (\mathbf{x}, \Phi(\mathbf{x}, \Theta))$ representa de forma general el conjunto de variables completo de la red eléctrica definido a partir de la entrada y salida del predictor.

A esta se le suman penalizaciones proporcionales a la violación de las restricciones de igualdad y desigualdad. Sea de forma genérica $\{h_i(\mathbf{p}) = 0\}$ el conjunto de restricciones de igualdad, se define una función de penalización para igualdades $\phi(h_i(\mathbf{p}))$. Esta función ϕ debe cumplir con la condición de crecer con el valor absoluto de $h_i(\mathbf{p})$, además de ser diferenciable. Se introduce un abuso de notación, definiendo aplicar la función de penalización a una igualdad como $\phi_i(\mathbf{p}) = \phi(h_i(\mathbf{p}))$. Para el problema a tratar, se tiene que la única igualdad a cumplir es tanto la parte real como imaginaria del flujo de carga en todas las barras. De esta forma, el conjunto de igualdades en este problema es:

$$\{h_i(\mathbf{p}) = 0\} = \{Re(s_i - v_i Y_{\text{bus},i}^* \mathbf{v}^*) = 0\} \cup \{Im(s_i - v_i Y_{\text{bus},i}^* \mathbf{v}^*) = 0\} , \quad (3.36)$$

mientras que el conjunto de penalizaciones queda

$$\{\phi_i(\mathbf{p})\} = \{\phi(Re(s_i - v_i Y_{\text{bus},i}^* \mathbf{v}^*))\} \cup \{\phi(Im(s_i - v_i Y_{\text{bus},i}^* \mathbf{v}^*))\} . \quad (3.37)$$

Por otro lado, se tiene de forma genérica el conjunto de restricciones de desigualdad $\{g_i(\mathbf{p}) \leq 0\}$. Se proponen dos estrategias para asegurar su cumplimiento. En primer lugar, se utiliza una función de penalización para desigualdades $\psi(g_i(\mathbf{p}))$. En este caso, la penalización debe cumplir con ser cero o muy chica para valores negativos y crecer de forma acelerada para valores que se acerquen a cero, o que sean positivos. A su vez, debe ser diferenciable. La segunda estrategia a utilizar es, para aquellas desigualdades que acotan el valor de una variable de forma individual, forzar a la salida del predictor que las variables se encuentren en este rango. Para esto, luego de la última capa de la red neuronal, se agrega una función de activación $\gamma(x_{\min}, x_{\max}, x)$ que limite el rango de salida. Esta función debe cumplir con ser no lineal, diferenciable, y que tienda a x_{\min} para valores negativos, y a x_{\max} para valores positivos. De esta forma, si se tiene una pareja de desigualdades $\{x_i - x_{\max} \leq 0\}$ y $\{-x_i + x_{\min} \leq 0\}$, donde x_i es una de las salidas

3.5. Aprendizaje Automático para ORPD

de la red, en lugar de agregar estas desigualdades como penalizaciones a la función de pérdida, se limita el rango de salida mediante la función $\gamma(x_{\min}, x_{\max}, x)$. Esta segunda estrategia es más eficiente a la hora de asegurar el cumplimiento de las desigualdades, ya que directamente acota el valor de las variables, sin embargo, es solo posible utilizarla en desigualdades sencillas, que no combinen múltiples variables de salida en la desigualdad.

Para el problema a tratar, se tienen varias desigualdades que se pueden atacar con la segunda estrategia. En particular, los parámetros que tienen restricciones y son predichos por el modelo son v_m , δ , $q^{\text{gen},v}$, $p_i^{\text{gen,ref}}$, $q_i^{\text{gen,ref}}$ y q_i^{comp} . De esta forma, las últimas cinco desigualdades en la Ecuación 2.52 quedan resueltas.

En el caso de las restricciones sobre la capacidad de las líneas es necesario utilizar penalizaciones, ya que vinculan varios voltajes en cada desigualdad. Para esto, se tiene:

$$\{\psi_i(\mathbf{p})\} = \{\psi(|Y_{\text{line,to}\mathbf{v}}| - I_{\text{line,máx}})\} \cup \{\psi(|Y_{\text{line,from}\mathbf{v}}| - I_{\text{line,máx}})\} . \quad (3.38)$$

Finalmente, la función de pérdida queda definida a partir de combinar la función objetivo con las penalizaciones. Al agregar estas penalizaciones, se multiplican por un factor λ_i en el caso de las igualdades, y por μ_i las desigualdades. Estos son parámetros que ponderan las penalizaciones, permitiendo regular el impacto que genera esa penalización:

$$\mathcal{L}(\mathbf{x}_t, \Phi(\mathbf{x}_t, \theta)) = \mathbf{c}(\mathbf{x}_t, \Phi(\mathbf{x}_t, \theta)) + \sum_{i=1}^n \lambda_i \phi_i(\mathbf{x}_t, \Phi(\mathbf{x}_t, \theta)) + \sum_{i=1}^m \mu_i \psi_i(\mathbf{x}_t, \Phi(\mathbf{x}_t, \theta)) . \quad (3.39)$$

Similar a cualquier enfoque de aprendizaje automático, el problema resulta en encontrar el conjunto de parámetros θ que cumpla:

$$\theta^* = \min_{\theta} \mathcal{L}(\mathbf{x}_t, \Phi(\mathbf{x}_t, \theta)) . \quad (3.40)$$

Esta estrategia presenta una alternativa al problema de optimización sin necesidad de contar con soluciones óptimas predefinidas. La clave radica en la correcta definición de la función de pérdida, la cual debe ser lo suficientemente robusta para capturar tanto el objetivo como las restricciones del problema de forma eficiente y diferenciable. Aunque la introducción de penalizaciones agrega complejidad al proceso de entrenamiento, permite flexibilizar la estrategia de aprendizaje, haciendo posible el ajuste de las predicciones del modelo con las condiciones impuestas por el sistema eléctrico.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 4

Redes de Transmisión a Estudiar

El objetivo principal de esta tesis es abordar el despacho óptimo de potencia reactiva en la red eléctrica uruguaya. Para alcanzar este objetivo, se ha decidido incluir dos redes de transmisión adicionales en los casos de estudio: las redes IEEE 30 e IEEE 118. Estas redes son ampliamente reconocidas en la literatura y se utilizan frecuentemente para el análisis y la optimización de sistemas eléctricos. La razón de su incorporación es que permiten aumentar progresivamente la complejidad del problema. Además, al trabajar con redes IEEE, se tiene la certeza de que estos modelos están correctamente construidos, lo que no siempre se puede garantizar con el modelo de la red uruguaya desarrollado como parte de este proyecto.

Otra simplificación importante que ofrecen las redes IEEE es la facilidad en el manejo de los datos. Estas redes no disponen de un historial de datos de potencia, por lo que es necesario generar datos sintéticos a partir de valores nominales. Esto da lugar a un conjunto de datos más simple y evita posibles errores en el entrenamiento debido a problemas en la adquisición o procesamiento de los datos. De este modo, las estrategias propuestas pueden evaluarse en un entorno controlado antes de su aplicación a la red eléctrica uruguaya, más compleja y realista.

Posteriormente de definidos estos dos casos de estudio, se presenta el desarrollo del modelo de la red eléctrica uruguaya y base de datos. Se comienza explicando el sector eléctrico en Uruguay, lo que permite introducir generalidades sobre el proceso de despacho de carga en el país y contextualizar el origen de los datos a utilizar. A continuación, se expone la motivación para desarrollar un modelo propio de la red y se justifica por qué los modelos existentes no son adecuados para este trabajo. Se detallan todos los aspectos de la implementación necesarios para garantizar la mayor fidelidad posible con la realidad, concluyendo con una descripción exhaustiva de las barras, líneas y elementos que conforman el modelo. Finalmente, se explica el proceso de curado de los datos para asegurar una base de datos limpia y lista para su uso. Tanto los datos como el modelo de la red eléctrica uruguaya se dejan públicos en Google Drive¹, como fue mencionado anteriormente.

¹https://drive.google.com/drive/folders/121s67_IgW-r39hG-xwHIUI32-QIE97X?usp=sharing

Capítulo 4. Redes de Transmisión a Estudiar

4.1. Redes IEEE

Las redes de potencia IEEE son un conjunto de redes de transmisión presentadas como casos de estudio públicamente disponibles por la Universidad de Illinois a principios de los años 90. Estas consisten en 7 redes que varían en tamaño desde 14 hasta 300 barras. Estas redes proporcionan un entorno controlado y bien documentado para probar y evaluar tanto algoritmos como técnicas de optimización antes de su implementación en la red uruguaya. A su vez, permiten explorar diferentes arquitecturas y estrategias, facilitando la identificación de mejores prácticas y posibles desafíos que podrían surgir en la aplicación a la red uruguaya. Debido a la popularidad de estas redes, la librería PandaPower ya presenta una implementación hecha para ellas, lo que facilita aún más su uso en simulaciones y análisis de redes eléctricas.

Como fue mencionado previamente, se utilizan solo dos de estas redes: la IEEE30 y la IEEE118, que cuentan con 30 y 118 barras respectivamente. La razón de utilizar la IEEE30 es contar con una red de tamaño pequeño, lo cual hace que sea posible interpretar los resultados más intuitivamente, algo muy útil a la hora de detectar posibles causas de errores ante imprevistos. Por otro lado, la red IEEE118 es elegida ya que la cantidad de barras es similar a la de la uruguaya, la cual tiene 107 barras. De esta forma se asegura que una estrategia operativa para la red 30 mantiene su validez para una red de tamaño similar a la uruguaya.

4.1.1. IEEE30

El primer caso de prueba utilizado es el sistema IEEE de 30 buses, conocido como IEEE 30-Bus System [22]. Este es una representación simplificada del sistema American Electric Power (AEP) en diciembre de 1961. En la Figura 4.1 se puede observar un esquema unifilar de la misma.

Este consiste en $|\mathcal{B}| = 30$ barras, con $|\mathcal{G}| = 5$ generadores controladores de voltaje, $|\mathcal{C}| = 20$ cargas y un único generador de referencia. Esta red no presenta ni generadores estáticos ($|\mathcal{E}| = \emptyset$) ni compensadores de reactiva ($|\mathcal{S}| = \emptyset$). Por otro lado, tiene $|\mathcal{L}| = 41$ líneas y ningún transformador ($|\mathcal{T}| = \emptyset$). Al no contar con compensadores de reactiva, resolver el problema de despacho óptimo de reactiva resulta en determinar los 5 voltajes a controlar en los generadores.

4.1.2. IEEE118

El segundo caso de prueba empleado es el sistema IEEE de 118 buses, conocido como IEEE 118-Bus System [21]. Este representa una aproximación más compleja del sistema de American Electric Power (AEP) en diciembre de 1962. Un diagrama esquemático se puede observar en la Figura 4.2.

Este modelo consiste en $|\mathcal{B}| = 118$ barras, con $|\mathcal{G}| = 53$ generadores controladores de voltaje, $|\mathcal{C}| = 99$ cargas y un único generador de referencia. Esta red no presenta ni generadores estáticos ($|\mathcal{E}| = \emptyset$) ni compensadores de reactiva ($|\mathcal{S}| = \emptyset$). Por otro lado, tiene $|\mathcal{L}| = 173$ líneas y $|\mathcal{T}| = 43$ transformadores. Nuevamente,

4.1. Redes IEEE

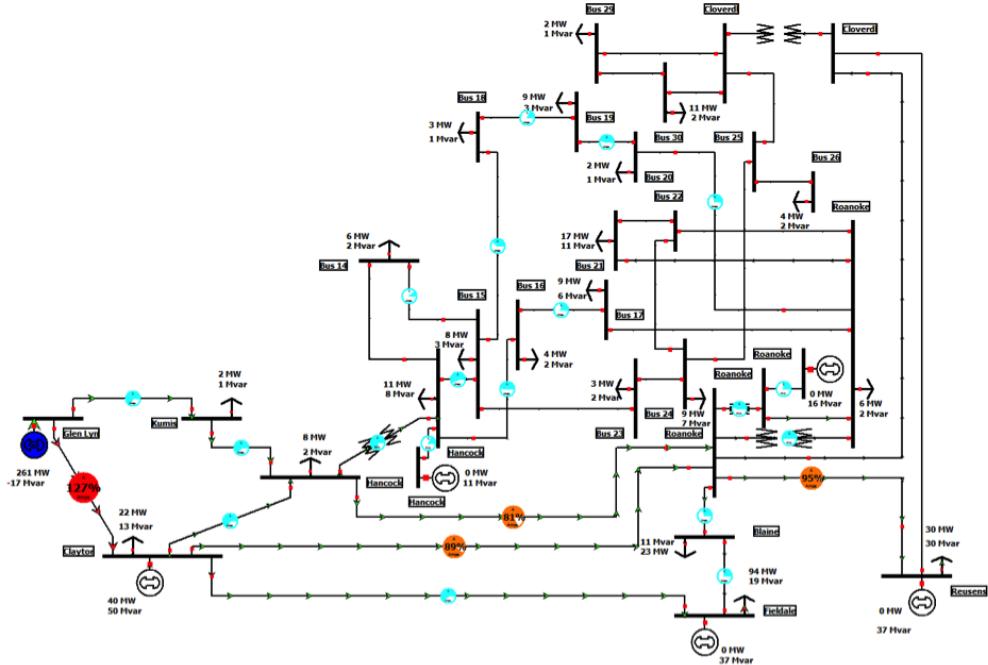


Figura 4.1: Esquema unifilar de la red eléctrica IEEE-30 [22].

no se cuenta con compensadores de reactiva, sin embargo, las variables de control aumentan significativamente, teniendo que resolver un total de 53 voltajes de generadores.

4.1.3. Datos

Como se adelantó al comienzo del capítulo, ninguna de las dos redes ofrece un histórico de generación o demanda de potencia para entrenar los modelos. Esto implica que para trabajar con estas redes es necesario generar una base de datos sintética. Los datos que son necesarios simular son aquellos que se toman como entrada al problema. Recordando la formulación del problema en el Capítulo 2, se deben generar valores de p_i^{dem} , q_i^{dem} y $p_i^{\text{gen},v}$ (se omite $s^{\text{gen},\text{est}}$ ya que la red no presenta generadores estáticos).

Como metodología utilizada para la generación de datos sintéticos se realiza un proceso similar a los utilizados en otros trabajos que abordan este problema con aprendizaje automático [28]. Esta consiste en tomar valores nominales de $p_i^{\text{dem,nom}}$, $q_i^{\text{dem,nom}}$ y $p_i^{\text{gen},v,\text{nom}}$ para cada nodo. Estos valores nominales son información disponible detallada junto con el modelo de la red de PandaPower. Luego, a partir de estos valores, se genera una distribución de generación/demanda, que consiste en una uniforme entre 0,7 y 1,3 del valor de referencia. Es decir, p_i^{dem} , q_i^{dem} y $p_i^{\text{gen},v}$ se obtienen como:

Capítulo 4. Redes de Transmisión a Estudiar

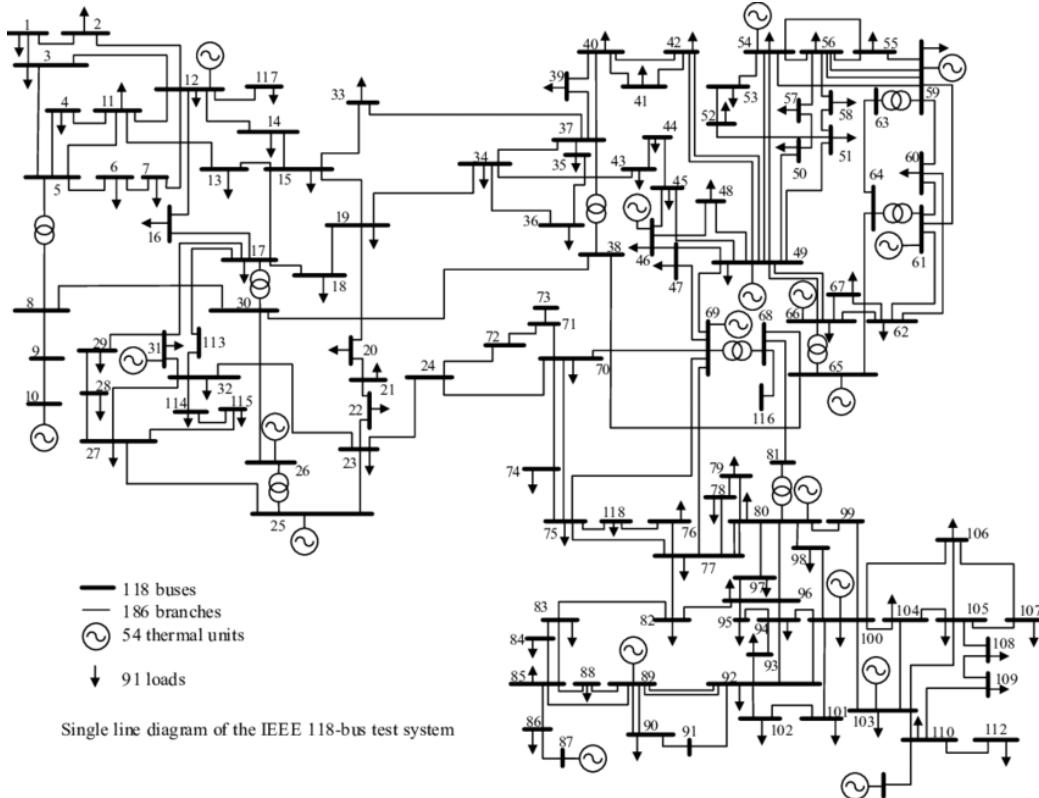


Figura 4.2: Esquema de la red eléctrica IEEE-118 [21].

$$p_i^{\text{dem}} \sim \mathcal{U}(0,7p_i^{\text{dem,nom}}, 1,3p_i^{\text{dem,nom}}) , \quad (4.1)$$

$$q_i^{\text{dem}} \sim \mathcal{U}(0,7q_i^{\text{dem,nom}}, 1,3q_i^{\text{dem,nom}}) , \quad (4.2)$$

$$p_i^{\text{gen},v} \sim \mathcal{U}(0,7p_i^{\text{gen},v,\text{nom}}, 1,3p_i^{\text{gen},v,\text{nom}}) . \quad (4.3)$$

$$(4.4)$$

De esta forma se obtiene una instancia de entrada a la red neuronal para cada barra $x_i = [p_i^{\text{dem}}, q_i^{\text{dem}}, p_i^{\text{dem},v}]$, donde se concatenan para las N barras para formar una instancia de entrada a la red, $\mathbf{x} = [x_1 | x_2 | \dots | x_N]$. Se supone que esta instancia corresponde para un instante de tiempo t . Repitiendo esto un total de $T = 10000$ veces, se obtiene el conjunto de datos de entrada $X = \{\mathbf{x}_t\}_{t \in [0,T]}$. De esta forma, se genera sintéticamente el histórico de conjunto de datos de entrada. La veracidad de generar los datos con una distribución uniforme se estudia más adelante, donde se observa que este modelo resulta simplista en comparación con la distribución real que siguen los datos.

Una vez obtenido el conjunto de entrada, se procede a resolver el despacho óptimo de potencia reactiva para cada una de las entradas \mathbf{x}_t . Para esto se utiliza el optimizador tradicional implementado por PandaPower. Esto se realiza por dos razones. En primer lugar, la estrategia de aprendizaje supervisado precisa de la

4.2. Red de Transmisión Uruguaya

salida para poder aprender el mapeo entrada/salida. En segundo lugar, estas soluciones se utilizan como punto de comparación con las estrategias estudiadas en este trabajo.

A continuación, se guardan las consignas de voltajes en los generadores $\{v_{m,i}\}_{i \in \mathcal{G}}$, que al no tener compensadores de reactiva corresponden a la totalidad de la salida $y_i = [v_{m,i}]$. Se remarca que, en las barras que no tienen generador controlador de voltaje, la salida se rellena con $y_i = 0$. Concatenando para todas las barras se obtiene el vector de salida $\mathbf{y} = [y_1|y_2|\cdots|y_N]$. Repitiendo esto para cada entrada, se obtiene el conjunto de datos completo, $\mathcal{D} = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t \in [0,T]}$. Un detalle a resaltar es que el despacho óptimo de reactiva no converge para el total de las entadas. En caso de que una entrada resulte en un despacho óptimo de reactiva no convergente, se descarta esta muestra. Esto resulta en un total de 6630 datos para la red IEEE30, y un total de 8173 para la IEEE118.

4.2. Red de Transmisión Uruguaya

En esta sección se presenta el modelo de la red eléctrica uruguaya utilizado para estudiar el despacho óptimo de potencia reactiva, junto con su correspondiente base de datos. Para ello, se inicia con una introducción al sector eléctrico de Uruguay, describiendo la matriz de generación energética, la red de transmisión, el mercado eléctrico nacional y los actores principales del proceso de despacho de carga del país. Tras esta introducción, se profundiza en el proceso de construcción del modelo de la red a utilizar. Se discuten los inconvenientes de trabajar directamente con los modelos empleados por los operadores y se detallan una por una las consideraciones necesarias para la creación del modelo. Finalmente, se describe el procedimiento para la obtención y curado de la base de datos, lista para entrenar las redes neuronales.

4.2.1. Introducción al Sector Eléctrico

El sector eléctrico uruguayo es un pilar fundamental en la estructura económica y social del país. Uruguay se destaca en la región por su avanzado modelo de generación eléctrica, que ha evolucionado hacia una matriz energética altamente diversificada y sostenible. Tradicionalmente, la generación de energía en Uruguay dependía en gran medida de las fuentes hidroeléctricas, gracias a la abundancia de recursos hídricos en el país. Sin embargo, en las últimas décadas el país ha dado pasos significativos hacia la incorporación de energías renovables no convencionales, especialmente la eólica y la solar, lo que ha transformado su matriz energética.

Hoy en día, Uruguay es un ejemplo a nivel mundial por su capacidad para generar la mayor parte de su electricidad a partir de fuentes renovables, con más del 95 % de su energía proveniente de estas fuentes. En lo que va del año 2024, la distribución de la generación eléctrica por fuente ha sido la siguiente: un 51.2 % proviene de fuentes hidroeléctricas, un 32.3 % de energía eólica, un 2.8 % de energía solar, un 1.4 % de fuentes térmicas, y un 12.3 % de biomasa [1]. Además, las importaciones de electricidad desde Argentina y Brasil han sido prácticamente despreciables, lo

Capítulo 4. Redes de Transmisión a Estudiar

que subraya la autosuficiencia y la eficiencia del sistema eléctrico uruguayo. Este logro ha sido posible gracias a una combinación de políticas públicas favorables, inversión en infraestructura, y una estrategia de planificación a largo plazo que ha promovido la sostenibilidad y la resiliencia del sistema eléctrico.

El mercado eléctrico uruguayo ha experimentado una transformación significativa desde su surgimiento en el siglo XX, cuando el sector comenzó a expandirse a nivel global. Inicialmente, el desarrollo del sector en Uruguay estuvo centralizado en manos del estado, a través de lo que hoy se llama Administración Nacional de Usinas y Trasmisiones Eléctricas (UTE). Este modelo verticalmente integrado permitió a UTE monopolizar todas las áreas del sector eléctrico, incluyendo la generación, transmisión, distribución y comercialización de la energía a los usuarios finales. Sin embargo, hacia finales del siglo XX, comenzaron a evidenciarse deficiencias que llevaron a la necesidad de reformas estructurales. La Ley N°16832, promulgada en 1997, introdujo un modelo de mercado mayorista eléctrico, donde la energía comenzó a ser tratada como un producto comercializable, permitiendo la entrada de nuevos agentes generadores en el mercado. Este mercado abierto ha sido clave para atraer nuevas inversiones, especialmente en energías renovables, lo que ha potenciado el desarrollo de la infraestructura eléctrica y ha contribuido al éxito del modelo energético uruguayo [15].

Para supervisar y regular el nuevo mercado mayorista, se estableció la Administración del Mercado Eléctrico (ADME), encargada de asegurar la correcta ejecución de los acuerdos comerciales y el despacho eficiente de la demanda eléctrica. Paralelamente, se creó el Despacho Nacional de Carga (DNC), un organismo clave para la gestión en tiempo real de la generación y distribución de energía en Uruguay. El DNC utiliza el sistema SCADA (*Supervisory Control And Data Acquisition*) para monitorear y controlar la red eléctrica del país en tiempo real, permitiendo a los operadores tomar decisiones inmediatas para optimizar el rendimiento de la red y responder a eventos imprevistos. A través de este sistema, el DNC garantiza la estabilidad y seguridad del suministro eléctrico, al tiempo que dispone de datos históricos y parámetros técnicos que son fundamentales para la planificación y operación eficiente del sistema eléctrico uruguayo.

El Despacho Nacional de Carga es fundamental en la gestión del Sistema Interconectado Nacional (SIN), la red que permite la transferencia eficiente de energía desde los centros de generación hasta los consumidores en Uruguay. En 2023, el SIN cuenta con 5,857 kilómetros de líneas y cables de alta tensión, distribuidos en circuitos de 60, 150, 230 y 500 kV, además de 93 subestaciones que transforman el voltaje para la transmisión de energía. La potencia instalada en transformadores en las líneas de transmisión alcanza los 5,326 MVA en 150 kV y 4,280 MVA en 500 kV, mientras que en las máquinas generadoras es de 2,188 MVA y 333 MVA, respectivamente. El SIN incluye importantes interconexiones con Brasil y Argentina, que refuerzan la seguridad del suministro y permiten la participación activa de Uruguay en el mercado eléctrico regional. Así, el DNC y el SIN garantizan una operación segura, eficiente y confiable del sistema eléctrico, apoyando el desarrollo sostenible del país [38].

El análisis y la planificación del sistema eléctrico uruguayo requieren una com-

4.2. Red de Transmisión Uruguaya

prensión detallada y precisa del sistema. Este trabajo se centra en las regiones que operan a 150 kV y 500 kV, que son la columna vertebral de la transmisión de energía en el país. Cuanto más fiel a la realidad sea el modelo utilizado, más significativos serán los resultados obtenidos en los estudios de flujo de carga y posterior aprendizaje. A continuación, se procede al modelado de este sistema.

4.2.2. Modelado de la Red Eléctrica Uruguaya

En la tesis se aborda el problema de aprender el despacho óptimo de reactiva en la red uruguaya a partir de datos históricos. Por lo tanto, es necesario armar un modelo de la red a la cual se le puedan cargar los datos para un instante de tiempo, y correr el despacho óptimo de reactiva. Para este fin, se dispone de dos fuentes principales de información, cada una con sus particularidades y limitaciones.

La primera fuente son los datos del sistema SCADA. Como se mencionó previamente, este sistema de monitoreo y control permite al DNC la operación segura y eficiente del despacho de carga. Cuenta con una representación de la red que permite visualizar los datos recopilados por los sensores en un contexto operativo, mostrando cómo se comportan los diferentes elementos de la red en tiempo real. En lugar de interpretar los datos como simples valores en una tabla, la visualización en SCADA integra estos datos dentro de un esquema gráfico que refleja la estructura física de la red. Sin embargo, esta representación es superficial, en el sentido que no cuenta con detalles técnicos de los elementos de la red, como impedancias en líneas de transmisión o valores máximos y mínimos en los generadores, ya que su principal utilidad reside en el monitoreo y guardado de los datos medidos en diversos instantes de tiempo.

Por otro lado, se cuenta con varias representaciones técnicas de la red en el software PSSE, utilizadas por los operarios para simular diferentes escenarios de interés. Estas sí incluyen los valores de todos los elementos eléctricos y permiten correr el flujo de carga. Los operarios manejan múltiples modelos, cada uno con diferentes niveles de detalle y abstracción, dependiendo del tipo de análisis que se desee realizar. Aunque estos modelos permiten correr simulaciones de despacho óptimo de potencia, los operarios no utilizan estos en el día a día para la resolución del flujo de potencia, sino que principalmente son utilizados para estudios de mejoras de la red, o simulaciones de casos particulares, como puede serlo para ciertos eventos atípicos.

Debido a que no es de utilidad para los operarios del DNC, no se cuenta con ningún modelo en PSSE que se alinee exactamente con la representación del SCADA, permitiendo una alimentación directa de los datos históricos. El primer objetivo en este trabajo es desarrollar un modelo en PSSE (para luego migrar a PandaPower) que permita ejecutar el despacho óptimo de potencia reactiva y que esté alineado con los datos del SCADA. Esto significa que cada elemento de la red de SCADA esté plenamente identificado dentro de los datos del modelo de PSSE, de forma tal de hacer la correspondencia entre los datos históricos y los parámetros técnicos de los elementos de la red. Así, se busca un sistema que pueda ejecutar flujos de carga y aprovechar los datos históricos.

Capítulo 4. Redes de Transmisión a Estudiar

Implementación

Para construir este modelo, se parte de una de las representaciones en PSSE provista por los operarios. Si bien fue desafiante hacer la correspondencia con las cargas y los generadores, se obtuvo un modelo PSSE donde las barras, líneas y transformadores en ambas representaciones se mantienen sin cambios. Esto se debe a que presentan una configuración y nomenclatura casi idénticas a las utilizadas en el SCADA. Existen pequeñas diferencias, como barras intermedias auxiliares sin elementos conectados que están en PSSE pero no en SCADA (*e.g.*, dos barras auxiliares en Arbolito), o barras que tienen nombres diferentes (*e.g.*, Sierras del Río en SCADA se denomina Ladaner en PSSE). Mientras se tengan en cuenta al conectar los elementos en las barras, estas diferencias no representan un problema para la carga de datos en el modelo.

Los datos históricos disponibles consisten en dos archivos tabulares: el primero contiene la generación de potencia activa por generador y el segundo la demanda de potencia activa total por barra. Cada columna corresponde a un generador o barra (según el archivo) y cada fila a una fecha específica. En la próxima subsección se profundiza en el manejo de estos datos, pero por ahora es necesario saber qué cargas y generadores están en el histórico de datos para asegurar su existencia y correspondencia dentro del modelo.

El trabajo comienza con el archivo de demandas. Dado que los datos están agrupados por barra, se debe asegurar que todas las barras del archivo tengan un elemento de tipo carga conectado. El modelo en PSSE ya incluye varias cargas predefinidas, pero no se corresponden exactamente con las de los datos históricos. Esto se debe a que algunas cargas son despreciadas por ser pequeñas o a que ciertas barras tienen la demanda total dividida en varias cargas conectadas. Se realizan modificaciones donde sea necesario para obtener una carga por barra, correspondiendo así las cargas del modelo con las de los datos.

Se continúa con el archivo de generación. Este caso es significativamente más complejo. En primer lugar, la nomenclatura utilizada es distinta. En PSSE, los generadores tienen un nombre predeterminado (*e.g.*, “GEN1”), mientras que en SCADA usan un acrónimo del nombre del generador (*e.g.*, “MELOWIND”). Esta diferencia complica la correspondencia entre los generadores del modelo y los datos, por lo que se opta por eliminar los generadores presentes en el modelo e ir agregando uno por uno los presentes en el archivo de datos de generación. Para incluir los generadores en el modelo, tal como se especifica en el capítulo 2, es necesario conocer a qué barra pertenece el generador y cuáles son sus límites de potencia reactiva mínima y máxima. Aunque los datos de potencia activa límites no son imprescindibles, ya que esta se toma como entrada en el despacho óptimo de reactiva, se añaden al modelo de los generadores por completitud.

Para determinar a qué barra pertenece cada generador, se utiliza un mapa unifilar de acceso abierto publicado por ADME [2]. Este consiste en una representación general de las barras, líneas, generadores y transformadores de la red, utilizando una nomenclatura muy similar a la de SCADA. Por otro lado, los límites de potencia se obtienen de un archivo que contiene la curva de capacidad de los generadores, proporcionado por los operarios del DNC. Cabe destacar que algunos

4.2. Red de Transmisión Uruguaya

generadores del archivo de SCADA no aparecen en el mapa o en el archivo de curvas de capacidad, en cuyo caso se solicitó información adicional a los operarios del DNC para completar los datos. De esta forma, todos los datos de generación tienen un generador correspondiente en el modelo. El modelo se simplifica agrupando los generadores de una misma barra en un único generador, en concordancia con lo presentado en el capítulo 2. Para esto, las barras que presentan múltiples generadores se sustituyen por un único generador, cuyos límites de potencia se obtienen sumando los límites individuales. Es importante tener esto en cuenta a futuro al cargar los datos de generación, donde será necesario sumar las columnas de generadores de una misma barra para tener un archivo de generación por barra, en lugar de por generador. Esto se explica con más detalle más adelante.

El último elemento a modelar son los compensadores de reactiva. Dado que la reactiva que producen no es un dato de entrada, sino una variable de control, no existe un archivo de datos al cual corresponder. Por lo tanto, se dejan los compensadores predefinidos en el modelo. Estos corresponden a seis compensadores ubicados en las barras de Montevideo A, Montevideo I, Melo B, San Carlos, Palmar y Salto Grande Uruguay.

De esta forma, se obtiene un primer modelo en el que es posible cargar los datos históricos y ejecutar un flujo óptimo de carga. Sin embargo, aún hay varios detalles del modelo que es necesario modificar para que el caso de estudio sea lo más relevante posible.

En primer lugar, se debe decidir qué generador tomar como referencia. Como se explicó previamente, es necesario elegir una barra como referencia, que permita absorber las diferencias de potencia entre generación y demanda, para que las simulaciones cierren. Es estándar utilizar como referencia el generador con mayor capacidad, o un punto de interconexión con una red extranjera. Por ello, el candidato más adecuado es la represa hidráulica de Salto Grande. Este generador es compartido entre Uruguay y Argentina, con una capacidad total de hasta 1890 MW, energía que se divide en partes iguales entre ambos países. Por lo tanto, no solo representa el generador con mayor capacidad del país, sino que también es un punto de interconexión con Argentina.

Otra decisión importante es cuál generador utilizar como controlador de voltaje y cuál no. Primero se consideran las restricciones técnicas de los generadores, es decir, cuáles permiten asignarles una consigna de voltaje y cuáles no. Aquellos en los que no es posible hacerlo se definen directamente como generadores estáticos. En particular, todos los generadores solares y algunos eólicos no permiten asignarles una consigna. En el caso de los que sí aceptan consignas, no todos son utilizados para controlar voltaje. Comenzando por los generadores térmicos, estos son una fuente no renovable y de alto costo, y su uso es limitado y está reservado para casos de emergencia. Además, en los pocos casos en los que se activan, es necesario extraer la mayor cantidad de potencia activa posible, lo que reduce significativamente su capacidad de producir reactiva. Por lo tanto, utilizar estos generadores para controlar reactiva resulta ineficiente. La excepción son dos generadores térmicos, CTR1 y CTR2, que no se utilizan como generadores, sino como máquinas sincrónicas. Estos están en la misma barra y su función principal

Capítulo 4. Redes de Transmisión a Estudiar

es consumir reactiva. Por lo tanto, de los térmicos, solo estos dos se utilizan como controladores de voltaje, mientras que el resto se consideran estáticos. En cuanto a los generadores de biomasa, aunque son renovables, su costo también es elevado, por lo que se tratan de manera similar a los térmicos, considerándolos estáticos.

Finalmente, se tienen los generadores hidráulicos y parte de los eólicos. Todos los hidráulicos, excepto Salto Grande que se usa como referencia, se modelan como controladores de voltaje. Estos son tres: las centrales hidráulicas de Baygorria, Bonete y Palmar. Respecto a los eólicos, existen casos donde dos parques eólicos en barras adyacentes permiten control de voltaje. Aunque en teoría se pueden utilizar ambos, en la práctica esto resulta en interferencia entre los controladores. En estos casos, los operarios utilizan solo uno de estos parques eólicos como controlador de voltaje, mientras que al otro se le fija la potencia reactiva. Considerando esta particularidad, se llega a un total de 15 parques eólicos que permiten controlar voltaje, distribuidos en 11 barras. Resumiendo, se tiene un total de 15 barras de tipo PV, de las cuales una corresponde a generadores térmicos, tres a hidráulicos y 11 a eólicos.

La última decisión sobre el modelo involucra los límites de voltaje en las barras. Los operarios tienen como directriz mantener los voltajes entre $0,93 \text{ p.u}$ y $1,07 \text{ p.u}$ por unidad en las barras de 150 kV , mientras que en las barras de 500 kV deben respetarse los límites de $0,95 \text{ p.u}$. y $1,05 \text{ p.u}$. por unidad. Para la mayoría de las barras, se utilizan estos valores. Sin embargo, aunque estos son los límites de seguridad, hay dos barras en las que es deseable restringir aún más la variabilidad del voltaje. Estas son las barras correspondientes a la central Montevideo A (MVA_150) y a la central hidráulica de Bonete (BON_150). Los operarios buscan mantener voltajes cercanos a 154 kV (o $1,026 \text{ p.u.}$) y 157 kV (o $1,046 \text{ p.u.}$) respectivamente. Por lo tanto, para estas barras se limitan los valores de voltaje en el rango $[1,02; 1,03] \text{ p.u.}$ y $[1,03; 1,04] \text{ p.u.}$.

Modelo Final

Se observa un esquemático unifilar del modelo final en la Figura 4.3. Este consiste en $|\mathcal{B}| = 107$ barras, de las cuales 95 son en 150kV y 12 en 500kV . A su vez se tienen $|\mathcal{L}| = 144$ líneas, de las cuales 14 conectan barras de 500kV y 130 conectan barras de 150kV . Todos las barras de 500 tienen asociadas un transformador, que las conecta a una barra del mismo nombre pero a 150, por lo que $|\mathcal{T}| = 12$. Se cuenta con 43 generadores, de los cuales $|\mathcal{G}| = 15$ son controladores de voltaje, $|\mathcal{E}| = 27$ son estáticos y $|\mathcal{X}| = 1$ es generador de referencia. Finalmente, se tienen $|\mathcal{C}| = 55$ cargas y $|\mathcal{R}| = 6$ compensadores de reactiva.

Migración a PandaPower

Una vez que el modelo de la red eléctrica uruguaya ha sido creado y ajustado en el software PSSE, el siguiente paso es trasladar este modelo a la librería de Python PandaPower. Dado que esta biblioteca no incluye una utilidad para importar modelos a partir de archivos crudos exportados del PSSE, es necesario implementar el código para trasladarlo. Para lograr esto, se recurre al manual de usuario del

4.2. Red de Transmisión Uruguaya

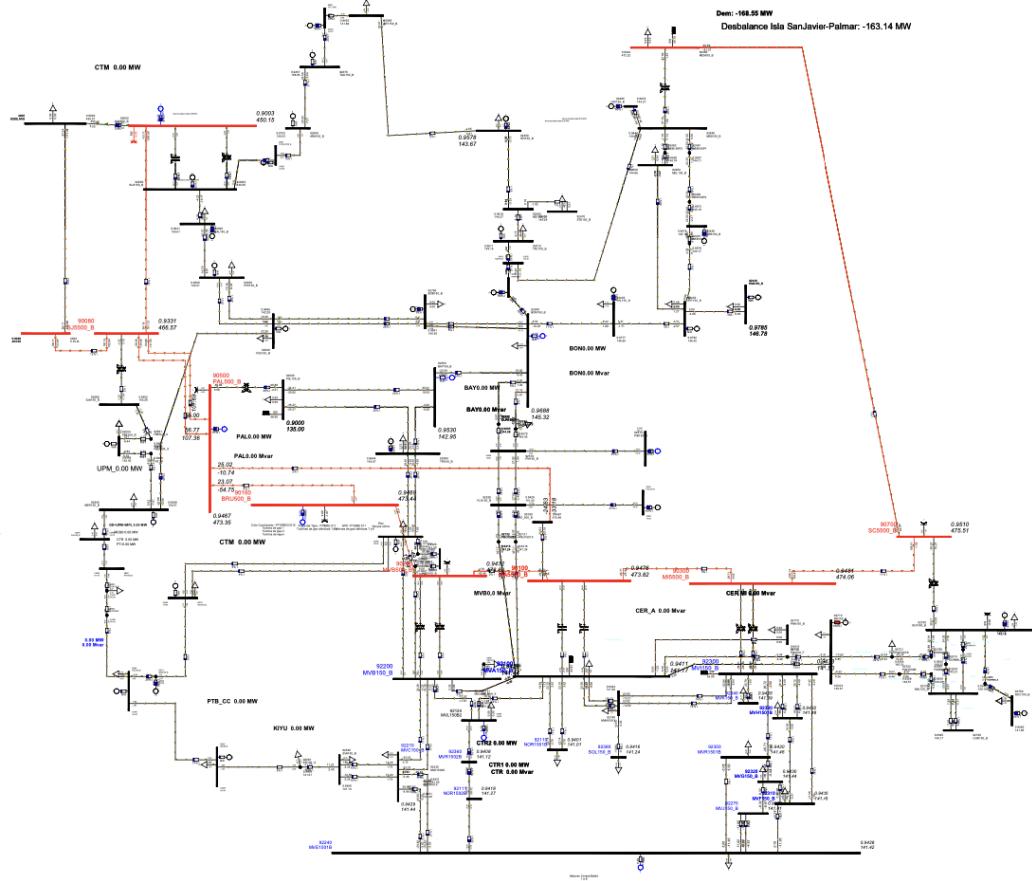


Figura 4.3: Diagrama en PSSE de la red eléctrica uruguaya.

PSSE, el cual detalla la estructura de estos archivos, describiendo en qué orden se presentan los diferentes elementos de la red, como buses, generadores, líneas de transmisión y cargas, así como el significado de cada columna en las tablas que lo componen. Además, es necesario tener ciertas consideraciones en cuenta debido a diferencias en cómo se modelan elementos en ambos softwares. Por ejemplo, mientras que en PSSE las impedancias se expresan en valores por unidad (*p.u.*), en PandaPower estos valores deben ser convertidos a ohmios.

4.2.3. Procesado de Datos

Similar para el caso de las redes IEEE, es necesario armar una base de datos para luego poder entrenar los modelos de aprendizaje automático. Esto implica, en un primer lugar, obtener el conjunto de datos de entrada X , donde un dato $\mathbf{x}_t = [x_{t,1}, x_{t,2}, \dots, x_{t,N}]$ corresponde a concatenar la señal

$$x_{t,i} = [p_{t,i}^{\text{dem}}, q_{t,i}^{\text{dem}}, p_{t,i}^{\text{gen,est}}, q_{t,i}^{\text{gen,est}}, p_{t,i}^{\text{gen,v}}]$$

para las N barras de la red en un instante de tiempo t . A diferencia de las redes IEEE, en este caso se va a armar tanto una base de datos sintética como una real.

Capítulo 4. Redes de Transmisión a Estudiar

Comenzando por la base de datos real, en la subsección anterior se menciona que se cuenta con dos archivos tabulares, uno de generación y otro de consumo de potencia activa. El primero tiene una columna por generador, y una fila por instante de tiempo. Las fechas datan del 1º de enero de 2021 al 31 de diciembre de 2023, con espaciados por intervalos de 1 hora. Esto da un total de 26280 datos. En el caso de las cargas, las columnas representan la potencia consumida total por la barra, para aquellas que presentan cargas. En este caso las fechas datan del 21 de marzo de 2021 al 17 de octubre de 2023 y están espaciadas en intervalos de 10 minutos. Esto da un total de 135504 datos.

El primer paso del curado consiste en analizar y corregir datos faltantes o erróneos. Los datos faltantes corresponden a aquellos casos donde el generador o carga no existían en ese momento. Estos son muy pocos y se resuelven rellenándolos con cero. Un ejemplo de esto es UPM2, que comenzó a operar en abril de 2023. Por otro lado, se llevó a cabo un análisis para identificar valores anormalmente altos. Los datos considerados atípicos fueron reemplazados por el valor válido más cercano anterior. Sin embargo, la cantidad de estos casos es insignificante en relación al conjunto total de datos. Un ejemplo concreto de este fenómeno se observa el 10 de enero de 2022, entre las 10:10 y las 11:00 horas, cuando la demanda eléctrica alcanzó entre 3000 MW y 4000 MW, muy por encima de los máximos usuales. Al investigar la causa, se detectó que la estación Libertad mostró valores de demanda superiores a los 1000 MW, algo claramente anormal. Para corregir este error, se asignó un valor más razonable, cercano a los 18 MW, que es el valor típico anterior y posterior al período de error.

El siguiente paso consiste en agrupar las columnas del archivo de generación por barras. Para esto se suman las columnas de generadores que pertenecen a una misma barra, información recabada en la subsección anterior. En el caso de las cargas no es necesario, ya que los datos vienen agregados por barra.

De esta forma se tienen curados los datos de entrada $p_{t,i}^{\text{dem}}$, $p_{t,i}^{\text{gen},v}$ y $p_{t,i}^{\text{gen},\text{est}}$ para cada barra del modelo. Desafortunadamente, los datos almacenados en el DNC no cuentan con información sobre la potencia reactiva, ni de generación ni de demanda. Por lo tanto es necesario simular estos datos. Para esto se discute con operarios del DNC sobre qué modelo tomar para simular la potencia reactiva, y la fidelidad de estos con la realidad. En el caso de la demanda, la forma de simular la potencia reactiva es utilizar un factor de potencia inductivo de $FP = 0,995$ para los datos en horas de la madrugada (00:00 h a 06:00 h) y $FP = 0,980$ el resto del día. Esta estrategia es la que utilizan frecuentemente los operarios cuando se precisa estimar la reactiva en la demanda. La diferencia entre las horas de la madrugada y el resto del día se debe a que durante la madrugada la demanda es significativamente menor, lo que disminuye los efectos inductivos en las líneas de distribución, provocando un factor de potencia muy cercano a uno o, en algunos casos, hasta ser capacitivo. Matemáticamente, la reactiva demandada queda:

$$q_{t,i}^{\text{dem}} = \begin{cases} p_{t,i}^{\text{dem}} \tan(\arccos(0,980)), & \text{si } 06:00 \leq \text{hora} < 24:00 \\ p_{t,i}^{\text{dem}} \tan(\arccos(0,995)), & \text{si } 00:00 \leq \text{hora} < 06:00 \end{cases}. \quad (4.5)$$

4.2. Red de Transmisión Uruguaya

Para la reactiva en los generadores estáticos, se opta por un método sencillo por sobre uno que sea fuertemente fiel a la realidad. Se decide utilizar para todos los generadores estáticos:

$$q_{t,i}^{\text{gen,est}} = 0 .$$

De esta forma, se obtiene el conjunto de entrada X para la base de datos histórica. Para el caso de la base de datos sintética, es necesario, en primer lugar, obtener valores de generación y demanda nominales de forma de centrar la distribución uniforme. Estos se obtienen promediando cada característica de cada barra respecto del tiempo. Esto es:

$$p_i^{\text{dem,nom}} = \sum_{t=0}^{t=T} p_{t,i}^{\text{dem}} , \quad (4.6)$$

$$q_i^{\text{dem,nom}} = \sum_{t=0}^{t=T} q_{t,i}^{\text{dem}} , \quad (4.7)$$

$$p_i^{\text{gen,est,nom}} = \sum_{t=0}^{t=T} p_{t,i}^{\text{gen,est}} , \quad (4.8)$$

$$p_i^{\text{gen,v,nom}} = \sum_{t=0}^{t=T} p_{t,i}^{\text{gen,v}} . \quad (4.9)$$

Luego, similar que para los casos de la IEEE, la base de datos se genera muestreando 10000 muestras de las siguiente distribución:

$$p_{t,i,\text{sint}}^{\text{dem}} \sim \mathcal{U}(0,7p_i^{\text{dem,nom}}, 1,3p_i^{\text{dem,nom}}) , \quad (4.10)$$

$$q_{t,i,\text{sint}}^{\text{dem}} \sim \mathcal{U}(0,7q_i^{\text{dem,nom}}, 1,3q_i^{\text{dem,nom}}) , \quad (4.11)$$

$$p_{t,i,\text{sint}}^{\text{gen,est}} \sim \mathcal{U}(0,7p_i^{\text{gen,est,nom}}, 1,3p_i^{\text{gen,est,nom}}) , \quad (4.12)$$

$$p_{t,i,\text{sint}}^{\text{gen,v}} \sim \mathcal{U}(0,7p_i^{\text{gen,v,nom}}, 1,3p_i^{\text{gen,v,nom}}) . \quad (4.13)$$

$$(4.14)$$

Finalmente, tanto para la base de datos real como la sintética, se resuelve para cada entrada \mathbf{x}_t el despacho óptimo de reactiva utilizando la librería PandaPower. En este caso, aparte de las consignas de voltaje, se observa la reactiva en los compensadores, resultando en $y_{t,i} = [v_{m,i}, q_i^{\text{comp}}]$. Concatenando para todas las barras se obtiene la salida $\mathbf{y}_t = [y_1 | y_2 | \dots | y_N]$. De esta forma, quedan definidas las bases de datos a utilizar. El siguiente capítulo se dedicará a analizar los datos, antes de reportar el desempeño de predicción de óptimos.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Análisis de Red y Datos

5.1. Análisis de Datos

Una vez recopilados todos los datos necesarios para la resolución del problema, se presenta un análisis sobre los datos disponibles. Para esto, se comienza con un estudio general de los datos de generación y demanda de activa. Se compara la demanda y la generación total como primera validación de coherencia. Luego, se estudian las distribuciones que siguen las cargas y generadores. Esto permite comparar la metodología de simulación de datos utilizada contra el escenario real. Por último, se analizan las salidas óptimas obtenidas luego de correr el despacho óptimo de reactiva.

Resulta natural realizar un chequeo general de manera de verificar la consistencia de los datos. Esto es, para un instante dado, hacer la suma de todos los datos de consumo y compararlos con la suma de generación. Teóricamente, ambos valores deberían ser muy parecidos, con la diferencia dada por las pérdidas en la red. Sin embargo, esto no es lo que sucede. En la Figura 5.1 se puede ver un histograma que muestra las diferencias entre potencias generadas y consumidas para todos los datos históricos. Se logra observar que existe cierta discrepancia entre cada valor de generación y consumo, con una distribución aproximadamente normal. Esto puede deberse a varias razones. Primero, las pérdidas de la red causan que siempre se tenga que generar más de lo que en realidad es consumido por los consumidores. Esta es la razón por la cual el centro de este histograma se encuentra a la derecha del cero.

Por otro lado, las diferencias entre la generación y el consumo pueden atribuirse a errores en los datos, los cuales pueden surgir tanto de imprecisiones en las mediciones como de la ausencia de información sobre algunos generadores y cargas. Esta falta de datos puede deberse a valores faltantes o a que ciertas cargas o generadores no se incluyen debido a que su potencia es considerada despreciable. Por último, errores o imprecisiones del modelo de la red eléctrica generado pueden causar una discrepancia entre valores de carga y generación. De todas maneras, como fue mencionado anteriormente, se tiene una barra de referencia que se encarga de generar o consumir el remanente de potencia para que cierren perfectamente

Capítulo 5. Análisis de Red y Datos

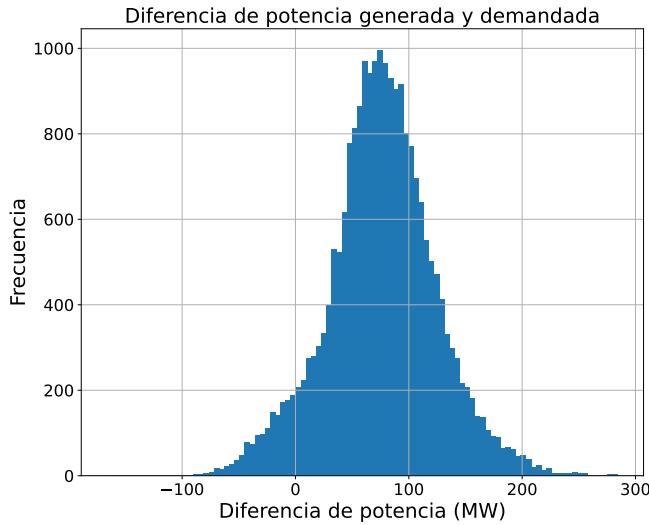


Figura 5.1: Histograma de $P_{GEN_{TOT}} - P_{LOAD_{TOT}}$

las ecuaciones de la red.

5.1.1. Distribución de los Datos

Analizar la distribución de los datos históricos de la red uruguaya es clave, por un lado, para aprender sobre los patrones en los datos de la red real y, por otro, para compararlos con los datos sintéticos generados. Esto permite evaluar si el enfoque de generación de datos que se utilizó es el más adecuado o si sería mejor hacerlo de otra forma.

Datos de Generación

A modo de recordatorio, la distribución de los datos simulados en la red IEEE y uruguaya sintética consiste en una distribución uniforme centrada en un valor nominal. En la Figura 5.2 se observan histogramas de la potencia activa de 3 generadores en la red uruguaya en la que se utiliza generación sintética. A continuación, se analizan las distribuciones de los datos históricos de la red uruguaya y se comparan con esta estrategia de generación.

Para empezar, es interesante ver cómo se distribuyen los datos según los distintos tipos de generación: eólica, solar, biomasa, hidráulica y térmica. Separar los datos por tipo de generador da una visión más clara de cómo funciona el sistema de generación en la realidad. Para cada tipo de generación se presentan histogramas de los 4 generadores que mayor inyección de potencia activa han tenido en el histórico de datos disponible. En las figuras, están en orden de mayor a menor generación total, de izquierda a derecha y arriba a abajo.

En la Figura 5.3 se pueden ver los histogramas de los generadores eólicos. El parque eólico Pampa sobresale como el más influyente, seguido por Palomas, Va-

5.1. Análisis de Datos

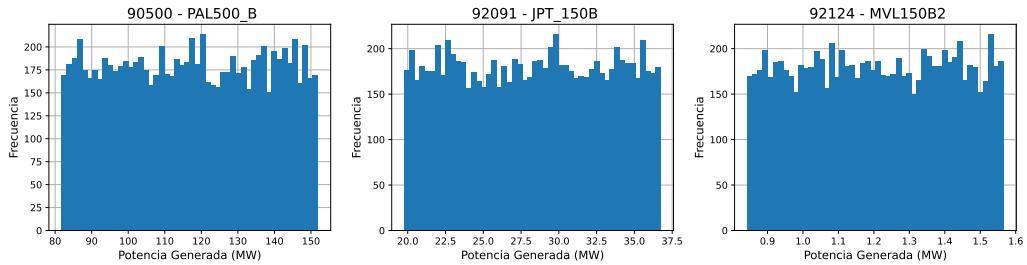


Figura 5.2: Histograma de generación de potencia activa en 3 generadores de la red uruguaya con datos sintéticos.

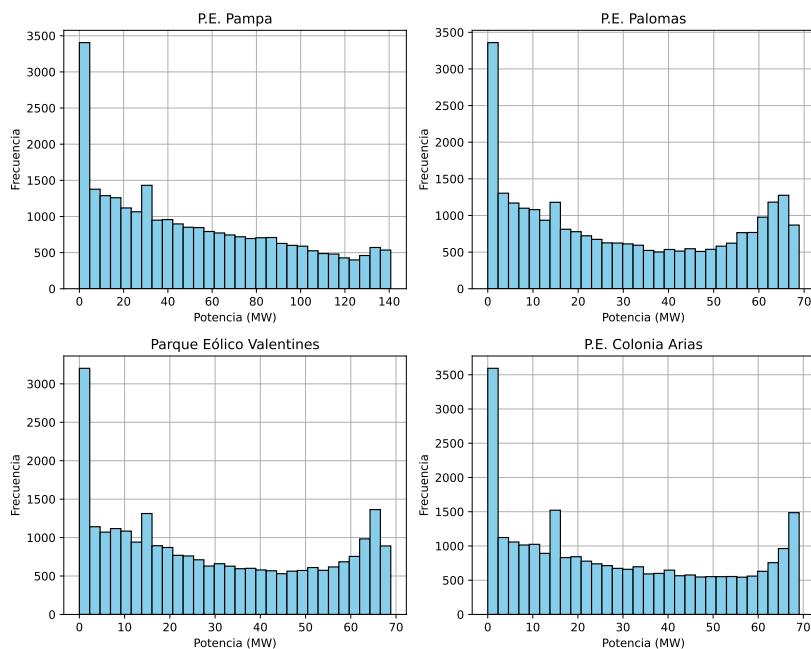


Figura 5.3: Histograma de generación de potencia activa de los 4 generadores eólicos más influyentes de la red.

lentines y Colonia Arias. Todos ellos muestran un pico en cero, correspondiéndose a momentos que están apagados, muy posiblemente debido a la falta de viento. Durante funcionamiento, la potencia que generan está distribuida de forma aproximadamente uniforme entre 0 y el valor máximo.

En la Figura 5.4 se presentan los histogramas de los generadores hidráulicos. Salto Grande se destaca claramente como el generador con la mayor potencia inyectada, y se observa que rara vez se encuentra entregando potencia nula. Por otro lado, los siguientes tres generadores más importantes, Palmar, Rincón del Bonete y Rincón de Baygorria, están apagados con mayor frecuencia. La distribución para estos generadores parece ser en dos tipos de regímenes; en alta o en baja generación. En alta generación se encuentra produciendo valores muy cercanos al máximo, mientras que en baja se encuentra con valores cercanos a cero respecto

Capítulo 5. Análisis de Red y Datos

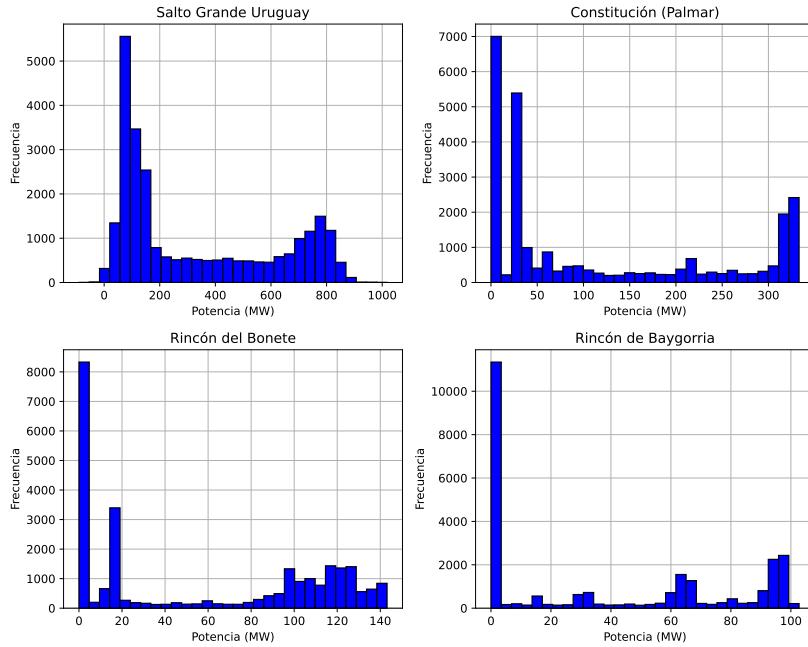


Figura 5.4: Histograma de generación de potencia activa de los 4 generadores hidráulicos más influyentes de la red.

del rango posible de generación

En la Figura 5.5 se muestran los histogramas de los parques fotovoltaicos. El Naranjal y La Jacinta destacan como los parques con la mayor potencia inyectada, seguidos por Alto Cielo y Menafra Solar. Como es de esperar, una gran parte de la generación se encuentra en cero, ya que durante la noche y en días muy nublados no hay producción. En cambio, cuando los parques están generando, la potencia inyectada tiende a distribuirse de manera aproximadamente uniforme entre los valores mínimos y máximos del parque fotovoltaico.

Respecto a los generadores de biomasa, en la Figura 5.6 se presentan los histogramas de los generadores. Montes del Plata se destaca con una producción significativamente superior en los últimos años. Le siguen Liderdat, UPM y UPM2. Se observa alta frecuencia en valores de no generación, mientras que en funcionamiento, la generación suele ser variada.

Por último, en la Figura 5.7 se observan los histogramas de los generadores térmicos. Como es de esperar, estos generadores permanecen apagados la mayor parte del tiempo. Esto indica que su uso se reserva para situaciones en las que es necesario aumentar la producción y satisfacer la demanda cuando las fuentes de energía renovable no son suficientes.

Al comparar la distribución de datos generados sintéticamente con la distribución observada en los datos históricos, se destacan varias diferencias importantes que reflejan las particularidades de cada sistema. En el caso de los datos sintéticos, la generación de datos sigue una distribución uniforme centrada en un valor inicial para cada generador. Esto es útil para evaluar el rendimiento de algoritmos

5.1. Análisis de Datos

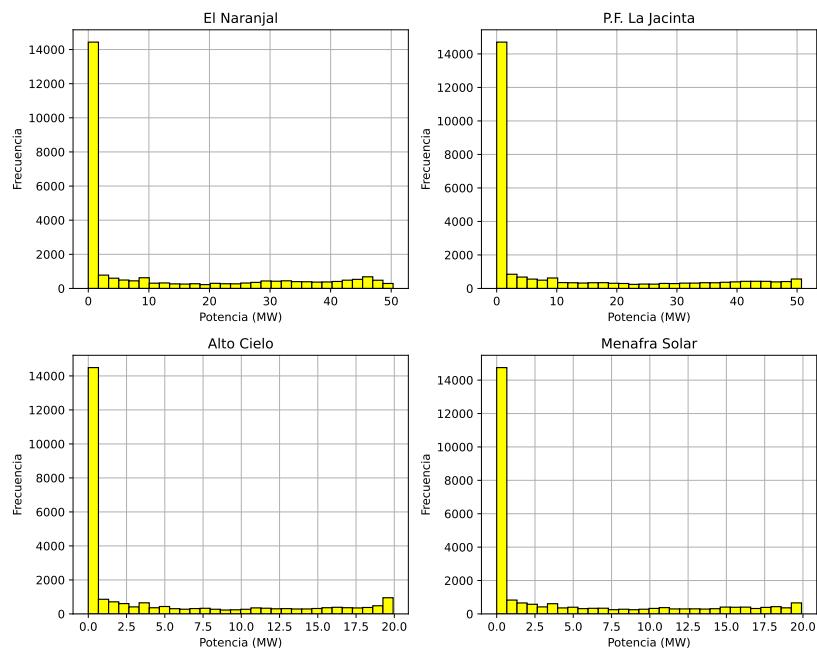


Figura 5.5: Histograma de generación de potencia activa de los 4 parques fotovoltaicos más influyentes de la red.

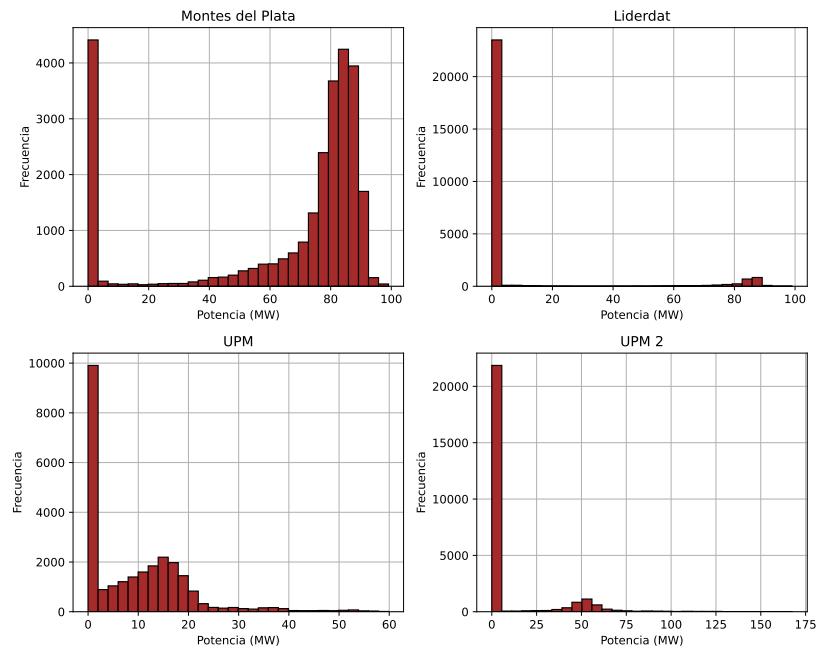


Figura 5.6: Histograma de generación de potencia activa de los 4 generadores biomasa más influyentes de la red.

Capítulo 5. Análisis de Red y Datos

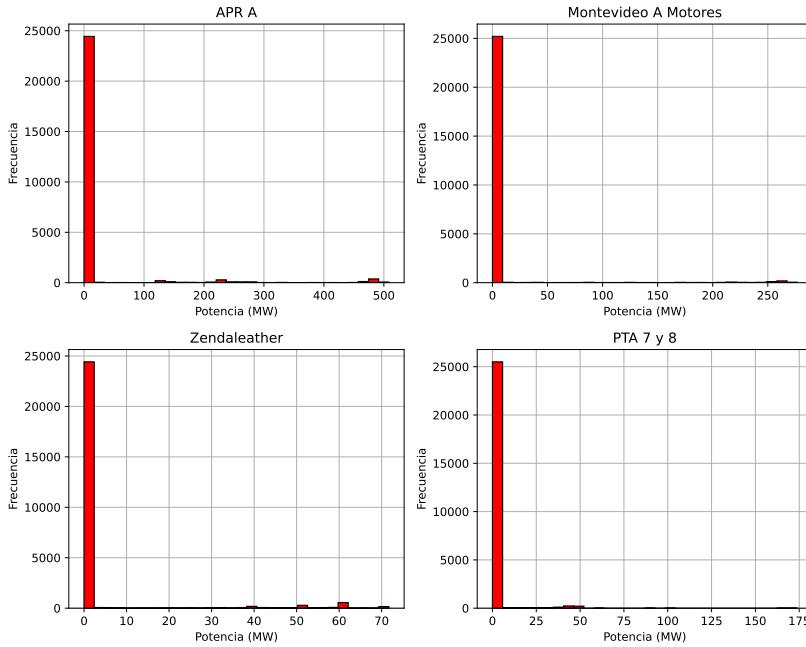


Figura 5.7: Histograma de generación de potencia activa de los 4 generadores térmicos más influyentes de la red.

y sistemas de control alrededor de un punto de operación estándar. Sin embargo, la realidad de los datos históricos es mucho más compleja.

Los datos reales muestran una alta variabilidad en las distribuciones dependiendo del tipo de generación. Por ejemplo, los generadores eólicos y solares tienen una alta prevalencia de períodos sin generación debido a la naturaleza de sus fuentes de energía. Los generadores hidráulicos presentan una distribución más constante, pero con diferencias significativas en la generación entre las distintas plantas. Los generadores térmicos, por su parte, están mayormente apagados, utilizándose solo en momentos específicos de alta demanda. Esta complejidad no se captura adecuadamente con el modelo de datos sintético, lo que sugiere que, para aplicaciones en redes reales, es necesario desarrollar métodos de generación de datos más complejos. Esto permitiría una mejor simulación y análisis del comportamiento real de la red, mejorando la toma de decisiones y la planificación en el sistema eléctrico.

Aparte de las diferencias mencionadas, un factor altamente importante a la hora de simular datos es la variabilidad cíclica a lo largo del tiempo. Por ejemplo, en invierno y verano el consumo de energía es notoriamente mayor que en otoño y primavera, lo que se refleja en un aumento en la generación. Además, algunos generadores presentan mayor actividad en una estación del año que en otra, debido a la naturaleza de su funcionamiento. En la Figura 5.8, se presentan ejemplos que evidencian estas complejidades en los generadores que mayor producción han tenido, separados según el tipo de generación. Se muestra para cada año la generación media, en función de la hora y la estación del año.

Como primer ejemplo, se muestra el caso del parque eólico Pampa. Presen-

5.1. Análisis de Datos

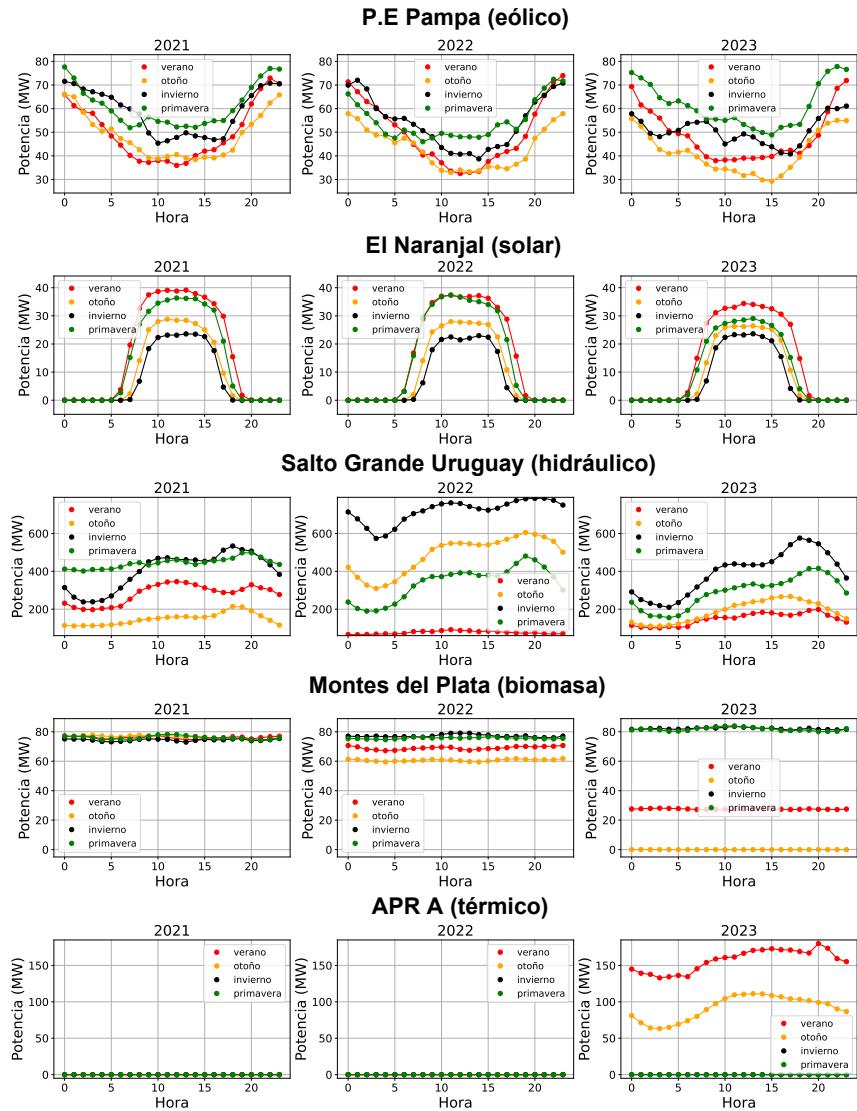


Figura 5.8: Generación de potencia activa promedio por año, estación y hora para cada generador.

ta una generación relativamente constante a lo largo de los años, con una ligera superioridad en primavera e invierno. Se destaca que entre las 10 y 15 horas se registra la menor generación promedio, mientras que la mayor se concentra entre las 20 y las 05 horas. Este comportamiento se debe a la fuente de energía eólica, que depende del viento y se genera de manera continua. En este aspecto, la gene-

Capítulo 5. Análisis de Red y Datos

ración solar muestra un patrón similar. El Naranjal produce energía entre las 06 y las 19 horas, cuando el sol está presente. Además, se observa que la generación máxima ocurre en verano, seguido de primavera, otoño e invierno. Por otro lado, la generación hidráulica tiende a seguir el comportamiento de la demanda, debido a que suele ser la opción de despacho para complementar la generación de solar y eólica, las cuales están siempre generando al máximo de sus capacidades. En la figura se muestran los gráficos para Salto Grande Uruguay, donde se evidencia un aumento de la generación durante las horas pico de consumo, entre las 17 y las 21 horas, y una disminución en horas de la madrugada. En cuanto a las diferencias estacionales, es común que en verano haya menos generación hidráulica debido a la menor reserva de agua. Por otro lado, la generación de biomasa en Montes del Plata fue relativamente constante en 2021 y 2022, como se observa en los gráficos. Sin embargo, en 2023 se produjo una disminución en otoño y verano, posiblemente debido a la interrupción de la generación en esas fechas. Finalmente, la generación térmica en el generador APR A presenta una particularidad. No se registró generación en 2021 ni en 2022, pero en verano y otoño de 2023 se observa un aumento significativo. Esto se debe a que Uruguay enfrentó una sequía importante en ese período, lo que obligó a encender los generadores térmicos, incluyendo APR A, para satisfacer la demanda.

Carga

Al igual que para generación, es importante estudiar la distribución que siguen las cargas de la red eléctrica, no solo para entender mejor el funcionamiento de la red, sino también para evaluar si el enfoque de generación de datos de carga que se utilizó para las redes IEEE es razonable.

En la Figura 5.9 se presenta la distribución de las 9 cargas que implican el mayor drenaje de potencia activa en la red uruguaya. Primero, se destaca la curiosidad de que todas ellas corresponden al área metropolitana en Montevideo, excepto Pando, Solymar y Maldonado. Respecto a la distribución, se observa que estas cargas siguen una distribución similar a una normal, con cierta asimetría en la caída de la mitad derecha de la distribución.

Al comparar la distribución de datos de carga activa generados sintéticamente con la distribución observada en los datos históricos de la red eléctrica uruguaya, existen algunas diferencias. En las redes con datos sintéticos, la carga activa se generó mediante un muestreo uniforme centrado en un valor de referencia para cada carga. Este método es útil en simulaciones estándar, sin embargo, los datos reales de la red uruguaya muestran que las cargas no siguen una distribución uniforme, sino más bien una distribución que tiende a ser cercana a la normal, con variabilidad dependiendo del tipo de carga y su ubicación dentro de la red. De esta forma, se concluye que una mejor forma de generar datos de carga activa en una red eléctrica es usando distribuciones normales en lugar de uniformes.

Por otro lado, en la Figura 5.10 se observa la suma de todas las cargas según el año, estación y hora. Al igual que para los datos de generación, los datos de carga dependen de la fecha y hora. En verano e invierno se tiene una mayor carga que en otoño y primavera, a lo largo de todo el día. Además, se observa como

5.1. Análisis de Datos

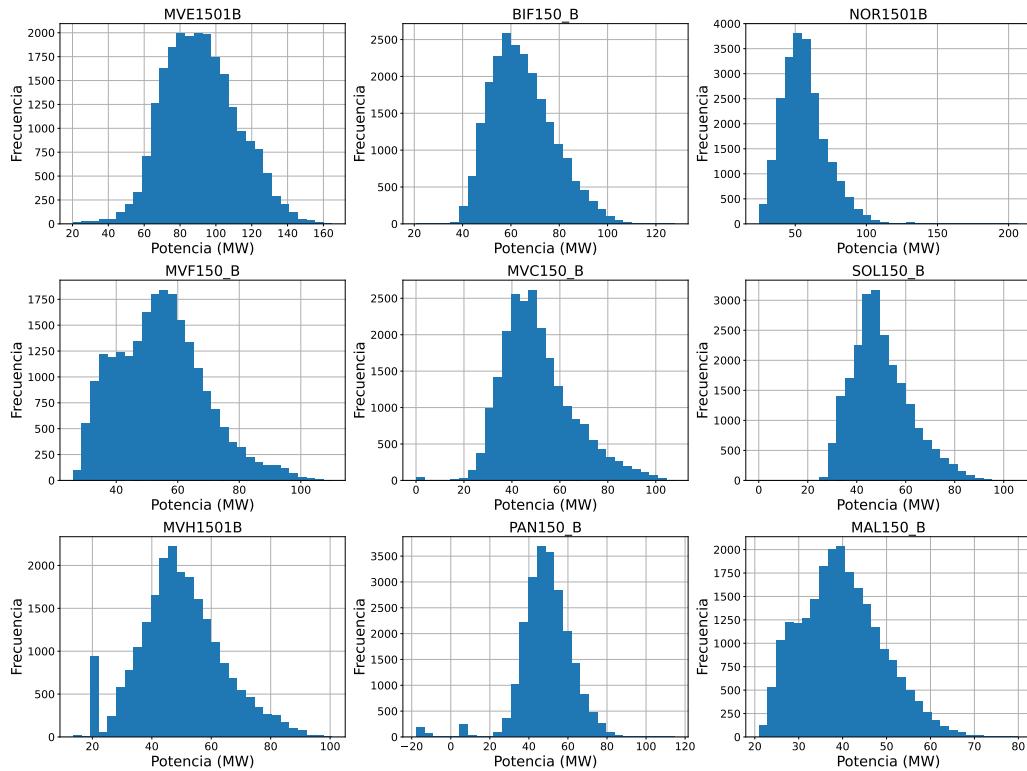


Figura 5.9: Distribución de las 9 cargas con mayor consumo en la red uruguaya. Estas corresponden, en orden de izquierda a derecha y descendiente, a: Central de Montevideo E, Bifurcación, Central Norte, Central de Montevideo F, Central de Montevideo C, Solymar, Central de Montevideo H, Pando y Maldonado.

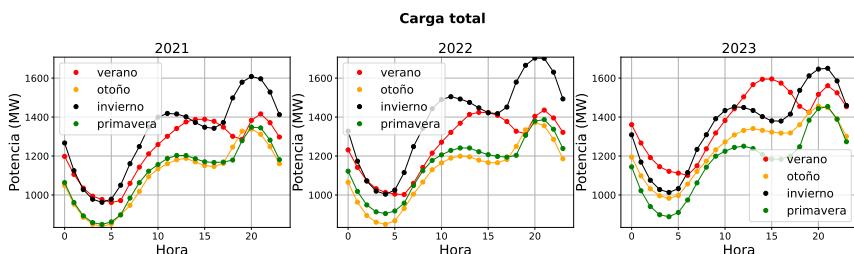


Figura 5.10: Carga total por año, estación y hora.

varía en función de las horas. En la madrugada el consumo es mínimo y sube hasta un primer pico alrededor de las 12:00 del medio día. Luego baja y alrededor de las 20:00 se encuentra el pico máximo. Algo curioso es que en verano ambos picos tienen similar potencia consumida, seguramente por el uso de refrigeración, ya que en el primer pico se encuentra el momento de mayor calor en el día. Por otro lado, se observa que en general los momentos de mayor consumo del año se dan en invierno, cerca de las 20:00 horas.

Capítulo 5. Análisis de Red y Datos

5.1.2. Distribución de los Óptimos

Después de procesar los datos y obtener los valores de consigna para los generadores controladores y la potencia reactiva en los compensadores, se procede a realizar un análisis detallado de estos resultados.

Desde el punto de vista del aprendizaje, observar la variabilidad de estos permite estudiar la dificultad del problema a resolver. En particular, si se observan valores constantes en la consigna de alguno de estos generadores, o un compensador siempre apagado, el problema puede ser considerado trivial, comparado con el escenario donde los datos óptimos tengan variabilidad grande.

Por otro lado, desde una perspectiva física, es interesante analizar la distribución de los óptimos, ya que brinda información del funcionamiento de la red eléctrica y permite verificar que sea razonable. Por ejemplo, si los voltajes resultantes se encuentran en la mayoría de los casos cerca de los extremos, esto podría generar problemas de seguridad en la red eléctrica. Asimismo, en el caso de los compensadores, si un compensador opera siempre al máximo de su capacidad, esto puede indicar que esa parte de la red podría estar inestable en términos de potencia reactiva.

Comenzando con los valores óptimos de compensación de energía reactiva en los compensadores, los histogramas correspondientes se muestran en la Figura 5.11. Se observa que en 5 de los 6 casos, la solución resulta en valores máximos y mínimos. Sin embargo, para el compensador conectado a la barra Salto Grande Uruguay (SU5500_B) se predicen valores más distribuidos.

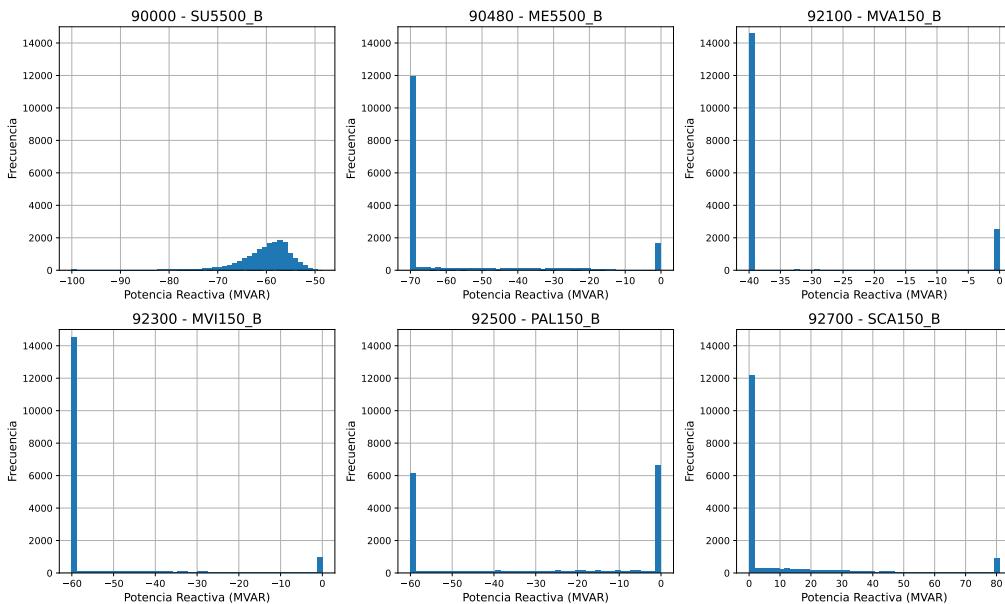


Figura 5.11: Histogramas de generación o consumo óptimo en los compensadores de reactiva. Estas corresponden, de izquierda a derecha y en orden descendente, a: Salto Grande Uruguay, Melo, Central Montevideo A, Central Montevideo I, Palmar y San Carlos.

Por otro lado, se analizan los valores óptimos de voltaje. En la Figura 5.12

5.1. Análisis de Datos

se muestran los histogramas de las consignas de voltaje para 9 generadores controladores de voltaje. Es importante recordar que las barras de 150kV tienen sus voltajes limitados entre 0,93 y 1,07 en *p.u.*, mientras que las barras de 500kV están limitadas entre 0,95 y 1,05. En los histogramas se observa que algunas barras presentan una mayor variación entre sus límites, como es el caso de Juan Pablo Terra (JPT_150B), Rocha (ROC150_B) y Pintado B (PIB150_B), mientras que otras, como Palmar 500kV (PAL500_B), Bifurcación (BIF150_B) o Montevideo L (MVL150B2), muestran menor variación. Por otro lado, los generadores Bonete y Bonete B (BOB150_B), como se mencionó en la sección anterior, tienen su voltaje restringido entre 1,04 y 1,05, causando que los valores óptimos se encuentren dentro de este rango.

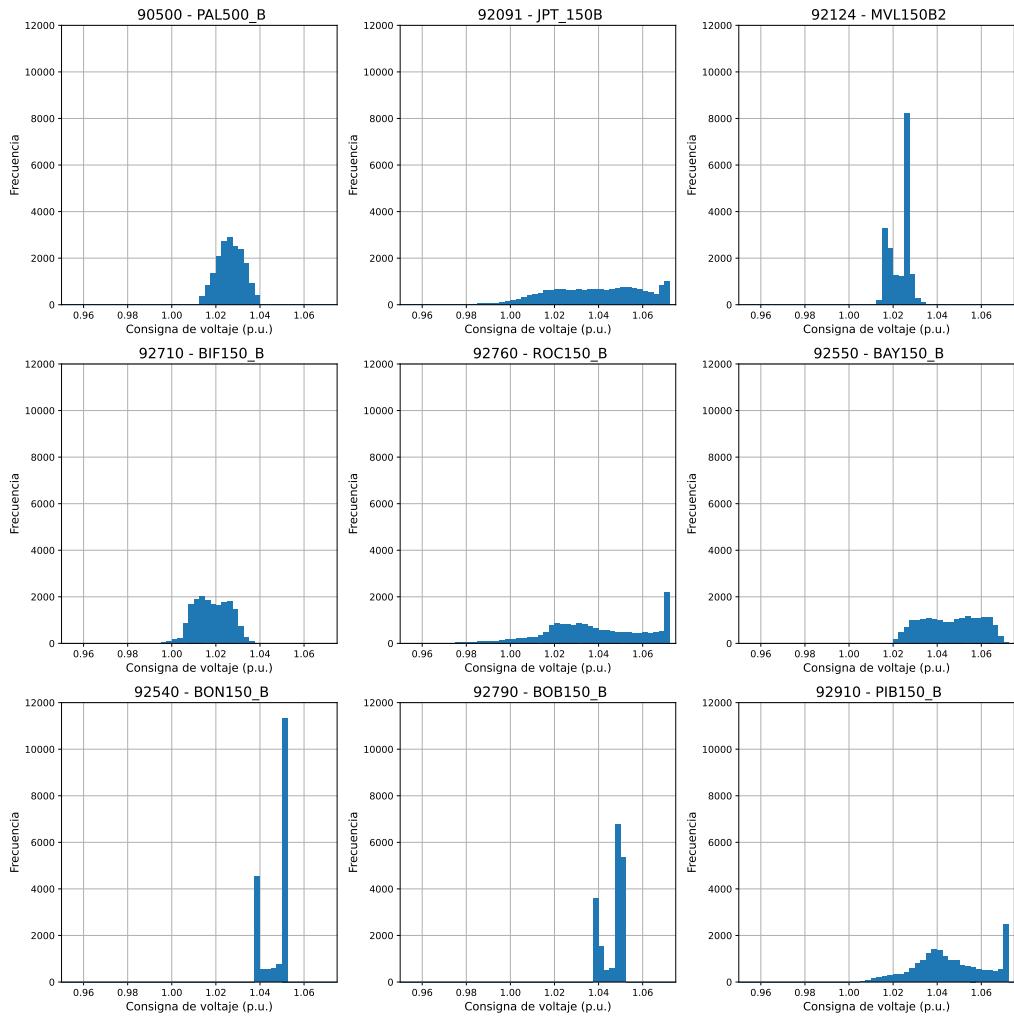


Figura 5.12: Histogramas de voltajes óptimos para 9 generadores de la red eléctrica uruguaya

A modo de comparación, también se analizan las consignas de voltaje óptimo para la red uruguaya generadas con los datos sintéticos. En la Figura 5.13 se presentan histogramas de las consignas de voltaje para los mismos 9 generadores

Capítulo 5. Análisis de Red y Datos

que se presentaron anteriormente. Se observa que estos siguen una distribución similar a la normal, lo cual en la mayoría de los casos difiere de los datos reales.

La simplicidad de las distribuciones observadas puede deberse a varias razones, siendo la principal la forma en que los datos fueron generados. Muestrear una distribución uniforme para los datos de carga y generación no reproduce todos los escenarios posibles dentro de una red eléctrica real, lo que simplifica la tarea. Al comparar con la distribución de los voltajes en el conjunto de datos de la red uruguaya con datos históricos, se observa que las distribuciones de la red sintética son más simples, lo que facilita de gran manera la aplicación de técnicas de aprendizaje. Esto se comprobará más adelante, donde se entrenan modelos de aprendizaje para el caso de la red uruguaya con datos sintéticos y para los datos históricos, y se evalúan sus desempeños.

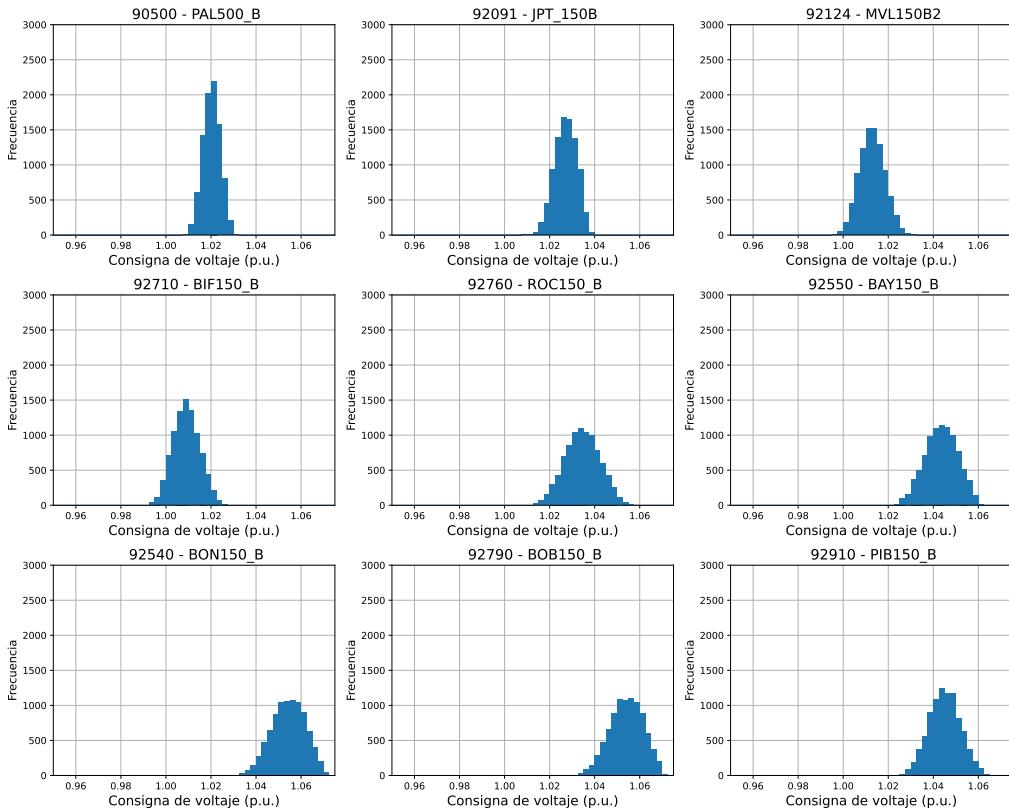


Figura 5.13: Histogramas de voltajes óptimos para 9 generadores de la red uruguaya con datos sintéticos. Estas corresponden, de izquierda a derecha y en orden descendente, a: Palmar, Juan P. Terra, Central Montevideo L, Bifuración, Rocha, Baygorria, Bonete, Bonete B y Pintado B.

5.2. Análisis del Grafo

En esta sección se aborda el análisis del grafo subyacente al modelo de la red eléctrica uruguaya. El estudio detallado del grafo permite no solo visualizar la

5.2. Análisis del Grafo

conectividad y la topología general de la red, sino también explorar propiedades estructurales que puedan influir en el desempeño del sistema y las soluciones a desarrollar. A través de la exploración de métricas de centralidad y la detección de comunidades, se busca identificar nodos y subestructuras críticas que desempeñan un papel importante en la estabilidad y eficiencia operativa de la red.

En particular, el análisis de centralidad permite detectar los nodos más influyentes en la red, aquellos que, debido a su posición o número de conexiones, podrían afectar significativamente el flujo de energía y la estabilidad del sistema en caso de fallos o perturbaciones. Por otro lado, la detección de comunidades ofrece una visión más amplia de cómo los nodos se agrupan en subconjuntos con comportamientos similares o conexiones más densas entre sí, lo que puede indicar áreas de la red que requieren un tratamiento especial en términos de gestión de reactiva.

5.2.1. Centralidad

A continuación se procede a presentar y calcular distintas métricas que miden la centralidad de un grafo. Se realiza una representación visual de cada una de ellas, en donde para la representación del grafo se utilizan las coordenadas reales de cada nodo y mediante la escala de valores se indica cuán central es cada nodo. De esta manera se logra una representación sobre un mapa real del Uruguay.

Se remarca que las centralidades pueden potencialmente resultar similares, ya que un nodo influyente en la red tiende a tomar valores altos para cualquiera de las centralidades. Sin embargo, resulta particularmente interesante analizar las diferencias encontradas en los resultados, debido a que pueden resaltar las cualidades que hacen que un nodo sea influyente.

Centralidad de Grado

La centralidad de grado es una medida básica que evalúa la importancia de un nodo según el número de conexiones que tiene. En el grafo sin pesos de la red eléctrica definido anteriormente $\mathcal{G} = (\mathcal{B}, \mathcal{L} \cup \mathcal{T})$, la centralidad de grado $C_D(i)$ para una barra $i \in \mathcal{B}$ se define como el número de líneas incidentes a i . En este caso, se utiliza la versión normalizada, fraccionando dicho valor por la cantidad de nodos del grafo:

$$C_D(i) = \frac{\text{grado}(i)}{N} .$$

Los nodos con alta centralidad de grado pueden representar subestaciones o puntos de distribución que están altamente conectados con otras partes de la red, indicando su importancia en la transmisión de energía.

En la Figura 5.14 se muestra la medida de centralidad de grado del grafo, en donde, como fue mencionado anteriormente, los nodos con más conexiones resultan ser más centrales. Se puede observar que el nodo Montevideo A, con 10 nodos vecinos, es el que mayor métrica de centralidad presenta. Lo siguen Montevideo B con 7 líneas incidentes y Bonete B junto con Montevideo E con 6. Que el nodo

Capítulo 5. Análisis de Red y Datos

Centralidad de Grado

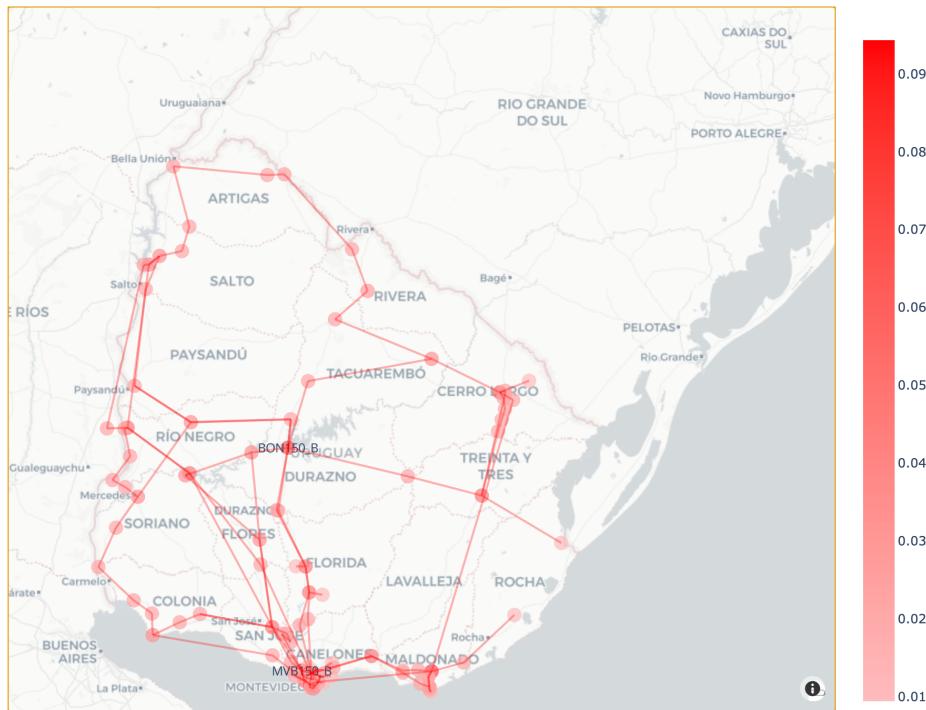


Figura 5.14: Mapa del Uruguay con indicador de centralidad de grado en cada nodo de la red. Cuanto más intenso el color del nodo, implica una mayor centralidad. Notar que la medida de centralidad se encuentra normalizada, tomando valores entre 0 y 1.

Montevideo A sea un nodo central tiene sentido, ya que es un punto de conexión entre centrales de generación en otros departamentos y la distribución en la capital del país. Otro detalle a destacar es que normalmente los nodos suelen tener entre 2 y 3 conexiones, por eso mismo es que se observa una diferencia significativa entre los valores de centralidad del resto de los nodos con los mencionados.

Centralidad de Betweenness

Esta medida de centralidad muestra qué tan importante es cada nodo como puente de los demás nodos. Es particularmente útil en redes eléctricas, donde los nodos con alta centralidad de *betweenness* podrían corresponder a elementos de la red que, si fallan, podrían causar interrupciones significativas en el flujo de energía entre diferentes regiones.

Para calcularla en el caso del grafo sin pesos, se toman todos los caminos más cortos entre cada pareja de nodos de la red. Un camino $c = (v_0, e_1, v_1, e_2, \dots, e_k, v_k)$ es una secuencia alternada de vértices y aristas, donde se debe cumplir que e_i sea la arista que conecta el vértice v_{i-1} con el vértice v_i , por lo que se define el número total de aristas k como el largo del camino. Luego, la centralidad de un nodo es igual a la cantidad de ocurrencias del nodo dentro de este conjunto de caminos.

5.2. Análisis del Grafo

Centralidad de Betweenness

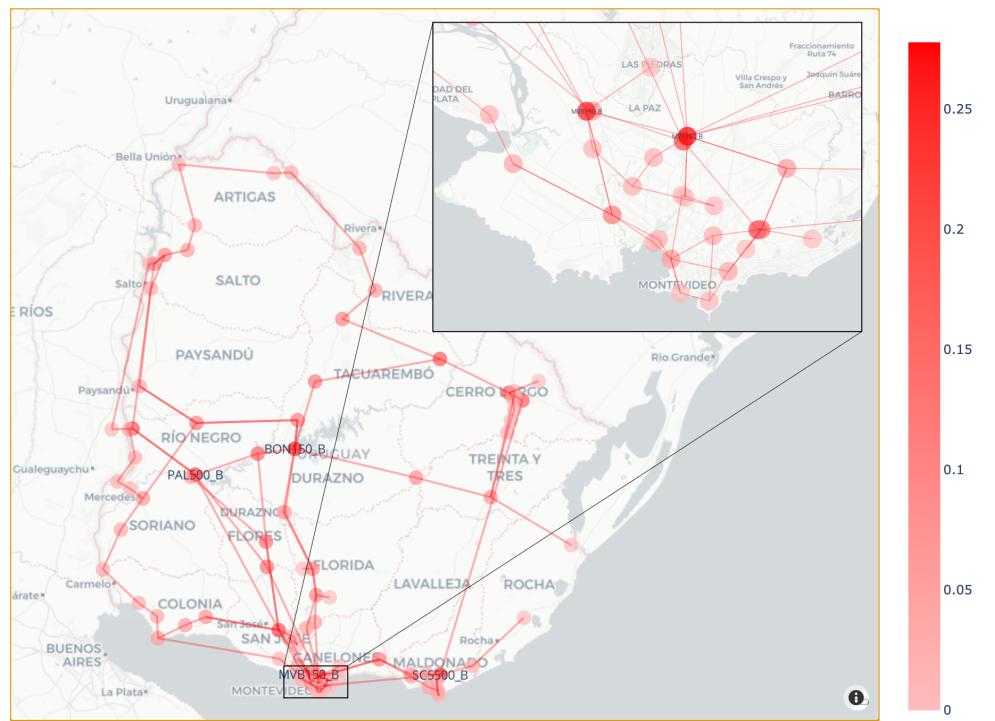


Figura 5.15: Mapa del Uruguay con indicador de centralidad *betweenness*. La escala de colores es similar al caso de centralidad de grado.

Matemáticamente, la centralidad de *betweenness* $C_B(i)$ para una barra $i \in \mathcal{B}$ se define como:

$$C_B(i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}} ,$$

donde σ_{st} es el número total de caminos más cortos entre los nodos s y t , y $\sigma_{st}(i)$ es el número de esos caminos que pasan por i .

En la Figura 5.15 se puede observar el mapa con indicadores que resaltan el valor de esta centralidad para cada nodo. En este caso, el nodo más central resulta ser Bonete B (con un valor de 0,24), mientras que el que lo sigue es Montevideo A con 0,23. Que Bonete B en este caso resulte más central que Montevideo A se debe a que este nodo se encuentra en un punto más central del país, lo que hace que sirva de puente entre muchos de los nodos. Por ejemplo, una caída en el nodo Bonete B resultaría en grandes cantidades de energía a lo largo de todo el país tomando caminos más largos e inefficientes, mientras que una caída en Montevideo A supondría sobrecargar las líneas auxiliares que permiten el acceso de la energía a la zona más densa del país.

Capítulo 5. Análisis de Red y Datos

Centralidad de Cercanía

La última métrica de centralidad a analizar es la centralidad de cercanía. Esta evalúa la importancia de una barra según su proximidad al resto de las barras de la red. Se define como la inversa de la suma de las distancias más cortas desde una barra i hasta todas las demás barras:

$$C_C(i) = \frac{1}{\sum_{t \in \mathcal{B}} d(i, t)} ,$$

donde $d(i, t)$ es la distancia del camino más corto entre los nodos i y t . En el caso del grafo sin pesos, esta distancia se mide en cantidad de saltos para llegar de un nodo a otro.

Una barra con alta centralidad de cercanía podría representar una subestación ubicada estratégicamente, desde la cual la energía puede distribuirse eficientemente a gran parte de la red.

En la Figura 5.16 se pueden observar nuevamente los valores de esta centralidad para cada nodo. En este caso se notan más nodos con valor de centralidad alto, lo que significa que muchos de ellos se encuentran bien posicionados para lograr un sistema interconectado más robusto. Los valores más altos se encuentran precisamente en Montevideo, donde se puede identificar una comunidad de nodos con varias aristas. Por otro lado, más hacia el centro del país se identifican nodos con centralidad también alta, debido a que son un nexo de distribución hacia la zona perimetral del Uruguay.

Centralidad de Vector Propio

La centralidad de vector propio o autovector es una medida más sofisticada que asigna un valor alto a los nodos que están conectados con otros nodos altamente conectados. Se basa en la idea de que no todas las conexiones son iguales: estar conectado a nodos influyentes aumenta la influencia de un nodo. La centralidad de vector propio x_i es relativa a la centralidad de los demás nodos y se define a partir de los valores propios de la matriz de adyacencia del grafo, en este caso con pesos, W :

$$x_i = \frac{1}{\lambda} \sum_{j \in \mathcal{B}} W_{ij} x_j ,$$

donde λ es el valor propio más grande. Reordenando los términos y pasando a notación vectorial, se llega a la igualdad de vector propio, siendo \mathbf{x} el vector propio asociado a λ :

$$W\mathbf{x} = \lambda\mathbf{x} .$$

Por lo tanto, la centralidad de cada nodo está determinada por cada elemento del vector propio asociado al valor propio más grande. Se elige este vector dado que es el que mejor capta las estructuras del grafo, aunque podría elegirse cualquier otro. Un bus con alta centralidad de vector propio podría estar ubicado en una región

5.2. Análisis del Grafo

Centralidad de Cercanía

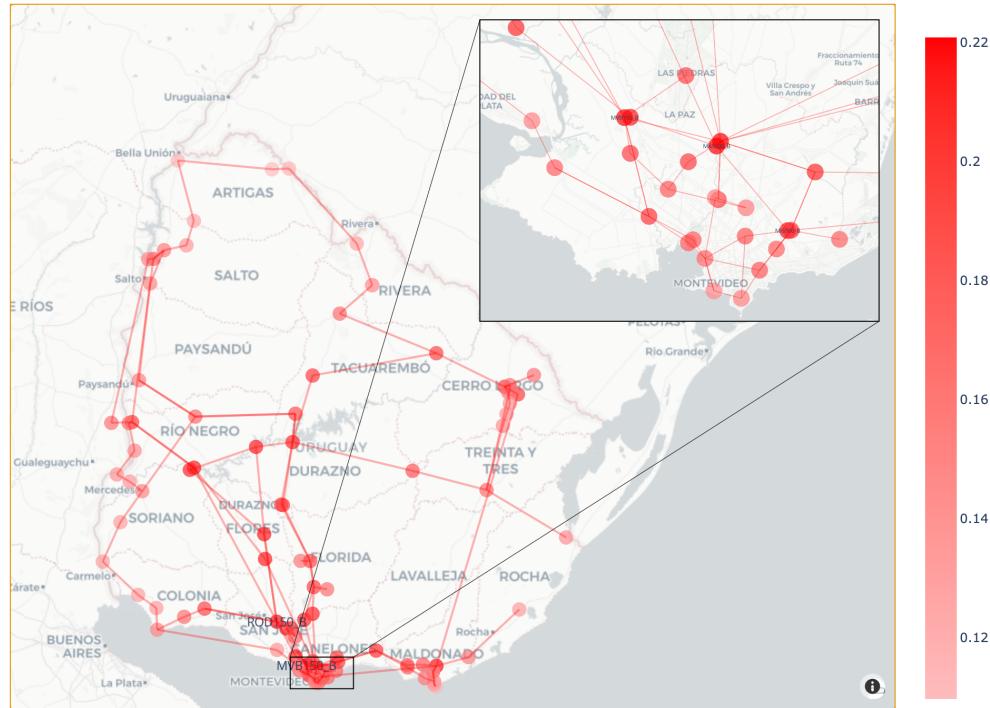


Figura 5.16: Mapa del Uruguay con indicador de centralidad de cercanía. La escala de colores es similar al caso de centralidad de grado.

estratégica, conectándose con otros buses que también son importantes para la transmisión de energía reactiva. Por lo tanto, estos nodos pueden ser críticos para la estabilidad del sistema.

En cuanto a los resultados sobre el grafo, en este caso se observa que los nodos Montevideo A y Montevideo B resultan los más centrales, seguido por varios nodos vecinos de ambos, como por ejemplo Pando. Esto quiere decir que la parte estructural del grafo ubicada en la capital del país y sus alrededores es la más importante. Visualmente, en la Figura 5.17 se pueden apreciar los resultados mediante escala de colores.

Centralidad a Partir de la Señal de Potencia

Existen varios estudios acerca del uso de señales sobre el grafo para la detección de nodos importantes. Los métodos anteriormente mencionados se enfocan en un análisis únicamente topológico de la red, asumiendo que los nodos son homogéneos, es decir, que todos se comportan de igual manera. Por lo tanto, en esta sección se utiliza un método [41] que usa como base la centralidad de valor propio para identificar nodos influyentes, considerando tanto la heterogeneidad de la red como la estructura de la misma. Sea f una señal sobre el grafo, esta centralidad se define como:

Capítulo 5. Análisis de Red y Datos

Centralidad de Vector Propio

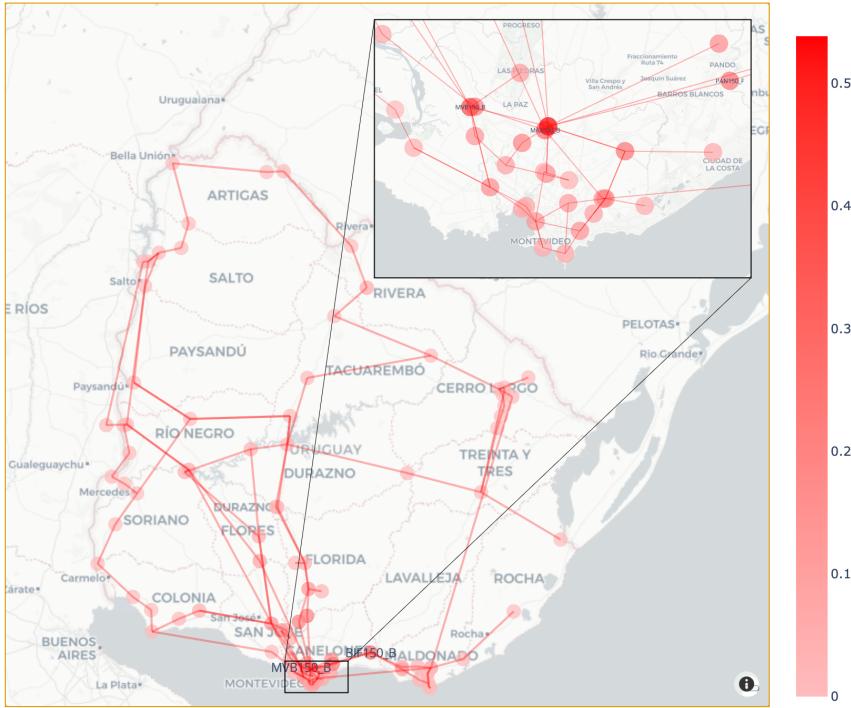


Figura 5.17: Indicador de centralidad de vector propio mediante escala de colores.

$$c_{N \times 1} = |f_{N \times 1}| \sqrt{\mathbf{v}_{N \times p}^2 \lambda_{p \times 1}^2} . \quad (5.1)$$

Los subíndices representan los tamaños de las matrices; la manera de obtener p se explica a continuación. Como se puede observar, la centralidad es muy similar a la de vector propio, aunque en este caso se incluye el valor de la señal y se utilizan distintos λ , como se explica a continuación. Definiendo la Transformada de Fourier para grafos como:

$$\tilde{f}(\lambda_l) := \langle f, \mathbf{v}_l \rangle = \sum_{i=1}^N f(i) \mathbf{v}_l^*(i) , \quad (5.2)$$

se ordenan los valores de la señal y se encuentra p tal que:

$$p = \max(\tilde{f}(\lambda_{p+1}) - \tilde{f}(\lambda_p)) . \quad (5.3)$$

Por lo tanto, para el valor de centralidad se ordenan los valores espectrales, se encuentran la máxima diferencia para encontrar el índice p y se toman los valores propios asociados a los p primeros valores espectrales.

En este caso se opta por utilizar valores de potencia activa como señal para cada nodo, más específicamente mediante la siguiente ecuación:

5.2. Análisis del Grafo

Centralidad a Partir de la Señal

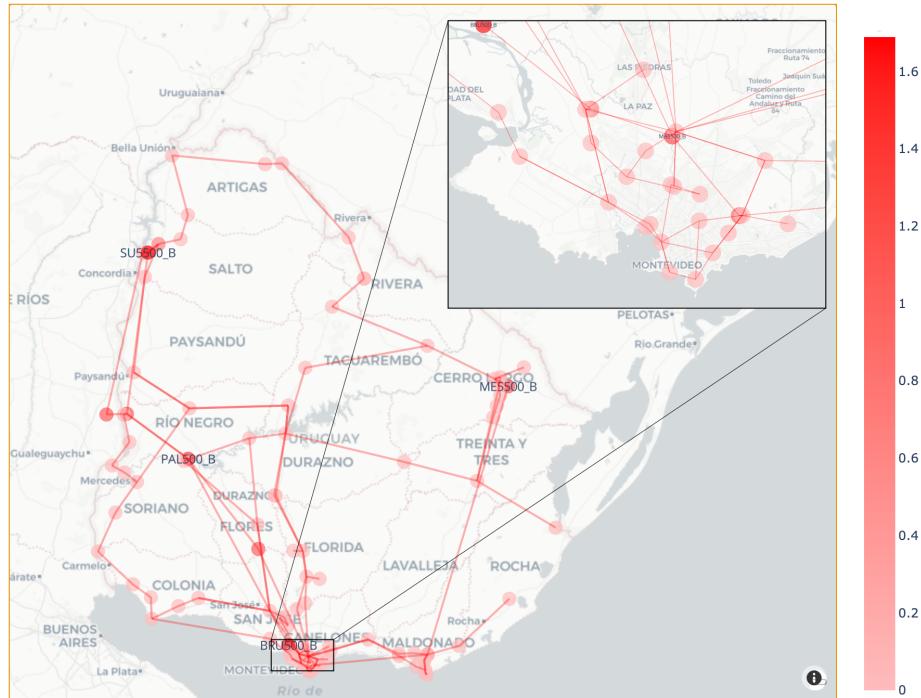


Figura 5.18: Mapa del Uruguay con indicador de centralidad utilizando la señal de potencia. La escala de colores es similar al caso de centralidad de grado.

$$s_i = |\bar{p}_i^{gen}| + |\bar{p}_i^{dem}| , \quad (5.4)$$

siendo \bar{p}_{gen_i} la potencia generada promedio en la barra i en todos los datos disponibles y \bar{p}_{load_i} la consumida promedio. Esto se debe a que los nodos que se consideran importantes en la red suelen tener un flujo de potencia importante, ya sea entrante o saliente.

Como fue presentado en el Capítulo 3, para hallar el valor de centralidad en cada nodo primero se deben calcular los vectores propios asociados a los valores propios con mayor importancia espectral. En este caso, la máxima caída en el espectro se da luego del primer valor propio, por lo que $p = 1$.

En la Figura 5.18 se aprecia la imagen con los valores de centralidad obtenidos según la escala de colores. En este caso, a diferencia de la centralidad por vector propio, el nodo más central pasa a ser Salto Grande Uruguay, seguido por Brujas y Palmar. Además, el resto de nodos con valores de centralidad elevados corresponden a aquellos que pertenecen a la red de 500kV, que son los que suelen tener valores de señal más elevados (ya que por ellos circula una potencia mayor que la mayoría de los demás nodos). Por lo tanto, utilizar como señal el valor potencia que fluye por cada nodo cambia por completo el análisis, resultando en que los nodos más influyentes de la red sean los de 500kV.

La conclusión sobre el análisis de centralidad es que existen ciertas barras que

Capítulo 5. Análisis de Red y Datos

repetidamente presentan roles importantes de centralidad para las distintas métricas utilizadas. Estas barras corresponden por un lado a las barras de Montevideo A, B y E. Tiene sentido que estas barras presenten un rol de importancia, ya que son las barras que conectan Montevideo con el resto del país. Por otro lado, se tiene a Bonete B, Salto Grande Uruguay, Palmar y Brujas, las cuales presentan mayor influencia especialmente al tener en cuenta la señal de potencia.

5.2.2. Detección de Comunidades

El análisis de centralidad tiene como objetivo identificar nodos importantes de forma aislada. Considerando la globalidad del grafo, resulta interesante identificar conjuntos de nodos que se encuentran fuertemente conectados (o más conectados que con el resto). A estos grupos de nodos que se forman en un grafo se los denomina comunidades.

La detección de comunidades en un grafo es una herramienta importante para comprender la estructura interna de una red, la cual permite identificar grupos de nodos que están más densamente conectados entre sí que con el resto del grafo. Existen diversos métodos para detectar comunidades en un grafo, en este caso se presentan dos de los más conocidos; el método de Girvan-Newman y el método de Louvain. La idea es obtener distintos resultados y así comparar ambos métodos.

Una métrica importante en el ámbito de detección de comunidades en grafos es la llamada modularidad, la cual cuantifica la calidad de una división en comunidades. Se define como la diferencia entre la fracción de aristas dentro de las comunidades en la red y la fracción esperada de dichas aristas si las aristas se distribuyeran aleatoriamente, manteniendo constante el grado de los nodos. Formalmente, la modularidad \mathcal{Q} se expresa como:

$$\mathcal{Q} = \frac{1}{2m} \sum_{ij} \left(W_{ij} - \frac{D_{ii}D_{jj}}{2m} \right) \delta(c_i c_j), \quad (5.5)$$

con m siendo el número total de aristas del grafo, c_i es la comunidad a la que pertenece el nodo i y $\delta(c_i, c_j)$ es una función que vale 1 si i y j están en la misma comunidad y 0 en caso contrario. Un \mathcal{Q} cercano a 0 indica que la partición no es mejor que la que se esperaría encontrar en un grafo aleatorio, por lo que las aristas están distribuidas sin una clara estructura de comunidad. Por otro lado, una mayor cercanía a 1 de este valor indica que existe una fuerte estructura de comunidad. Esto quiere decir que los nodos de una misma comunidad están fuertemente conectados, mientras que distintas comunidades se encuentran débilmente conectadas.

En el caso de la red eléctrica uruguaya, la detección de comunidades puede ayudar a identificar regiones que comparten comportamientos eléctricos similares o que están más interconectadas.

A continuación se presentan los resultados de las detecciones de comunidades. Nuevamente se incluyen imágenes del grafo con las coordenadas de los nodos en el mapa de Uruguay.

5.2. Análisis del Grafo

Método de Girvan-Newman

El método de Girvan-Newman [27] es un algoritmo iterativo utilizado para la detección de comunidades. Se basa en la eliminación progresiva de aristas importantes en el grafo hasta generar las particiones necesarias. La medida de importancia de aristas es la de *betweenness* presentada anteriormente pero para aristas en vez de nodos. Es decir, una arista tendrá un alto valor de *betweenness* si ejerce de puente entre varios nodos. El algoritmo funciona de la siguiente manera. Se calcula la *betweenness* de todas las aristas, se elimina la que tiene mayor valor, se recalculan nuevamente los valores sobre las aristas y se identifican las comunidades como las componentes conectadas resultantes. Un detalle es que el algoritmo tiene como criterio de parada una vez que genera k comunidades, siendo este valor una entrada al mismo. Por lo tanto, para hallar el valor de k adecuado, se realiza este método para una gran cantidad de valores de k y se elige aquel que maximice el valor de modularidad.

Pasando al análisis, las comunidades generadas por el algoritmo se visualizan mediante escala de colores en la Figura 5.19. Lo que se puede identificar son agrupaciones fuertemente marcadas según la región del país. Estas comunidades corresponden a una para el Norte del país, tres para al centro este, centro oeste y centro del país. Luego se tienen tres comunidades fuertemente marcadas para Montevideo, Maldonado y Canelones. Finalmente se tienen dos comunidades para la zona sureste, en los departamentos de Soriano, Colonia y San Jose.

Método de Louvain

El método de Louvain [13] se basa en la maximización de la modularidad, la cual consta de dos fases. En primer lugar está la fase de optimización local, donde a cada nodo se le asigna su propia comunidad. Es decir, *a priori* existen N comunidades, y el objetivo es fusionar estas comunidades iterativamente. Luego, para cada nodo se calcula cómo cambiaría la modularidad si se lo asignara a la comunidad de uno de sus vecinos. El nodo se reasigna a la comunidad que proporciona mayor ganancia de modularidad o permanece en la misma en caso de que no se observe aumento de modularidad. Esto se repite iterativamente hasta que no hayan más cambios. Una vez que no se pueden lograr más mejoras locales, comienza la segunda fase, donde las comunidades formadas se tratan como nuevos nodos (y las aristas que conectaban estas comunidades se convierten en una sola arista con peso). La primera fase se vuelve a repetir, pero ahora sobre este nuevo grafo. El algoritmo termina cuando no hay más cambios en la primera fase. El resultado final es una partición jerárquica del grafo en comunidades, donde cada nivel de la jerarquía representa una agregación más alta de nodos.

La detección de comunidades a través del método de Louvain se puede apreciar en la Figura 5.20. En este caso se detectan unas 11 comunidades. Además, las comunidades también se encuentran separadas según las distintas zonas del país, similar a lo que ocurre con el método anterior, pero con una interesante diferencia: se reconoce en amarillo una comunidad independiente para las barras de 500kV.

Capítulo 5. Análisis de Red y Datos

Detección de Comunidades mediante Girvan-Newman

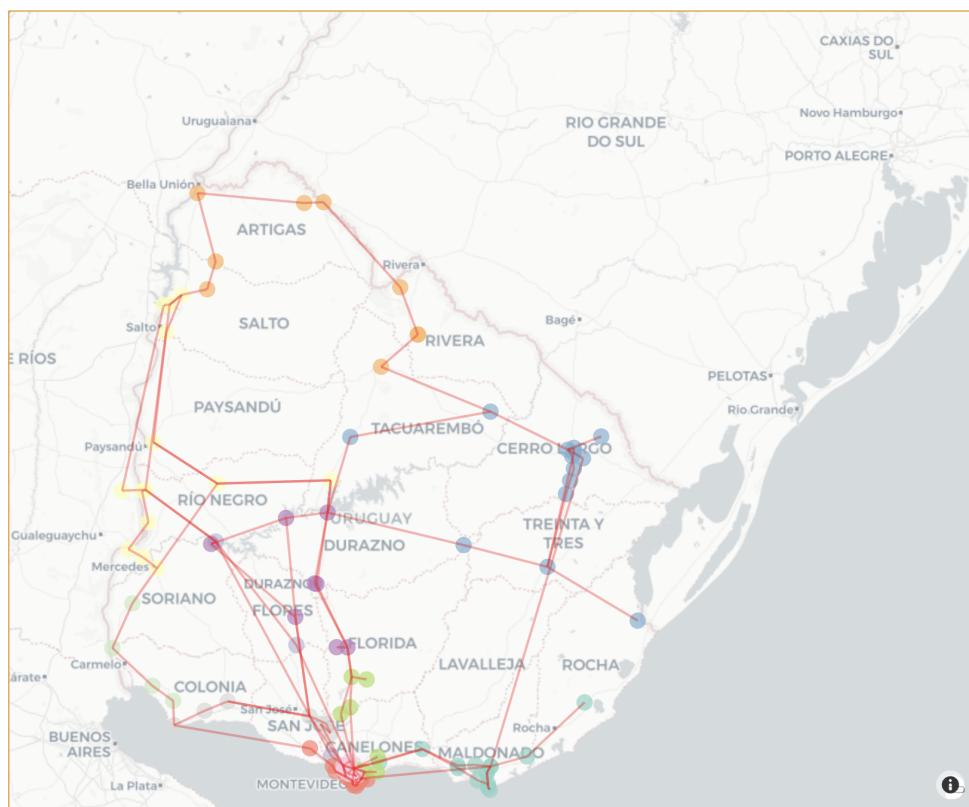


Figura 5.19: Comunidades detectadas mediante el algoritmo de Girvan-Newman.

5.2. Análisis del Grafo

Detección de Comunidades mediante Louvain

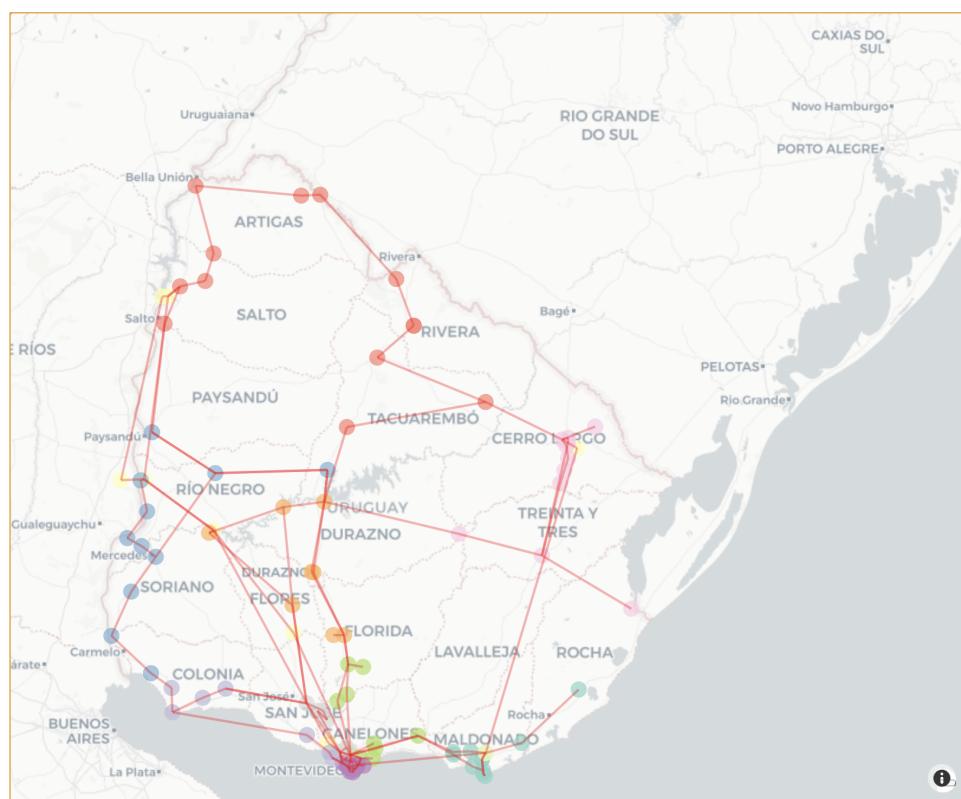


Figura 5.20: Detección de comunidades mediante el método de Louvain.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Experimentos

En esta sección se presenta todo lo relacionado con la implementación y resultados de los modelos. Para garantizar un proceso de entrenamiento estructurado y reproducible, se tiene un repositorio en GitHub¹ donde se registran las diferentes versiones de todos los cambios realizados en el código y configuraciones. Allí se encuentra todo el código necesario para el preprocesamiento de datos, definición y entrenamiento de modelos, evaluación de resultados y visualización de métricas.

Por otro lado, las bases de datos utilizadas para entrenar los modelos son almacenadas en Google Drive, a modo de tener respaldo sobre las mismas. Se dejan disponibles públicamente en [6] para futuras investigaciones. Todos los experimentos fueron realizados en una computadora con una GPU NVIDIA GeForce RTX 3060 con 12Gb de memoria, suficiente para realizar una cantidad necesaria de entrenamientos en un tiempo razonable. Para la implementación de código se utiliza PyTorch [31], una librería de Python diseñada especialmente para el desarrollo y entrenamiento de redes neuronales, cuyo funcionamiento está optimizado para tarjetas gráficas NVIDIA.

Como fue mencionado anteriormente, se proponen dos estrategias dentro del aprendizaje automático para atacar el problema: imitar el funcionamiento de un optimizador, que corresponde a aprendizaje supervisado, y definir una función de pérdidas, correspondiente a aprendizaje no supervisado. En las siguientes dos secciones se ahonda sobre los detalles de implementación y se presentan los resultados obtenidos para ambas estrategias.

6.1. Imitar Soluciones del Optimizador

En primer lugar se procede a detallar en profundidad el proceso seguido para implementar los modelos bajo la estrategia de aprendizaje supervisado. Se indican distintas técnicas evaluadas durante el entrenamiento, como el preprocesamiento de datos, la estructura de las arquitecturas utilizadas, el optimizador, las funciones de pérdida, entre otros. Luego, se presentan los resultados obtenidos, con su correspondiente análisis y extracción de conclusiones.

¹<https://github.com/tomyvazquez/DORAA-UY>

Capítulo 6. Experimentos

6.1.1. Implementación

En el Capítulo 3 se explica brevemente el proceso de entrenamiento de una red neuronal. Sin embargo, esta explicación consiste en una introducción del proceso, omitiendo varios detalles de implementación para ser más concreta. En esta subsección se explican en detalle todas las consideraciones tomadas para entrenar las redes neuronales.

Datos

Para asegurar que el desempeño observado en los resultados sea extrapolable ante nuevos datos, se divide el conjunto de datos en tres. Un subconjunto para entrenamiento, uno para validación y un último reservado únicamente para test. El primero se utiliza para entrenar los modelos, mientras que el de validación permite comparar el desempeño entre modelos entrenados. Una vez definido el mejor modelo, se utiliza el conjunto de test, de forma de tener una evaluación insesgada sobre los resultados.

El armado de estos conjuntos se hace de manera diferente para los datos sintéticos y los datos históricos. Como los datos sintéticos fueron generados aleatoriamente, no existe ningún tipo de orden temporal, por lo que la separación de estos conjuntos se hace de manera aleatoria, tomando un 77 % para entrenamiento, un 18 % para validación y un 5 % para test.

Por otro lado, los datos históricos correspondientes a la red uruguaya provienen de medidas en tiempo real, por lo que cada valor se corresponde con un instante de tiempo. En busca de que el modelo logre generalizar para datos futuros, se toman como conjunto de entrenamiento los datos entre enero de 2021 y abril de 2023 (78 % del total), desde abril a setiembre de 2023 (14 %) para validación y desde setiembre hasta diciembre de 2023 (8 %) como conjunto de test.

Luego de separados los conjuntos, se normaliza tanto la entrada como la salida. Esto es un paso importante en el preprocesamiento de datos para el entrenamiento de modelos de aprendizaje automático. El proceso implica ajustar los valores de las entradas y salidas a una escala común al normalizarlas a partir de restar la media y dividir por la varianza. La normalización de los datos de entrada permite que todas las características contribuyan equitativamente al aprendizaje del modelo. Esto evita que aquellas con valores numéricos más grandes, como las potencias, dominen sobre las que presentan valores numéricos más pequeños, como los voltajes por unidad. Por otro lado, la normalización de las salidas evita que las características con valores numéricos más grandes dominen en la función de pérdida. Un detalle muy importante de implementación es calcular el promedio y varianza utilizando solo el conjunto de entrenamiento, evitando contaminar los conjuntos de validación y test.

Luego de normalizada, se procede a dividir el conjunto de entrenamiento en lotes (*batches*). Esto es necesario por varias razones: primero, permite que el entrenamiento se realice en sistemas con memoria limitada, ya que no es necesario cargar todos los datos en la memoria al mismo tiempo; segundo, el *batcheo* facilita la paralelización, lo que puede acelerar significativamente el entrenamiento en

6.1. Imitar Soluciones del Optimizador

hardware especializado como GPU; or último, el uso de *batches* introduce cierta estocasticidad en el proceso de entrenamiento, lo que puede ayudar al modelo a escapar de mínimos locales y mejorar su capacidad de generalización.

Arquitecturas

Las dos arquitecturas a probar en este trabajo son las redes neuronales completamente conectadas y las redes neuronales para datos en grafos. Estas consisten de una secuencia de capas lineales o convolucionales, intercaladas con funciones de activación. En este trabajo, se utiliza como función de activación *Leaky ReLU*:

$$\text{LeakyReLU}(x) = \begin{cases} x, & \text{si } x \geq 0 \\ \alpha x, & \text{si } x < 0 \end{cases}, \quad (6.1)$$

donde α es una constante pequeña, en general $\alpha = 0,01$ o similar. Además de las capas lineales y la no linealidad, se agrega una capa de normalización por lotes (o *batch normalization*). Esta consiste en normalizar a la salida de las capas ocultas de la red neuronal, lo cual permite estabilizar y acelerar el proceso de entrenamiento.

Finalmente se tiene el uso de máscara a la salida del predictor. La salida óptima consiste en los voltajes en generadores controlables y reactiva en los compensadores para todas las barras, llenando con 0 aquellas barras que no tienen conectado este elemento. Por lo tanto, utilizar una máscara que multiplique la salida del predictor, llevando a cero las salidas que son siempre nulas, evita que este tenga que utilizar parte de su capacidad en aprender a llevar estas entradas a cero, potencialmente empeorando el desempeño.

Algoritmo de Entrenamiento

La función de pérdida utilizada para comparar los óptimos con las predicciones es el error cuadrático medio (MSE, por sus siglas en inglés). Sea \mathbf{y}_t la solución óptima para una entrada \mathbf{x}_t y $\Phi(\mathbf{x}; \theta)$ la predicción de la red, la función MSE se define como:

$$MSE = \frac{1}{T} \sum_{t=1}^T \|\mathbf{y}_t - \Phi(\mathbf{x}_t; \theta)\|^2. \quad (6.2)$$

Por otro lado se define una función de métrica. La tarea de esta es otorgar un valor más intuitivo que la función de pérdida respecto del desempeño del predictor. Al no ser utilizada en el entrenamiento, no es necesario calcular gradientes ni que sea diferenciable. En este caso se utiliza el error absoluto medio (MAE, por sus siglas en inglés), dividido por el largo de elementos en la salida. A su vez, se toma por separado la salida respecto a la predicción de voltajes y a los compensadores de reactiva. Por lo tanto, se tienen dos métricas:

$$MAE_v = \frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{y}_t^v - \Phi^v(\mathbf{x}_t; \theta)\|}{G}, \quad (6.3)$$

Capítulo 6. Experimentos

$$\text{MAE}_q = \frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{y}_t^q - \Phi^q(\mathbf{x}_t; \theta)\|}{R}, \quad (6.4)$$

donde $\Phi^v(\mathbf{x}_t; \Theta)$ y \mathbf{y}_t^v corresponden a la dimensión asociada a los voltajes en generadores, mientras que $\Phi^q(\mathbf{x}_t; \Theta)$ y \mathbf{y}_t^q corresponden a la asociada a la reactiva en los compensadores de reactiva. Se recuerda que G corresponde al número de generadores controladores de voltaje y R al número de compensadores de reactiva. De esta forma, se tienen como métricas el error absoluto promedio para el voltaje en los generadores y la reactiva en los compensadores. Remarcamos que en los casos de estudio de las redes IEEE solo se tiene la primera métrica, ya que no se tienen compensadores de reactiva.

Para actualizar los pesos del modelo se utiliza un algoritmo similar al descenso por gradiente estocástico (SGD), explicado brevemente en el Capítulo 3. Este es el algoritmo Adam, siglas de *Adaptive Moment Estimation*. Funciona similar al SGD, pero en lugar de utilizar una tasa de aprendizaje fija, esta se actualiza en función del primer y segundo momento del gradiente. Esto permite una convergencia más rápida y estable sin depender tanto de la elección inicial de la tasa de aprendizaje.

Regularización

Para poder generalizar ante nuevos datos, evitando el sobreajuste a los datos de entrenamiento, es necesario regularizar los modelos. Las tres técnicas usadas en este proyecto son decaimiento de pesos (*Weight decay*), descarte de neuronas (*dropout*) y frenado temprano (*early stopping*).

Weight decay es una técnica de regularización que añade una penalización al valor absoluto de los pesos del modelo durante el entrenamiento. Esto evita que los pesos crezcan demasiado, favoreciendo modelos más simples y reduciendo el sobreajuste. En esencia, impone una forma de suavidad en el modelo, limitando su capacidad expresiva.

Dropout es otra técnica de regularización que consiste en “apagar” aleatoriamente algunas neuronas durante cada iteración del entrenamiento. Esto obliga al modelo a no depender demasiado de ninguna neurona en particular, promoviendo redes más robustas y mejorando la capacidad del modelo para generalizar a nuevos datos. Al combinar estas dos técnicas, se logra un equilibrio entre la flexibilidad del modelo y su capacidad de generalización.

Finalmente *early stopping* es una técnica de regularización que consiste en detener el entrenamiento del modelo cuando el rendimiento en el conjunto de validación deja de mejorar. Durante el proceso de entrenamiento, se monitorea tanto el desempeño del modelo en el conjunto de entrenamiento como de validación; si el error en el conjunto de validación comienza a aumentar, el modelo está empezando a sobreajustarse a los datos de entrenamiento. Por lo tanto, se detiene el entrenamiento en este punto, previniendo el sobreajuste y mejorando la capacidad de generalización.

6.1. Imitar Soluciones del Optimizador

Búsqueda de Hiperparámetros

Durante el proceso de entrenamiento, existen varios parámetros del predictor y del proceso de entrenamiento para los cuales se debe elegir un valor, llamados hiperparámetros. Estos son, por ejemplo, la cantidad de capas de la red, la cantidad de neuronas en cada capa, la tasa de aprendizaje inicial, el tamaño del lote al dividir los datos, la probabilidad de apagar una neurona al utilizar dropout, entre otras. Debido a que el desempeño para diferentes conjuntos de hiperparámetros fluctúa bastante, es deseable poder entrenar múltiples modelos variando los hiperparámetros y quedarse con el que arroje mejores resultados.

La estrategia más básica para implementar esto es hacer una búsqueda por grillas. Esta consiste en definir un conjunto entero de posibilidades para cada hiperparámetro, y probar con fuerza bruta todas las combinaciones posibles. Aunque este método es robusto, en el sentido de que se prueban todas las combinaciones, es altamente ineficiente. En particular, la cantidad de modelos a entrenar crece exponencialmente con la cantidad de hiperparámetros y sus posibilidades. Una alternativa más eficiente al realizar la búsqueda de estos es mediante la optimización bayesiana.

La optimización bayesiana permite buscar el mejor conjunto de hiperparámetros de manera más eficiente que la búsqueda por grillas. En lugar de probar todas las combinaciones posibles, se construye un modelo probabilístico del espacio de hiperparámetros y selecciona las configuraciones más prometedoras basándose en resultados anteriores, reduciendo la cantidad de evaluaciones necesarias. Optuna es una biblioteca en Python que implementa esta técnica, la cual elige hiperparámetros de forma inteligente y eficiente, adaptando la búsqueda en tiempo real para optimizar los modelos más rápidamente [4].

6.1.2. Resultados

El proceso para obtener el predictor final consiste en entrenar múltiples modelos con diferentes conjuntos de hiperparámetros y luego evaluar para todos estos la métrica de MAE obtenida para el conjunto de validación. Por último, se elige como predictor final aquel que presente el mejor valor de métrica. Dado que el conjunto de validación es utilizado en el proceso de selección del predictor, su desempeño es sesgado. Para asegurarse que el desempeño es extrapolable ante nuevos casos, se evalúa el predictor para el conjunto de test. Por lo tanto, la primer mitad de los resultados se centra en observar el desempeño de los predictores respecto a su capacidad de imitar al optimizador. Para esto, se analizan las métricas de MAE obtenidas y se presentan gráficas que superponen la salida óptima con la predicha, lo que permite entender de forma visual la distancia de estas predicciones.

Aunque el entrenamiento consiste en imitar las salidas del optimizador, al final del día lo que interesa saber es la calidad de estas predicciones como solución al problema de ORPD. Es por esto que la segunda mitad de los resultados se centra en comparar el desempeño de estas como solución del ORPD y no como imitador del óptimo. Para esto, por un lado, se comparan las pérdidas de potencia del predictor contra las soluciones óptimas y por otro lado, se evalúa la factibilidad

Capítulo 6. Experimentos

Hiperparámetros IEEE30				
Hiperparámetro	Valores	FCNN	GNN	
Arquitectura				
L	2 3	2	3	
F	32 128 512	32	512	
K	2 3 4 5	-	5	
Entrenamiento				
B	32 64 128	32	32	
lr	1×10^{-6} - 1×10^{-3}	$3,5 \times 10^{-4}$	$6,5 \times 10^{-4}$	
Normalización				
<i>Norm_B</i>	True False	False	True	
<i>Norm_X</i>	True False	True	True	
<i>Norm_Y</i>	True False	True	True	
Regularización				
dropout	0,0 - 0,3	0	0,01	
weight decay	$0,0 - 1 \times 10^{-3}$	0	$3,2 \times 10^{-5}$	

Tabla 6.1: Búsqueda de hiperparámetros para la red IEEE30 para los entrenamientos de aprendizaje supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

de estas soluciones. Es decir, si cumplen o no con las restricciones impuestas por el sistema. Además de comparar contra el óptimo, se compara contra una tercera estrategia de resolución básica (o *naive*). Esta no pretende competir en desempeño con las salidas del optimizador, sino que se utiliza para brindar una referencia de comparación adicional, resaltando la magnitud de mejora que se obtiene al aplicar técnicas de optimización o aprendizaje. En el caso de las redes IEEE, la estrategia básica consiste en fijar en $1p.u$. los voltajes de los generadores. Por otro lado, para la red uruguaya, la estrategia básica consiste en fijar en $1,026p.u$ los voltajes de los generadores y apagar los compensadores. El utilizar $1,026p.u$ se debe a que los operarios buscan mantener cerca de $154kV$ ($1,026p.u$) las barras de $150kV$ en la red.

IEEE 30

Se detalla el conjunto de hiperparámetros a probar para este caso en la Tabla 6.1. Estos consisten en L (cantidad de capas), F (dimensión en capas ocultas), K (taps del filtro en GNN), B (tamaño del batch), lr (tasa de aprendizaje), $Norm_B$ (normalización por batches), $Norm_X$ (Normalización de la entrada), $Norm_y$ (Normalización de la salida), $dropout$ (probabilidad de *dropout* por neurona), $weightdecay$ (peso del *weight decay*). Además, en la misma tabla se indica la elección de hiperparámetros que resulta en el mejor desempeño obtenido para tanto FCNN como GNN.

6.1. Imitar Soluciones del Optimizador

Para los mejores modelos FCNN y GNN, cuyos parámetros se muestran en la Tabla 6.1, se evalúa el MAE_v en test, obteniendo $9,4 \times 10^{-4}$ en el primero, y $7,5 \times 10^{-4}$ en el segundo. Esta métrica se corresponde con el error absoluto promedio por generador. Por lo tanto, teniendo en cuenta que los voltajes toman valores entre $0,9p.u.$ y $1,1p.u.$, errores de tal magnitud son considerados pequeños. Esto se observa en mejor medida en la Figura 6.1 y Figura 6.2. Estas gráficas superponen los voltajes predichos sobre los óptimos en el conjunto de test, para FCNN y GNN respectivamente. Se observa cómo en ambos casos la curva de las predicciones sigue en gran medida a la curva de los óptimos, donde en la mayoría de los casos están prácticamente superpuestas, aunque se presentan ciertos casos esporádicos con errores mayores. De esta forma se concluye que los modelos logran aprender a imitar en gran medida el comportamiento de los óptimos. Cabe notar que que GNN presenta un desempeño ligeramente mejor, aunque la diferencia es casi despreciable.

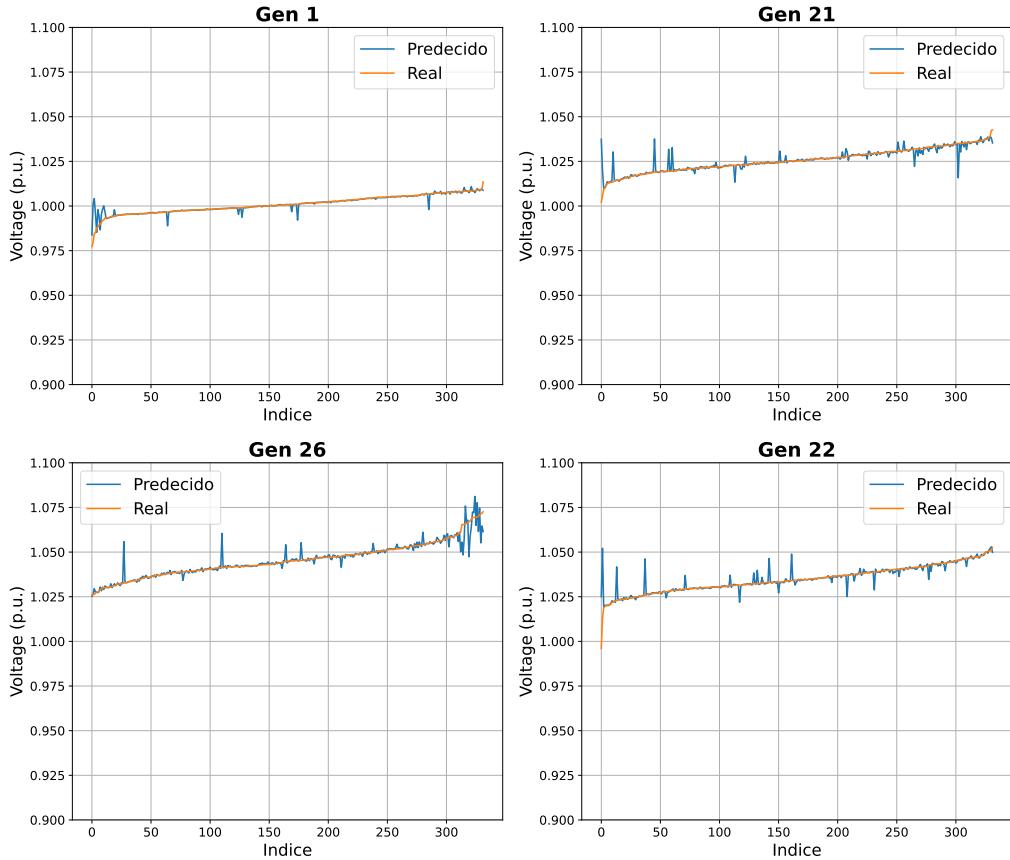


Figura 6.1: Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red IEEE30 en los datos de test, entrenada mediante aprendizaje supervisado. Se muestran 4 de los 6 generadores totales elegidos al azar.

Si bien que presenten una buena capacidad para imitar las soluciones es un gran indicio, este trabajo se centra en la resolución del problema de despacho de

Capítulo 6. Experimentos

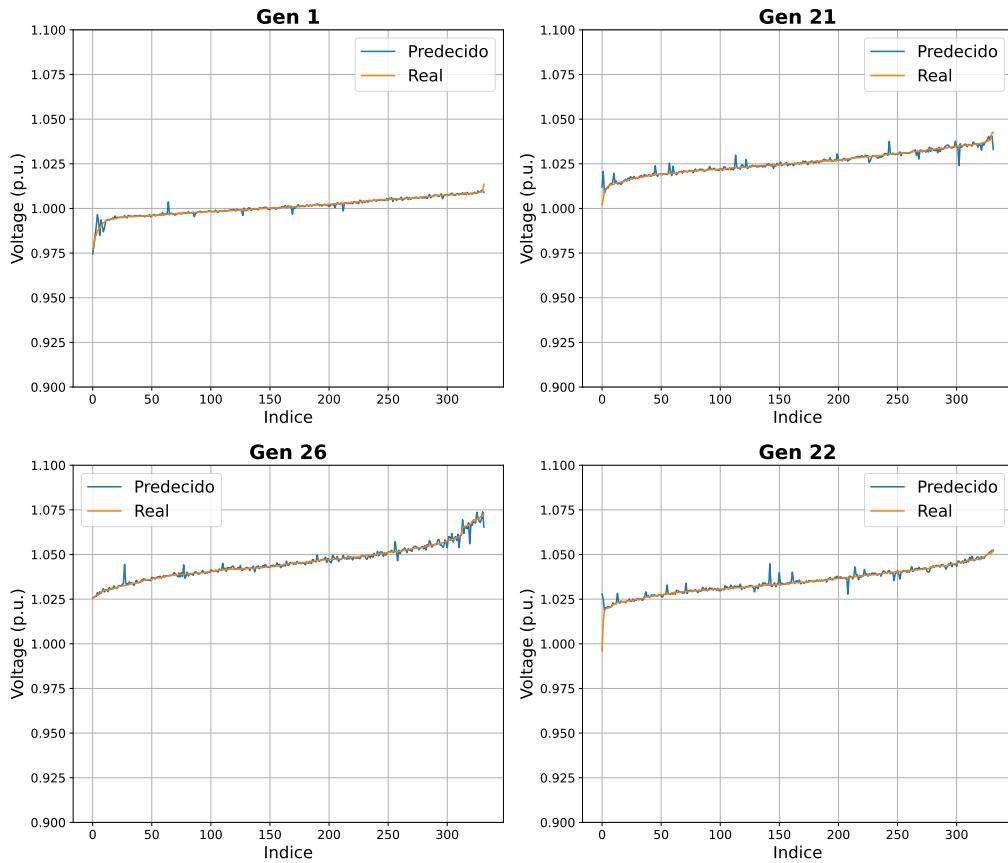


Figura 6.2: Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red IEEE30 en los datos de test en aprendizaje supervisado. Se muestran 4 de los 6 generadores totales elegidos al azar.

reactiva. Por lo tanto, es relevante analizar el desempeño de estas como soluciones al problema de optimización. A cada uno de los parámetros de entrada \mathbf{x}^t del conjunto de test se le predicen las variables de control $\Phi(\mathbf{x}^t, \theta)$ y se cargan a la red eléctrica. Luego, se corre un flujo de carga para obtener las variables implícitas y así analizar tanto las pérdidas en las líneas como la violación de las restricciones del problema. Esto se repite tanto cargando las variables de control óptimas como cargando voltajes de 1p.u. en los generadores, como estrategia básica.

Un detalle importante es que la implementación del punto interior en PandaPower presenta casos donde algunas de las restricciones son violadas muy levemente. Esto se debe a que toma algunos límites de manera laxa al resolver el ORPD, por lo que, al evaluar con las restricciones de manera estricta, algunos valores se exceden. Aunque violar una restricción implica que la solución no es factible, no es lo mismo violar una restricción por un porcentaje muy bajo que varias en gran medida. Es por esto que se propone cuantificar el incumplimiento de las restricciones. Para esto se calcula el cociente entre la violación absoluta de la restricción sobre el rango que puede tomar esta. Por ejemplo, si un voltaje puede tomar valores entre

6.1. Imitar Soluciones del Optimizador

$v_{m,min} = 0,9$ y $v_{m,max} = 1,1$, y luego de correr flujo de carga se obtiene un voltaje de $v_i = 1,102p.u.$, el porcentaje de violación para esta restricción es:

$$\frac{v_i - v_{m,max}}{v_{m,min} - v_{m,max}} \times 100 = 1\% . \quad (6.5)$$

En la Figura 6.3 se muestran histogramas que comparan los porcentajes de violación para las restricciones incumplidas en los datos de test para las distintas estrategias. Para formar cada uno de los histogramas, se evalúan todas las restricciones para todos los datos del conjunto test. Luego, para cada restricción violada, se calcula el porcentaje de violación según la Ecuación 6.5, y se agrega al histograma. Se observa que el óptimo presenta casos donde llega a violar una restricción por hasta un 4 %, mientras que FCNN y GNN presentan casos donde violan hasta un 7 % y 4 % respectivamente. Cabe destacar que para FCNN son solo dos casos aislados los que superan una violación de 4 %.

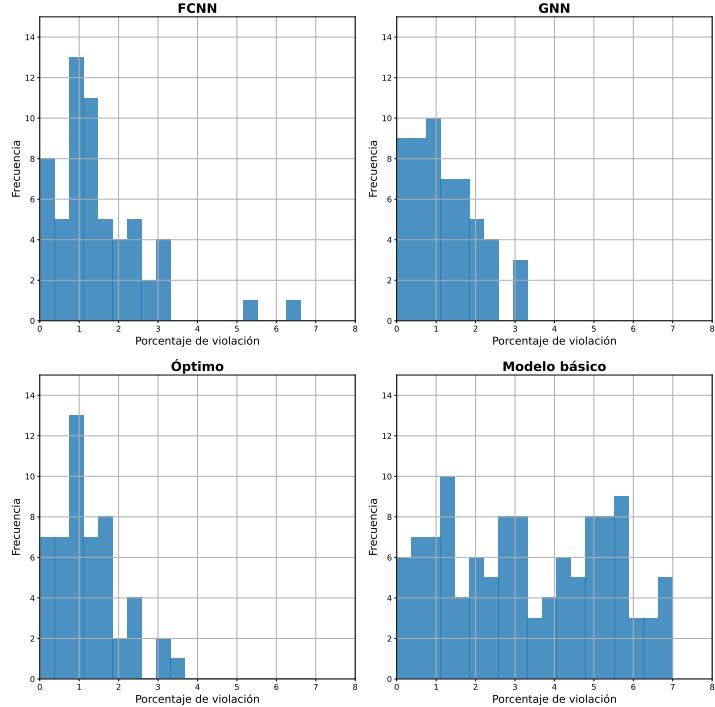


Figura 6.3: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE30 para el conjunto de datos de test, utilizando el método supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

Luego de analizadas las restricciones no cumplidas, se procede a observar las pérdidas de potencia en la red. Para esto, se grafican histogramas del cociente entre las pérdidas utilizando las estrategias de FCNN, GNN y básica, dividido por las

Capítulo 6. Experimentos

Resultados IEEE30			
	Pérdidas (MW)	Factibilidad	Factibilidad*(3,6 %)
Modelo base	2,84	60,8 %	80,1 %
Óptimo	2,63	84,6 %	100 %
FCNN	2,63	84,9 %	99,0 %
GNN	2,63	84,3 %	100 %

Tabla 6.2: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos de aprendizaje supervisado para la red IEEE30. El caso factibilidad* corresponde al porcentaje de soluciones factibles, relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a 3,6 %.

pérdidas utilizando las salidas del optimizador. Estas se observan en la Figura 6.4. Como se puede observar, para FCNN y GNN estos valores están muy cercanos a 1. Esto significa que las pérdidas son prácticamente iguales, lo que indica un desempeño similar. Por otro lado, las soluciones del básico tienden a tener entre un 5 % y 10 % más de costo.

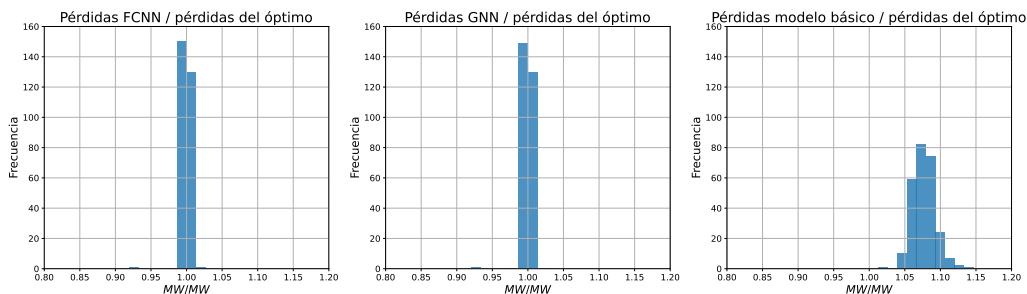


Figura 6.4: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE30 para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.

Finalmente se procede a hacer una comparación general de las distintas estrategias para resolver el ORPD. En la Tabla 6.2 se muestran las pérdidas promedio obtenidas para todos los datos del conjunto de test para los cuatro modelos. A su vez, se tiene el porcentaje de datos que resultan en un estado factible (no violan ninguna restricción). Dado que el optimizador tiene cierta flexibilidad al resolver algunas restricciones, se agrega una segunda columna de factibilidad, la cual aumenta la tolerancia de las restricciones al máximo porcentaje de violación presentado por el optimizador (3,6 %). Como se puede observar, los resultados de tanto FCNN como GNN son prácticamente idénticos al óptimo, solamente presentando un porcentaje apenas inferior de casos no factibles. Esto es altamente superior al rendimiento de la estrategia básica, donde no solo se tiene un mayor promedio de pérdidas por un 8 %, sino que se tiene una mayor cantidad de casos no factibles.

6.1. Imitar Soluciones del Optimizador

Hiperparámetros IEEE118			
Hiperparámetro	Valores	FCNN	GNN
Arquitectura			
L	2 3	3	3
F	32 128 512	128	512
K	2 3 4 5 6	-	6
Entrenamiento			
B	32 64 128	64	64
lr	1×10^{-6} - 1×10^{-3}	$9,9 \times 10^{-4}$	$8,5 \times 10^{-4}$
Normalización			
<i>Norm_B</i>	True False	False	True
<i>Norm_X</i>	True False	True	True
<i>Norm_Y</i>	True False	True	True
Regularización			
dropout	0,0 - 0,3	0	0,13
weight decay	$0,0 - 1 \times 10^{-3}$	0	$1,27 \times 10^{-5}$

Tabla 6.3: Búsqueda de hiperparámetros para la red IEEE118 para los entrenamientos de aprendizaje supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

Por lo tanto, se concluye que la capacidad de los predictores para aprender a imitar el comportamiento del IPOPT en el caso de la red IEEE30 es buena. Las redes predicen valores con errores muy pequeños respecto al óptimo, los cuales demuestran ser insignificantes, ya que el desempeño al resolver el problema es prácticamente idéntico al del optimizador.

IEEE118

Para el caso de la red IEEE118 se procede de igual manera. En la Tabla 6.3 se observan los valores de hiperparámetros a determinar, junto con su valor elegido para FCNN y GNN. La arquitectura de FCNN óptima consiste en 3 capas de 128 neuronas. Tiene sentido que para la red 118 se necesite de una red neuronal con mayor capacidad, ya que al tener más barras, los vectores de entrada y salida tendrán mayor cantidad de características. Por otro lado, en el caso de GNN, la arquitectura elegida es similar a la de la red IEEE30 pero con un valor de $K = 6$. Esto se puede atribuir a la estructura inherente de GNN, donde aumentar la cantidad de nodos no implica aumentar la complejidad de la entrada. Como las operaciones de convolución son locales, la salida para un nodo es independiente del tamaño grafo y solo son de interés los nodos vecinos de los que comparte información.

En este caso, el MAE_v en test para los mejores modelos da $1,6 \times 10^{-3}$ en el caso de FCNN, y $2,0 \times 10^{-3}$ en GNN. El desempeño obtenido es inferior que el de la

Capítulo 6. Experimentos

IEEE30. Esto es razonable, teniendo en cuenta que una red eléctrica con más barras puede presentar mayor dificultad para el modelo a la hora de aprender la relación entre sus datos. De todas formas, el desempeño sigue siendo significativamente bueno. Esto se confirma observando la gráfica que superpone los voltajes predichos con los óptimos en la Figura 6.5 para FCNN y la Figura 6.6 para GNN. Se observa como ambos predictores logran acoplarse en gran medida a los datos óptimos. En este caso es importante remarcar que GNN tiene un desempeño inferior a FCNN.

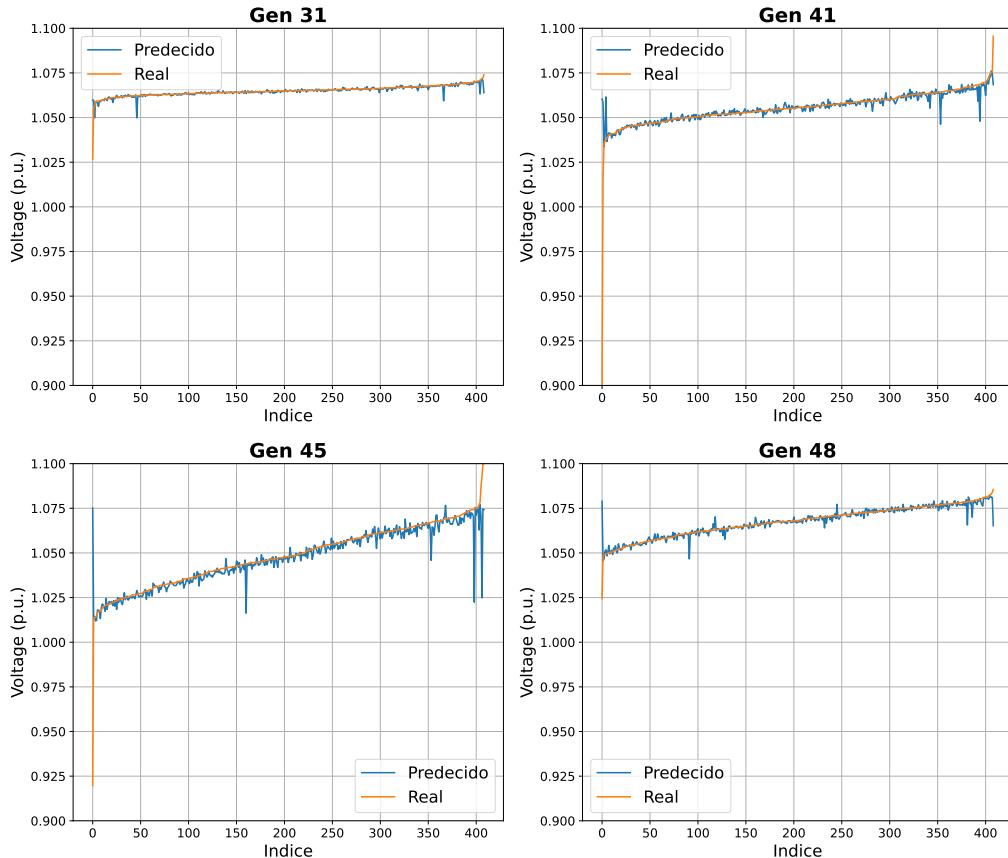


Figura 6.5: Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red IEEE118 en los datos de test. Se muestran 4 de los generadores totales elegidos al azar.

Pasando al análisis de factibilidad de las soluciones, en la Figura 6.7 se pueden observar los mismos histogramas de incumplimiento de restricciones que se realizaron para la red IEEE30. En este caso, el óptimo logra un porcentaje de violación máximo menor, siendo de tan solo un 1%. Esto se debe a que las restricciones que son más difíciles de resolver para el optimizador son las de máxima capacidad de corriente, que en esta red son holgadas. Por otro lado, FCNN presenta un porcentaje de violación máximo similar al óptimo, pero con levemente mayor cantidad de ocurrencias. Esto tiene sentido al tener en cuenta que logra imitarlo en gran medida. Por otro lado, GNN presenta una violación máxima levemente menor al 2%, pero con cantidad de ocurrencias menor. Estos resultados siguen

6.1. Imitar Soluciones del Optimizador

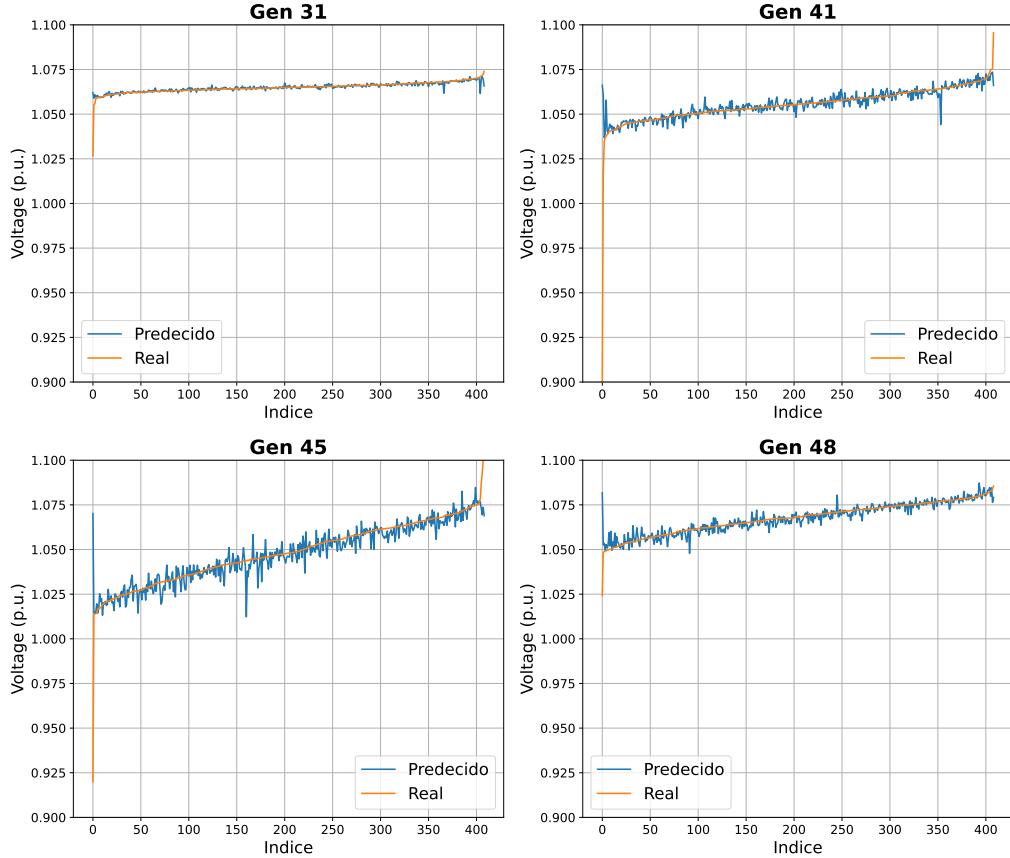


Figura 6.6: Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red IEEE118 en los datos de test. Se muestran 4 de los generadores totales elegidos al azar.

siendo un desempeño bueno, pero se evidencia cómo un peor desempeño al imitar el optimizador se traslada en mayor violación en las restricciones.

Comparando los histogramas de cociente de costos respecto del óptimo, se puede observar en la Figura 6.8 cómo ambas FCNN y GNN logran mantener costos muy similares al del óptimo. En particular, la diferencia de desempeños entre GNN y FCNN no parece afectar al costo objetivo en las soluciones. Por otro lado, el costo del modelo básico suele ser entre un 10 % y un 17 % mayor que para el optimizador.

Finalmente, se hace la comparación de las pérdidas promedio en la red eléctrica y factibilidad sobre todos los datos de test en la Tabla 6.4. En este caso se observa que los modelos entrenados suelen tener considerablemente menor factibilidad en sus soluciones, las cuales, al relajar las restricciones (en este caso un 1,0 %), se acercan en gran medida a un 100 % de factibilidad. Por otro lado, el promedio de pérdidas es levemente menor en los predictores que en el óptimo. Esto se debe a los casos donde los predictores violan una restricción, mientras que el óptimo no, resultando en pérdidas levemente menores. Sin embargo, esta diferencia es muy menor, donde FCNN tiene pérdidas un 0,3 % menores, mientras que GNN un

Capítulo 6. Experimentos

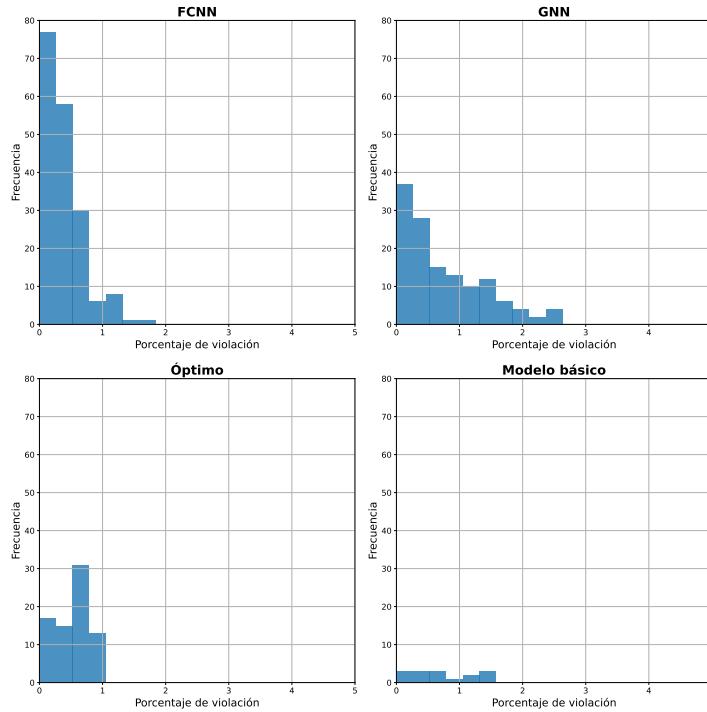


Figura 6.7: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE118 método supervisado para el conjunto de datos de test. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

0,5 % menores. Por otro lado, para el modelo base, la factibilidad es bastante alta. Esto se debe a que, en el caso de esta red eléctrica, tener voltajes cercanos a uno no implica violar restricciones, pero sí implica pérdidas mayores, presentando en promedio un 13 % más de pérdidas.

De estos resultados se puede sacar como conclusión que un peor desempeño al imitar el optimizador afecta en mucho mayor medida a los porcentajes de factibilidad que a las pérdidas en la red. Esto tiene mucho sentido, ya que las soluciones propuestas por el optimizador muchas veces se encuentran con varias restricciones cumpliéndose al límite. Por lo tanto, una variación pequeña en las variables de control se propaga en forma de una variación pequeña en las variables implícitas, provocando que aquellas que estén al límite puedan pasar a violarse. Sin embargo, esta violación es pequeña, donde con solo tener una tolerancia de un 1 % del rango de las variables se logra tener prácticamente todos los casos factibles. A su vez, esta pequeña perturbación en la solución no implica un gran impacto en las pérdidas, resultando en valores incluso menores que el óptimo.

6.1. Imitar Soluciones del Optimizador

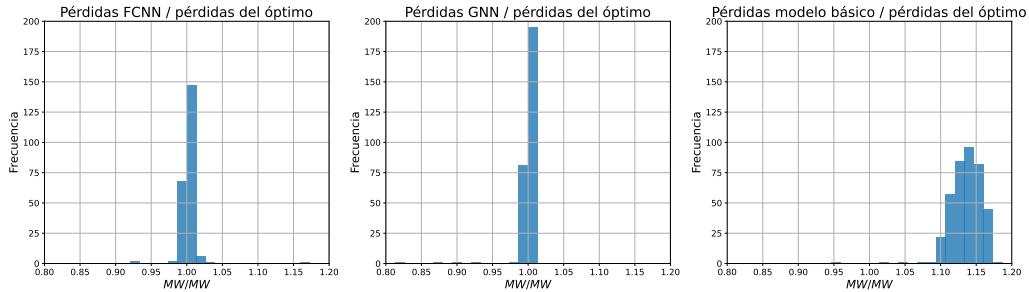


Figura 6.8: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE118 para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.

Resultados IEEE118			
Modelo	Pérdidas (MW)	Factibilidad	Factibilidad*(1,0 %)
Modelo base	95,6	96,3 %	98,7 %
Óptimo	84,5	85,1 %	100 %
FCNN	84,2	55,7 %	97,5 %
GNN	84,1	68,9 %	90,4 %

Tabla 6.4: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos para la red IEEE118, utilizando el método supervisado. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,0 %.

Red Uruguaya con Datos Sintéticos

El siguiente paso consiste en evaluar el desempeño para la red eléctrica uruguaya, pero restringiéndose a los datos sintéticos. En la Tabla 6.5 se muestra el conjunto de hiperparámetros que se prueba durante el entrenamiento. Debido a que se deben predecir tanto los voltajes en los generadores controladores como la potencia reactiva en los compensadores, se proponen dos posibles estrategias. La primera consiste en entrenar por separado dos predictores, uno para los voltajes y otro para la reactiva. La otra es entrenar un modelo que prediga ambas salidas a la vez. Las dos estrategias fueron implementadas y, debido a un desempeño similar, se eligió la segunda por simplicidad. Un detalle destacable en este caso es la necesidad de siempre normalizar la salida, ya que si no el error en las potencias reactivas de los compensadores predomina sobre los errores en los voltajes en la función de pérdida. A su vez, a la hora de realizar la búsqueda de hiperparámetros, es necesario definir una única métrica, teniendo en cuenta que se debe evaluar tanto el desempeño en la predicción de los voltajes en los generadores como la reactiva en los compensadores. Para esto, se propone utilizar como métrica el promedio entre MAE_v y MAE_q . Es decir:

Capítulo 6. Experimentos

$$\text{MAE}_{\text{prom}} = \frac{\text{MAE}_v + \text{MAE}_q}{2}. \quad (6.6)$$

Es importante remarcar dos cosas. En primer lugar, esta métrica se evalúa para la salida normalizada, ya que si no el error absoluto promedio en la reactiva predomina al del voltaje. Segundo, aunque esta métrica permite elegir el conjunto de hiperparámetros óptimo, no es muy intuitiva para analizar, ya que se hace sobre las salidas normalizadas.

En este caso, el mejor resultado se obtiene con $L = 3$ y $F = 1024$, similar al caso de la IEEE118. Esto tiene sentido, ya que al igual que la red 118, el número de barras de la red es grande y es necesario tener expresividad. En el caso de GNN, el mejor modelo se obtiene para $L = 3$, $F = 64$ y $K = 8$. Para poder comparar el desempeño con los casos de prueba anteriores, se desnormaliza la salida y se evalúan por separado las dos métricas de MAE_v y MAE_q . En el caso de los voltajes, el resultado es de $4,6 \times 10^{-4}$ y $5,5 \times 10^{-4}$ para FCNN y GNN respectivamente, mientras que para la reactiva en los compensadores es $2,8 \times 10^{-1}$ y $2,9 \times 10^{-1}$. Respecto a los voltajes, cuyo desempeño se puede comparar con los casos anteriores, se observa una mejoría en este tanto para la IEEE30 como para la IEEE118. Esto se observa en mejor medida en la Figura 6.9 y la Figura 6.10, donde el valor predicho es prácticamente idéntico al óptimo. Para el caso de los compensadores de reactiva, el desempeño también es muy bueno. Aunque el valor de la métrica para los compensadores es mucho mayor que para los voltajes, recordamos que esta métrica corresponde al error absoluto promedio. En este caso, al predecir la potencia reactiva, los compensadores presentan rangos del orden de decenas de $MVar$, por lo que errores de $0,29MVar$ son pequeños. Esto se confirma observando las predicciones en la Figura 6.11 y la Figura 6.12. Similar al caso de la IEEE118, ocurre que el desempeño de FCNN es superior al de GNN. Sin embargo, en este caso la calidad para imitar el optimizador en ambos modelos es sumamente preciso.

Esta afirmación se respalda al observar el desempeño de estas soluciones al resolver el problema de despacho de reactiva. Empezando con el análisis de factibilidad de las soluciones en la Figura 6.13, se observa en primer lugar que el óptimo no viola en ningún caso ninguna restricción. Esto seguramente se deba a que los valores nominales utilizados para generar la data sintética se toman a partir del promedio de los datos históricos, y la red uruguaya está muy bien dimensionada para casos cercanos al nominal, tanto en capacidad de corriente en las líneas como de compensación de reactiva. Es por esto que se supone que el optimizador no presenta dificultad en obtener soluciones sin necesidad de flexibilizar los límites. Observando los resultados para FCNN y GNN, se observa cómo presentan violaciones de hasta un 3% y 5% respectivamente. Aunque estas violaciones puedan parecer altas, sobre todo teniendo en cuenta que el optimizador no viola ninguna restricción, las instancias en las que ocurren estas violaciones son esporádicas. Adelantando los resultados obtenidos en la Tabla 6.6, se observa como los porcentajes de datos factibles son de 98,1% y 98,6% respectivamente. Se remarca que en este caso no hay columna con el porcentaje de factibilidad teniendo en cuenta la tolerancia del óptimo, ya que en este caso el óptimo no presenta violación alguna.

6.1. Imitar Soluciones del Optimizador

Hiperparámetros URU Sintética				
Hiperparámetro	Valores	FCNN	GNN	
Arquitectura				
L	2 3 4	3	4	
F	32 64 128 512 1024	1024	64	
K	2 3 4 5 6 7 8	-	8	
Entrenamiento				
B	32 64 128	64	32	
lr	1×10^{-6} - 1×10^{-3}	$5,8 \times 10^{-5}$	$4,7 \times 10^{-4}$	
Normalización				
Norm_B	True False	False	True	
Norm_X	True False	True	True	
Norm_Y	True False	True	True	
Regularización				
dropout	0.0 - 0.3	0.19	0.02	
weight decay	0.0 - 1×10^{-3}	$1,2 \times 10^{-5}$	$1,7 \times 10^{-5}$	

Tabla 6.5: Búsqueda de hiperparámetros para la red uruguaya con datos sintéticos para los modelos de aprendizaje supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

	Pérdidas (MW)	Factibilidad
Modelo base	19,0	91,1 %
Óptimo	17,53	100 %
FCNN	17,53	98,1 %
GNN	17,53	98,6 %

Tabla 6.6: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos de aprendizaje supervisado para la red uruguaya con datos sintéticos.

Respecto a las pérdidas, se observa en la Figura 6.14 cómo nuevamente los costos resultantes son prácticamente iguales entre el óptimo con FCNN y GNN, mientras que la estrategia base resulta en costos entre 5% y 10% mayores. En particular, observando las pérdidas promedio, se tiene que son iguales para FCNN, GNN y óptimo, mientras que para el modelo base son un 8% mayor.

Estos resultados permiten extraer dos conclusiones principales. En primer lugar, el modelo de la red uruguaya y los datos obtenidos sobre ella se validan parcialmente, no en términos de su precisión respecto a la realidad, sino en su utilidad como caso de estudio. Al utilizar datos muestreados alrededor de los valores promedio históricos, se logra un 100% de factibilidad. En segundo lugar, se concluye que la dificultad que enfrentan los predictores para replicar las soluciones del optimizador no está determinada solo por el tamaño de la red, sino también por la complejidad del problema de optimización. En este caso, se logra replicar

Capítulo 6. Experimentos

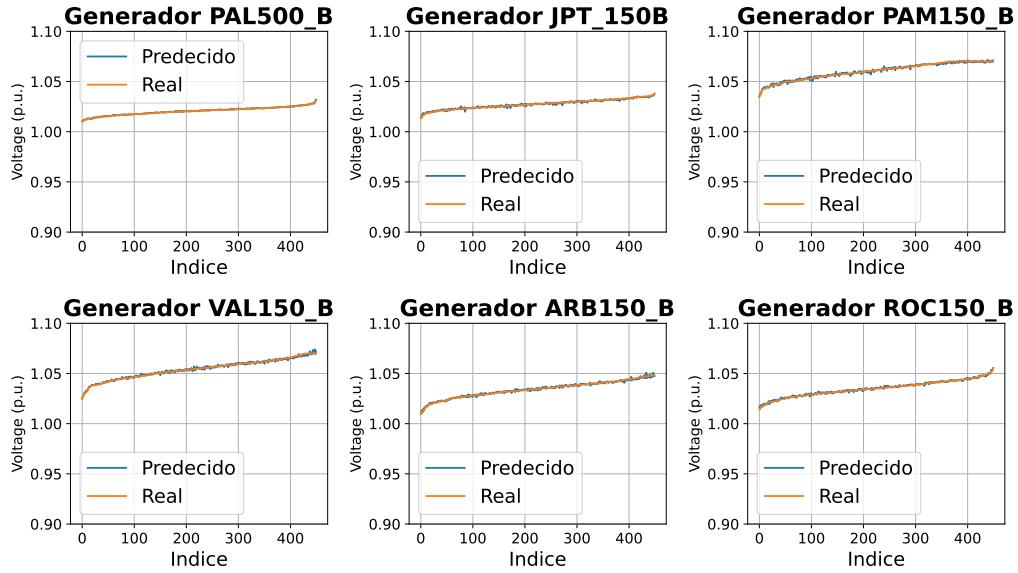


Figura 6.9: Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red uruguaya con datos sintéticos en los datos de test, mediante aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.

los voltajes del optimizador con mayor precisión que en la red IEEE30, a pesar de tener un mayor número de barras. Sin embargo, la facilidad de esta red para resolver el despacho de carga se refleja en el 100 % de factibilidad del optimizador. Esto se puede contrastar con el desempeño en la red IEEE30, donde en el 15,4 % de los casos, el optimizador tuvo que flexibilizar los límites, con violaciones relativas de hasta un 4 %, para encontrar una solución.

Red Uruguaya con Datos Históricos

El paso final para esta estrategia consiste en evaluar el desempeño al utilizar los datos históricos. En la Tabla 6.7 se muestra el conjunto de hiperparámetros probado durante el entrenamiento. Idéntico al caso sintético, se utiliza una única red para predecir tanto los voltajes como la potencia reactiva y se utiliza como métrica para comparar hiperparámetros el MAE_{prom} definido anteriormente sobre la salida normalizada.

En el caso de los datos sintéticos, el MAE_v para FCNN y GNN son $2,6 \times 10^{-3}$ y $3,7 \times 10^{-3}$, mientras que el MAE_v es de 2,75 y 3,76. Este desempeño es notablemente inferior que para el caso de los datos sintéticos, con valores de métrica aproximadamente 10 veces mayores. Nuevamente, el desempeño de GNN resulta peor que el de FCNN. Observando las predicciones para algunos de los voltajes en la Figura 6.15 y la Figura 6.16 y para algunos compensadores de reactiva en la Figura 6.17 y la Figura 6.18 se observa como en este caso las predicciones son significativamente más ruidosas, aunque logran seguir el promedio de los óptimos.

Continuando por el análisis de factibilidad de las soluciones, el cual se puede

6.1. Imitar Soluciones del Optimizador

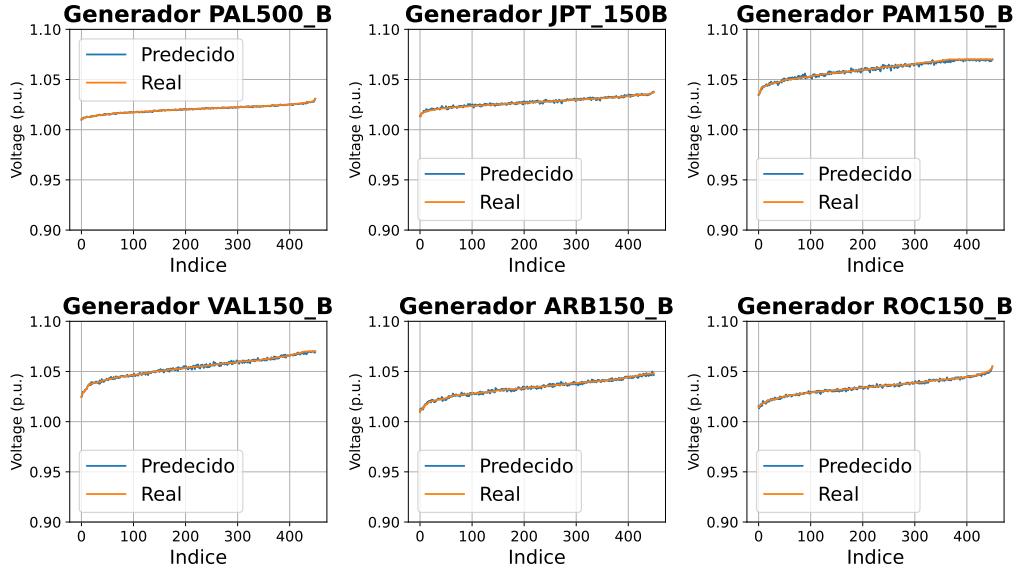


Figura 6.10: Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red uruguaya con datos sintéticos en los datos de test entrenada con aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.

observar en la Figura 6.19, se tiene que el óptimo viola las restricciones por un máximo de 2 %. En cuanto a FCNN, se tiene una violación máxima de alrededor de 5 %, lo cual es levemente mayor. Por otro lado, GNN sobrepasa los límites hasta un margen de 15 %. Esto desempeño es evidentemente inferior, y se hereda de la menor capacidad de este predictor para imitar al optimizador. Respecto a las pérdidas, se observa en la Figura 6.20 que estas no se ven casi afectadas por el desempeño inferior de los predictores, manteniendo valores muy cercanos a uno para los casos de GNN y FCNN, mientras que se tienen valores entre un 3 % y 10 % mayores para la estrategia básica.

Observando el resultado resumido en la Tabla 6.8, se tiene que el porcentaje de factibilidad es el principal afectado, con valores que bajan de 79,8 % en el óptimo a 64,8 % y 61,4 % en FCNN y GNN respectivamente. En el caso de FCNN, al tomar la misma tolerancia que el óptimo, el porcentaje de factibles sube hasta un 92,6 %, mientras que GNN se eleva solo hasta un 74,6 %. Por otro lado, las pérdidas son idénticas entre los predictores y el óptimo, mientras que el modelo base presenta valores 8 % mayores aproximadamente. Estos resultados confirman nuevamente el hecho de que al aprender a resolver problemas de optimización, la principal dificultad consiste en no pasar a violar las restricciones, ya que errores pequeños en la predicción pueden fácilmente propagarse como incumplimientos de restricciones que estaban cerca, o sobre el límite.

De los resultados obtenidos, se concluye que utilizar datos históricos en lugar de datos sintéticos impacta negativamente en el desempeño de los predictores. Esto puede deberse a diversas razones, de las cuales se destacan dos principales. La primera es la mayor complejidad en la distribución de los datos de entrada reales,

Capítulo 6. Experimentos

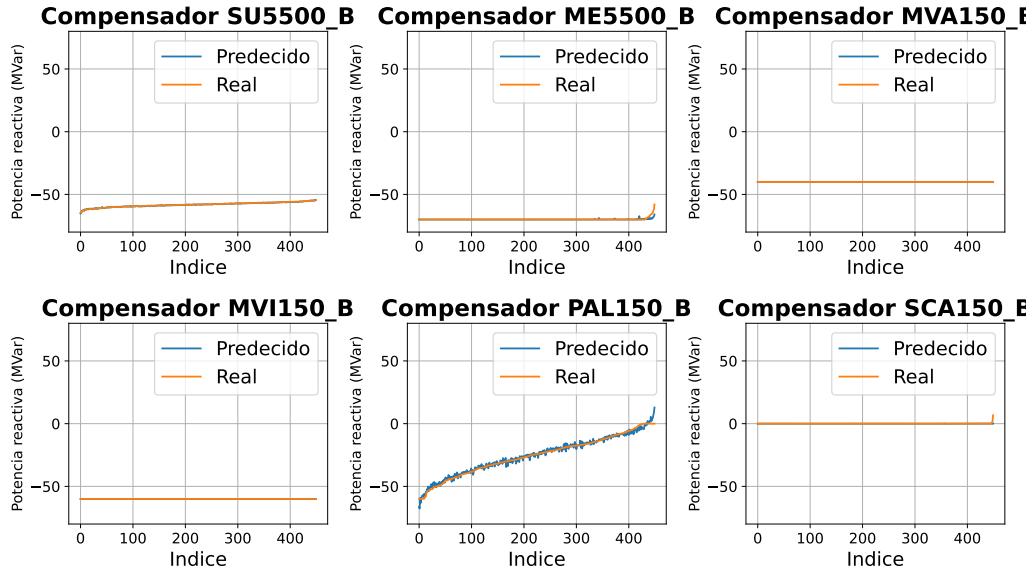


Figura 6.11: Comparación entre las potencias de los compensadores óptimas (naranja) y predichas por FCNN (azul) para la red uruguaya con datos sintéticos en los datos de test, entrenadas mediante aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.

lo cual se analizó en detalle en el Capítulo 5. Esta complejidad incrementada genera una mayor varianza en las distribuciones, lo que requiere una cantidad más amplia de datos para capturar adecuadamente la diversidad de posibles escenarios. Esta es una regla común en aprendizaje automático: cuanto más complejo es el escenario, no solo se necesita mayor complejidad en la red neuronal, sino también un volumen mayor de datos.

La segunda razón es que, además de la mayor complejidad en las distribuciones de los datos reales, es probable que existan diferencias significativas entre las distribuciones de los conjuntos de entrenamiento, validación y prueba. Esto se debe a la dependencia temporal inherente a los datos reales, algo que no ocurre con los datos sintéticos, donde los tres conjuntos se muestran de la misma distribución. Para confirmar esto, se comparan los promedios de la potencia activa demandada y generada en los tres conjuntos, tanto para los datos sintéticos como para los datos reales. En el caso de los datos sintéticos, el promedio de potencia consumida es de 1263 MW para el conjunto de entrenamiento, 1264 MW para el conjunto de validación y 1264 MW para el conjunto de prueba. En cambio, para los datos reales, los promedios son 1260 MW, 1009 MW y 1130 MW, respectivamente. Estas diferencias son esperables por razones como al aumento promedio anual en el consumo, la variación estacional, diferentes condiciones climáticas, entre otras.

Existen diversas técnicas en el ámbito del aprendizaje automático que estudian la capacidad de transferir un predictor entrenado en una distribución de datos a otra distribución diferente pero relacionada. Este campo es conocido en la literatura como adaptación de dominio [16] y puede ser útil, por ejemplo, para abordar el problema de la variación en el consumo de potencia media de un año a otro.

6.1. Imitar Soluciones del Optimizador

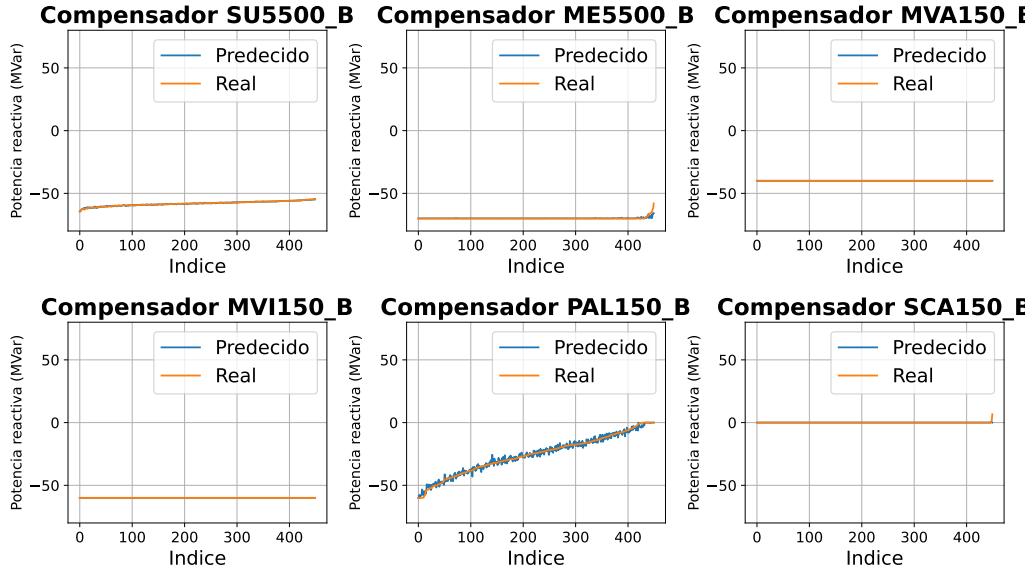


Figura 6.12: Comparación entre la potencia de los compensadores óptimas (naranja) y predichas por GNN (azul) para la red uruguaya con datos sintéticos en los datos de test, para entrenamientos de aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.

Hiperparámetros URU Histórico			
Hiperparámetro	Valores	FCNN	GNN
Arquitectura			
L	2 3 4	4	4
F	32 64 128 512	512	64
K	2 3 4 5 6 7 8	-	8
Entrenamiento			
B	32 64 128	64	32
lr	1×10^{-6} - 1×10^{-3}	$1,02 \times 10^{-5}$	$2,4 \times 10^{-4}$
Normalización			
Norm_B	True False	False	True
Norm_X	True False	True	True
Regularización			
dropout	0.0 - 0.3	0.16	0.056
weight decay	$0.0 - 1 \times 10^{-3}$	$3,5 \times 10^{-6}$	$3,9 \times 10^{-6}$

Tabla 6.7: Búsqueda de hiperparámetros para la red uruguaya estrategia supervisada con datos históricos. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

Capítulo 6. Experimentos

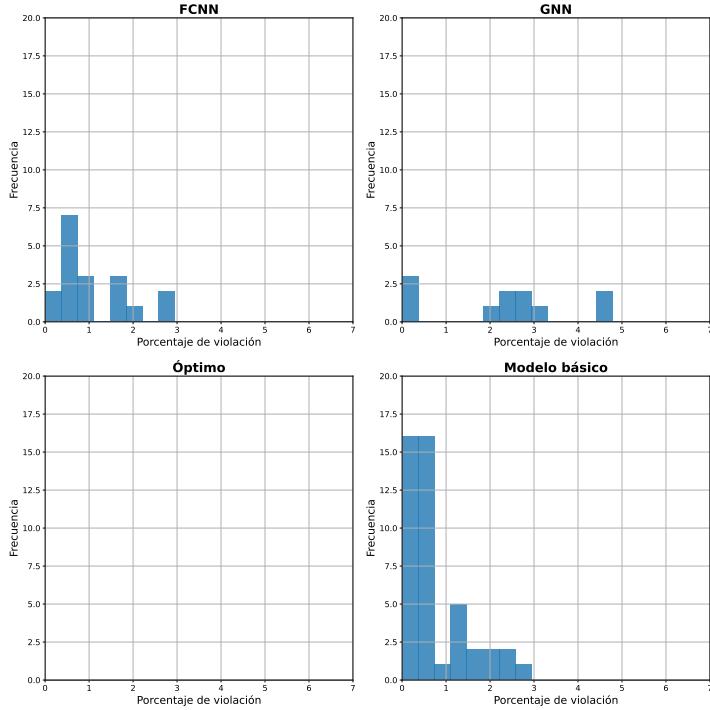


Figura 6.13: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red uruguaya con datos sintéticos para el conjunto de datos de test, utilizando el método supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

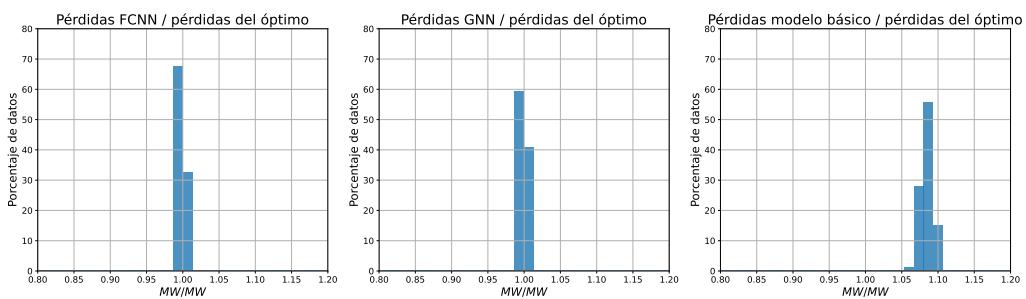


Figura 6.14: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos sintéticos para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.

6.1. Imitar Soluciones del Optimizador

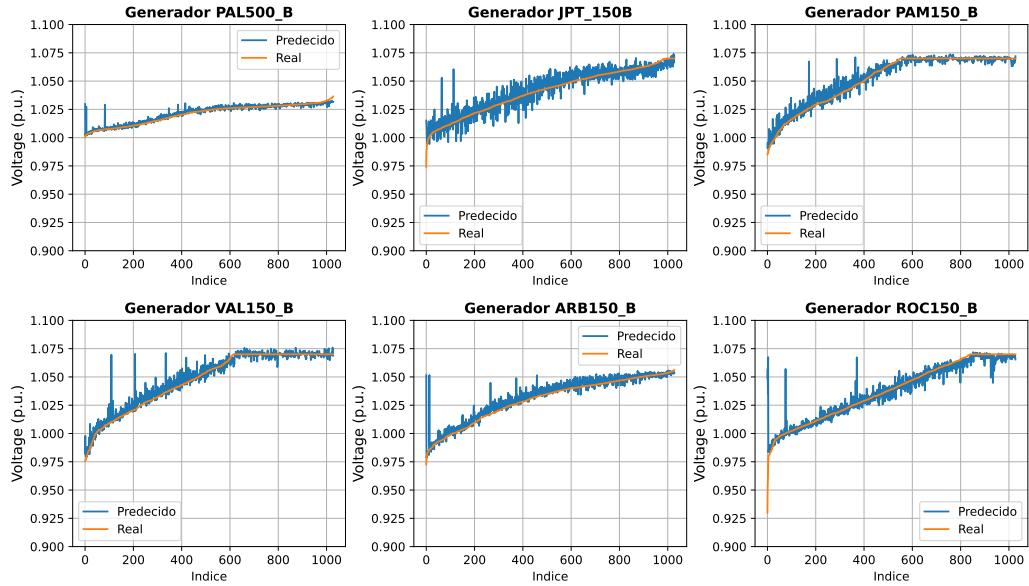


Figura 6.15: Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.

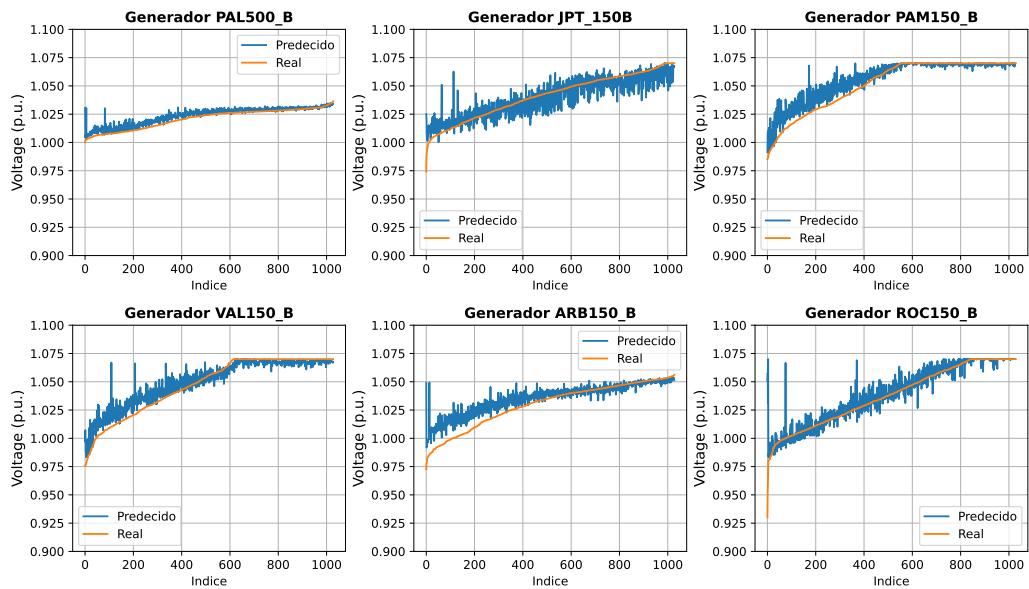


Figura 6.16: Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.

Capítulo 6. Experimentos

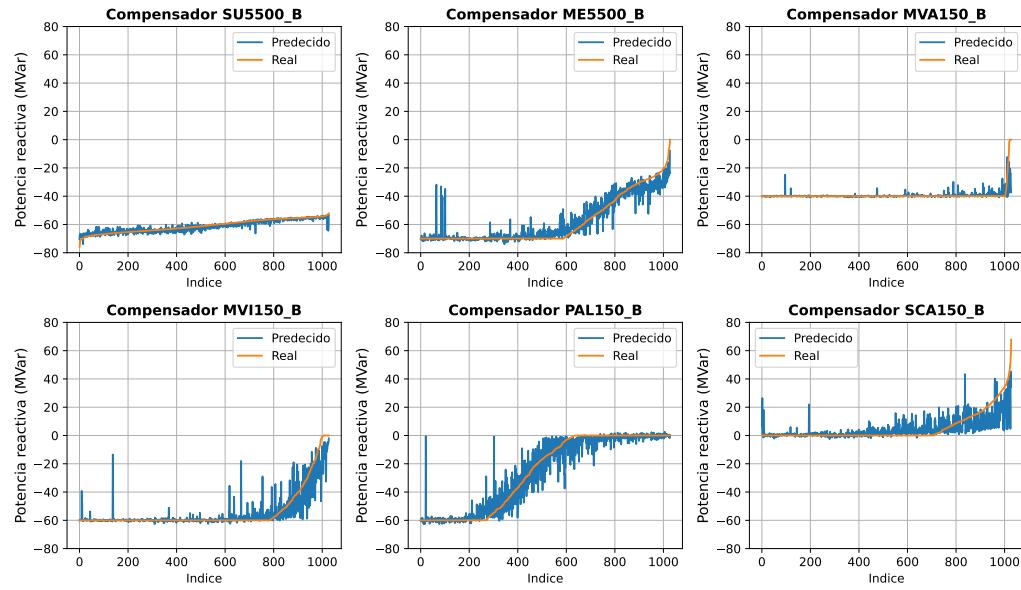


Figura 6.17: Comparación entre las potencias en los compensadores óptima (naranja) y predicha por FCNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.

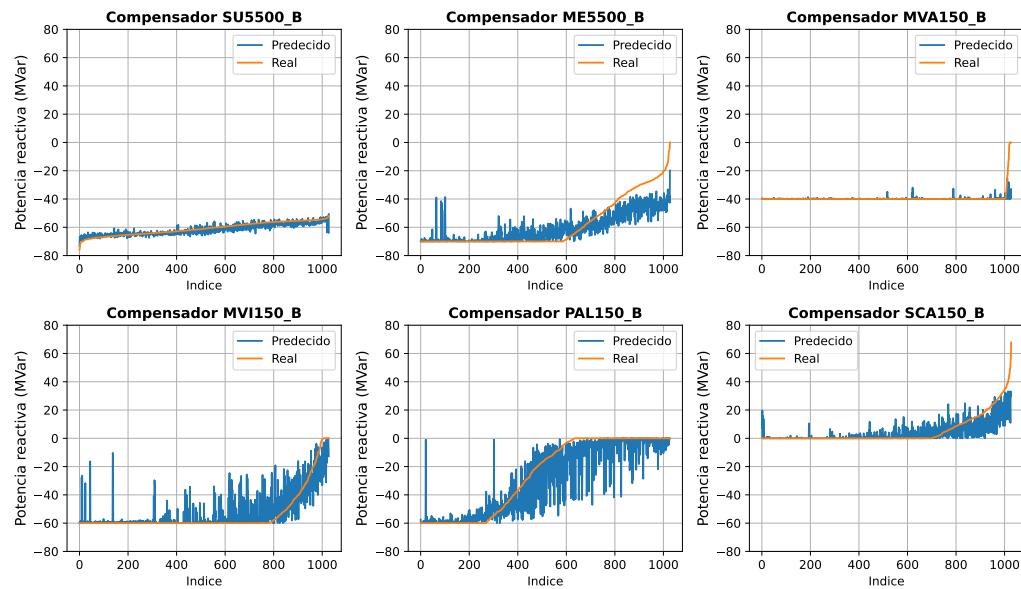


Figura 6.18: Comparación entre las potencia en los compensadores óptima (naranja) y predicha por GNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.

6.1. Imitar Soluciones del Optimizador

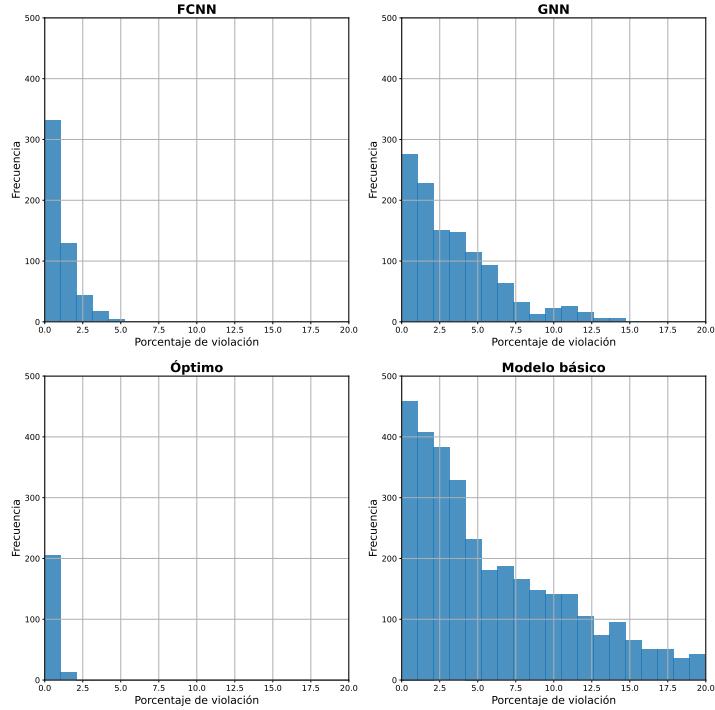


Figura 6.19: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red eléctrica uruguaya con datos históricos para el conjunto de datos de test, utilizando el método supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

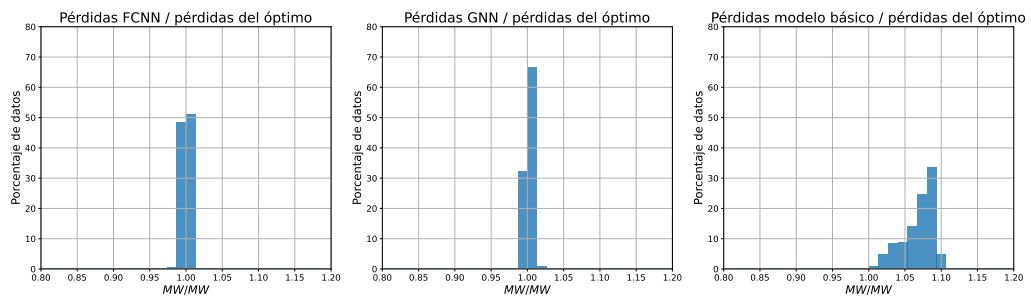


Figura 6.20: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos históricos para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.

Capítulo 6. Experimentos

	Pérdidas (MW)	Factibilidad	Factibilidad*(1,8 %)
Modelo base	30,7	37,1 %	51,9 %
Óptimo	28,3	79,8 %	100 %
FCNN	28,3	64,8 %	92,6 %
GNN	28,3	61,4 %	74,6 %

Tabla 6.8: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos para la red uruguaya con datos históricos, utilizando el método supervisado. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,8 %.

6.2. Minimización de Pérdidas

6.1.3. Tiempos de inferencia

La principal razón por la cual se propone implementar esta estrategia es lograr un desempeño similar al optimizador pero con tiempos de inferencia significativamente menores. Es por esto que se estudian los tiempos de inferencia para los diferentes métodos en los distintos casos de prueba. Para esto, se propone observar el tiempo promedio de inferencia para un dato. Sin embargo, una gran ventaja de las redes neuronales es que permiten calcular en paralelo la salida para varios datos. Por lo tanto, también se compara el tiempo de inferencia para el conjunto de test completo. Los resultados se observan en la Tabla 6.9.

Lo primero que se puede observar es que, para un dato, utilizar tanto FCNN como GNN implica tiempos significativamente menores. En particular, FCNN presentan aproximadamente tiempos tres órdenes de magnitud menores, mientras que GNN dos. Esto es una mejora notable sobre el IPOPT. A su vez, esta diferencia se amplía para el caso del conjunto de test, donde las redes neuronales aprovechan la paralelización inherente de su arquitectura. En este caso, las mejoras representan órdenes 4 y 3 veces menores para FCNN y GNN respectivamente. De esta forma, se afirma el hecho de que utilizar aprendizaje automático efectivamente resulta en soluciones significativamente más veloces. Remarcamos que aunque el tiempo de inferencia para un dato de IPOPT no es significativo para un humano (menos de 1 segundo), los órdenes de magnitud más rápidos se hacen notar cuando es necesario correr varios despachos de reactiva con diferentes entradas. En particular, observando los tiempos para el conjunto de test en la red uruguaya, utilizar un predictor entrenado resulta en tiempos menores a 1 segundo, mientras que el IPOPT demora más de 7 minutos en realizar la inferencia.

Red / Método	Tiempos de Inferencia (s)					
	Conjunto de Test			Promedio Dato		
	IPOPT	FCNN	GNN	IPOPT	FCNN	GNN
IEEE30	$3,5 \times 10^1$	$4,2 \times 10^{-3}$	$4,4 \times 10^{-2}$	$1,1 \times 10^{-1}$	$1,7 \times 10^{-4}$	$1,9 \times 10^{-3}$
IEEE118	$2,3 \times 10^2$	$3,7 \times 10^{-3}$	$1,9 \times 10^{-1}$	$5,7 \times 10^{-1}$	$1,9 \times 10^{-4}$	$2,2 \times 10^{-3}$
URU	$4,5 \times 10^2$	$1,1 \times 10^{-2}$	$1,0 \times 10^{-1}$	$4,4 \times 10^{-1}$	$2,4 \times 10^{-4}$	$3,4 \times 10^{-3}$

Tabla 6.9: Comparación de tiempos que demoran FCNN, GNN y IPOPT en inferir la solución en las tres redes de prueba. Se evalúa por un lado el promedio de tiempo que demoran en predecir un único dato de salida, y por otro, el tiempo que demoran en inferir la salida para todo el conjunto de datos de test. La cantidad de datos en el conjunto de test es 328 para la red IEEE30, 409 para la red IEEE118 y 1032 para la red uruguaya.

6.2. Minimización de Pérdidas

Similar a la estrategia anterior, se comienza explicando los detalles de implementación. En este caso, se reciclan varios de los conceptos utilizados en la sección anterior. Por lo tanto, en esta sección se explican únicamente las técnicas de implementación que presentan diferencias. Respecto a los resultados, la principal diferencia es que en este caso no se busca imitar al optimizador. Por lo tanto, no

Capítulo 6. Experimentos

hay métrica de MAE, ni gráficas comparando los voltajes predichos con los óptimos. En este caso, los resultados corresponden a observar el efecto de los distintos hiperparámetros sobre el desempeño de los modelos, junto con la calidad de estos para resolver el ORPD, comparándose contra el optimizador y la estrategia básica.

6.2.1. Implementación

Respecto a los datos, la implementación es casi idéntica que la estrategia anterior, con la diferencia de que en este caso no hay señal de salida óptima. Sin embargo, los conceptos como partición en entrenamiento, validación y test, división en lotes y normalización de entradas se repiten.

Acotación del Rango de Salidas

Respecto a la implementación de la arquitectura de los modelos, se repite el uso de la función no lineal *Leaky ReLU*, junto con la normalización por lotes. Sin embargo, en este caso se agrega una nueva función no lineal $\gamma(x, x_{\min}, x_{\max})$, utilizada únicamente en la última capa. Esta fue mencionada en el Capítulo 3, y se utiliza para asegurar que se cumplan las restricciones que limitan el rango de algunas variables. Esta función requiere ser diferenciable, y tender a x_{\min} para valores muy negativos y a x_{\max} para valores muy positivos. Para esto, se utiliza la función:

$$\gamma(x, x_{\min}, x_{\max}) = \frac{x_{\max} - x_{\min}}{1 + e^{-x}} + x_{\min} . \quad (6.7)$$

Un detalle importante a la hora de implementar esta función es que al realizar los primeros experimentos se observa que las salidas tienden a tomar valores extremos (x_{\min} o x_{\max}) al inicio del entrenamiento al tomar pesos aleatorios. Esto genera que el entrenamiento comience estancado, impidiendo que el modelo aprenda. Es por esto que, para realizar una transición más suave entre ambos extremos y así disminuir esta sensibilidad, se utiliza un factor sobre la entrada. Empíricamente se llega a que utilizar $e^{-\frac{x}{10}}$ en lugar de e^{-x} mitiga estos problemas. En la Figura 6.21 se puede observar un ejemplo de esta función para los valores $x_{\min} = 1$ y $x_{\max} = 4$.

Función de Pérdida

La implementación del algoritmo de entrenamiento es la principal diferencia con el caso anterior. En este caso, la función de pérdida no se corresponde a una medida de error relativo con las salidas del óptimo, sino que se deriva una función de pérdida a partir del problema de optimización:

6.2. Minimización de Pérdidas

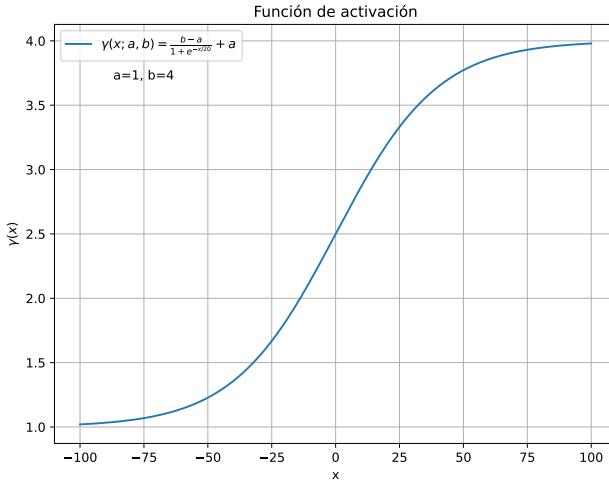


Figura 6.21: Función de activación utilizada a la salida de los modelos no supervisados para controlar que las predicciones no violen las restricciones.

$$\mathcal{L}(x_t, \Phi(x_t, \Theta)) = \mathbf{c}(x_t, \Phi(x_t, \Theta)) \quad (6.8)$$

$$+ \sum_{i=1}^n \lambda_i \phi_i(x_t, \Phi(x_t, \Theta)) \quad (6.9)$$

$$+ \sum_{i=1}^m \mu_i \psi_i(x_t, \Phi(x_t, \Theta)) . \quad (6.10)$$

Para poder implementar esta función, es necesario definir dos detalles. En primer lugar, la elección de las funciones de penalización ϕ y ψ . Se recuerda que ambas deben ser diferenciables, donde la primera debe crecer con el valor absoluto de la restricción de igualdad, mientras que la segunda debe ser nula para valores negativos, y crecer rápidamente al acercarse a 0, o al pasarse a valores positivos. Es por esto que se elige a ϕ como el cuadrado de la norma de la entrada. Es decir:

$$\phi(\mathbf{x}) = \|\mathbf{x}\|^2 . \quad (6.11)$$

La elección de ψ es levemente más complicada. Una primera idea es utilizar una penalización de barrera logarítmica. Esto es $\psi(x) = -\log(-x)$. Esta cumple que es muy cercano a 0 para valores negativos, y a medida que se acerca a 0 crece de forma acelerada. El problema de esta función es que no está definida para valores positivos. Por lo tanto, no es posible utilizarla. Para esto, se extiende la definición del logaritmo para valores negativos [29]. Esta extensión consiste en definir un valor s negativo y cercano a cero, a partir del cual se sustituye la función logarítmica por una función lineal. Esto define el logaritmo extendido $\bar{\log}_s(u)$:

Capítulo 6. Experimentos

$$\bar{\log}_s(u) := \begin{cases} \log(u), & \text{si } u \geq \frac{1}{s} \\ s \left(u + \frac{1}{s} \right) - \log \left(\frac{1}{s} \right), & \text{de lo contrario} \end{cases}. \quad (6.12)$$

Finalmente, se cumple que:

$$\psi(u) = -\bar{\log}_s(-u). \quad (6.13)$$

En la Figura 6.22 se observa un ejemplo del gráfico de esta función para $s = 0,1$, valor que será utilizado durante los experimentos numéricos.

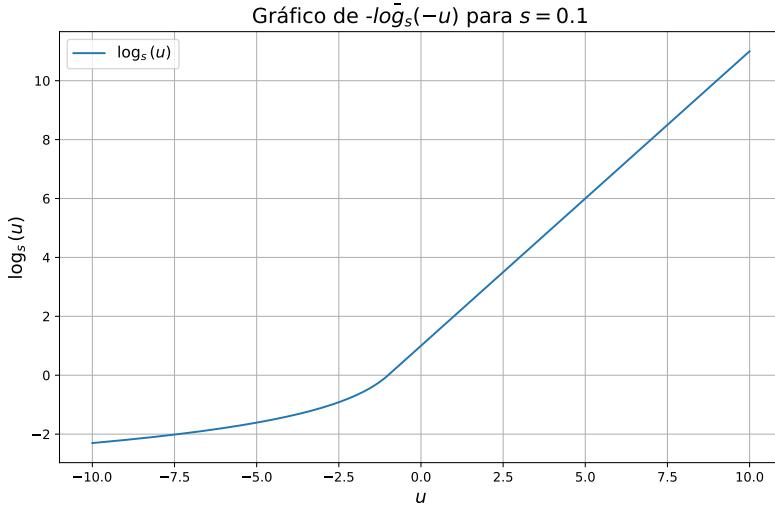


Figura 6.22: Gráfico de la función $\psi(u) = -\bar{\log}_s(-u)$ para $s = 0,1$.

El segundo detalle necesario para implementar esta métrica es elegir las ponderaciones de penalización λ_i y μ_i . Una primera opción es agregar como hiperparámetros todas estas ponderaciones. Sin embargo, la cantidad de estas es demasiado alta. Es por esto que se propone utilizar el mismo coeficiente λ_{re} para la penalización de la parte real de las ecuaciones de flujo de carga, λ_{im} para la parte imaginaria y μ_{corr} para las penalizaciones de desigualdad sobre la capacidad de las corrientes. Por lo tanto, la función queda:

$$\mathcal{L} = (\mathbf{s}_{i \rightarrow j} + \mathbf{s}_{j \rightarrow i}) \quad (6.14)$$

$$+ \lambda_{re} \phi(Re\{\mathbf{s} - diag(\mathbf{v})Y_{bus}^* \mathbf{v}^*\}) \quad (6.15)$$

$$+ \lambda_{im} \phi(Im\{\mathbf{s} - diag(\mathbf{v})Y_{bus}^* \mathbf{v}^*\}) \quad (6.16)$$

$$+ \mu_{corr} \psi(\{I_{i \rightarrow j} - I_{i \rightarrow j, \max}\}). \quad (6.17)$$

6.2. Minimización de Pérdidas

Métricas

En este caso las métricas deben reflejar por un lado las pérdidas de potencia y, por otro, la factibilidad en las soluciones. Es por esto que se propone utilizar tres métricas para comparar los modelos. Estas son: las pérdidas de potencia promedio en la red, el porcentaje de datos que resultan factibles y el máximo porcentaje de violación en las restricciones, definido en la Ecuación 6.5 de los resultados de la sección anterior. Para esto, una vez entrenado el modelo, se corre el flujo de carga para todos los datos del conjunto correspondiente y se calculan estos tres valores. Estas métricas permiten estimar de *grosso modo* la capacidad del modelo entrenado para resolver el despacho de reactiva.

6.2.2. Resultados

En este caso no hay optimizador a imitar, por lo que los resultados se centran únicamente en el desempeño de las soluciones para resolver el problema de despacho de carga. Para esto, igual que para el caso anterior se comienza presentando los hiperparámetros a elegir, junto con la opción elegida en cada caso. Se tienen tres métricas por modelo, las cuales son las pérdidas de potencia promedio, el porcentaje de datos factibles, y el máximo porcentaje de violación. Sin embargo, a la hora de realizar la búsqueda de hiperparámetros, es necesario tener un único criterio de elección, pues un modelo con buen desempeño en una, puede tener un desempeño malo en otra. El criterio utilizado es el siguiente. En primer lugar, se utilizan las dos métricas relativas a la factibilidad para filtrar parte de los modelos. Para esto, solo se toman los modelos que, al tomar la tolerancia máxima del optimizador, resultan en una factibilidad mayor a 95 %. Una vez realizada la preselección, se elige el modelo que menor promedio de pérdidas presenta. Finalmente, se hace un análisis similar al de la estrategia anterior, donde se evalúan las pérdidas en la red junto con la violación sobre las restricciones, comparándolos tanto contra el optimizador como contra la estrategia básica.

IEEE30

Los hiperparámetros a elegir son muy similares a los elegidos para la estrategia supervisada, agregándose las ponderaciones de las penalizaciones λ_{re} , λ_{im} y μ_{corr} , y descartando la normalización en la salida Norm_Y. Se observa el rango de variación, junto con la elección para cada modelo en la Tabla 6.10. En este caso, la arquitectura elegida en FCNN es $L = 3$ y $F = 128$. Por otro lado, la arquitectura elegida para GNN es $L = 3$, $K = 4$ y $F = 128$.

De esta forma quedan elegidos los modelos de FCNN y GNN. Para evaluar su desempeño, se comienza comparando los histogramas de porcentaje de violación de restricciones en la Figura 6.23. Se puede observar cómo en este caso las violaciones en FCNN y GNN superan levemente a las del óptimo, llegando a valores de hasta aproximadamente un 6 % en ambos casos. Por otro lado, en la Figura 6.24 se observa el cociente de las pérdidas de potencia para los dos modelos y la estrategia

Capítulo 6. Experimentos

Hiperparámetro	Valores	FCNN	GNN
Arquitectura			
L	2 3	3	3
F	32 128 512	128	128
K	2 4 6 8 10	-	4
Penalizaciones			
λ_{re}	10 - 10000	172.9	63.7
λ_{im}	10 - 10000	143.5	62.8
μ_{corr}	$1 \times 10^{-6} - 1 \times 10^{-2}$	$5,2 \times 10^{-5}$	$9,5 \times 10^{-6}$
Entrenamiento			
B	32 64 128	32	32
lr	$1 \times 10^{-6} - 1 \times 10^{-3}$	$2,3 \times 10^{-4}$	$4,7 \times 10^{-3}$
Normalización			
<i>Norm_B</i>	True False	False	True
<i>Norm_X</i>	True False	False	False
Regularización			
dropout	0.0 - 0.3	0.023	0.001
weight decay	0.0 - 1×10^{-3}	$6,2 \times 10^{-4}$	$1,6 \times 10^{-4}$

Tabla 6.10: Búsqueda de hiperparámetros para la red IEEE30 en el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

básica frente al óptimo. En este caso, los costos en ambos predictores son muy cercanos a uno, siendo levemente inferiores en promedio.

Esto se observa más claramente en la Tabla 6.11. Se ve cómo las pérdidas promedio obtenidas con los modelos aprendidos son menores que las obtenidas con el optimizador de PandaPower. Además, los porcentajes de factibilidad son comparables, aunque levemente inferiores para los predictores. Al relajar estos límites a la tolerancia máxima del optimizador, se tiene que los modelos aumentan su factibilidad hasta valores levemente mayores a un 95 %, coherente con el método de preselección definido previamente.

De estos resultados se concluye que este método de solución es comparable con el desempeño del optimizador. Se logran pérdidas levemente menores, pero a costa de tener porcentajes de factibilidad un poco menores. Por otro lado, comparando el desempeño entre FCNN y GNN, los valores son muy similares, aunque levemente mejores en la primera.

IEEE118

Para el caso de la red IEEE118 se procede de igual manera. La Tabla 6.12 muestra los hiperparámetros considerados para el entrenamiento, junto con los valores elegidos. La arquitectura seleccionada para FCNN es de $L = 3$ y $F = 32$,

6.2. Minimización de Pérdidas

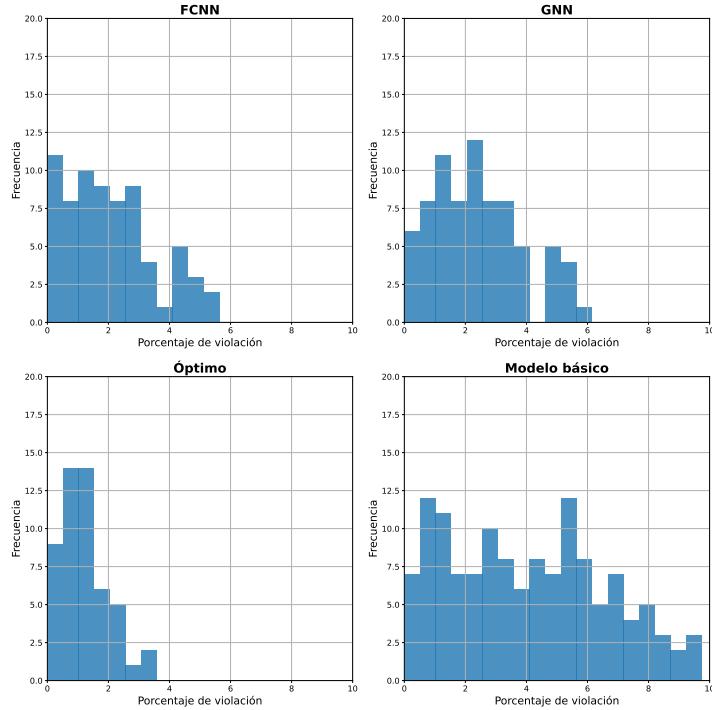


Figura 6.23: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE30 para el conjunto de datos de test, utilizando la estrategia no supervisada. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

	Pérdidas (MW)	Factibilidad	Factibilidad*(3,6 %)
Modelo base	2,84	60,8 %	80,1 %
Óptimo	2,64	84,6 %	100 %
FCNN	2,62	78,4 %	96,9 %
GNN	2,63	76,2 %	96,2 %

Tabla 6.11: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red IEEE30, utilizando la estrategia no supervisada. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 3,6 %.

mientras que GNN es de $L = 3$, $F = 128$ y $K = 2$.

En la Figura 6.25 se pueden observar los histogramas de incumplimiento de restricciones. Por un lado se tiene FCNN que presenta una violación máxima de 0,8 %, la cual es menor que el óptimo, aunque la cantidad de violaciones es mayor. Por otro, se tiene que GNN no viola ninguna restricción en ninguno de los casos. Respecto a los costos, en la Figura 6.26 se observa el cociente entre los predictores y

Capítulo 6. Experimentos

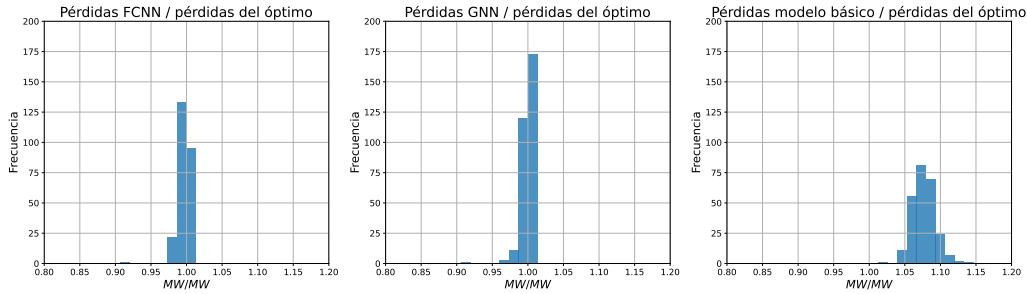


Figura 6.24: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE30 para datos de test, para la estrategia no supervisada. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.

Hiperparámetro	Valores	FCNN	GNN
Arquitectura			
L	2 3	3	3
F	32 128 512	32	128
K	2 4 6 8 10	-	2
Penalizaciones			
λ_{re}	1 - 300	17.30	2.36
λ_{im}	1 - 300	1.04	3.68
μ_{corr}	$1 \times 10^{-5} - 1 \times 10^{-1}$	$4,38 \times 10^{-4}$	$2,59 \times 10^{-5}$
Entrenamiento			
B	32 64 128	32	32
lr	$1 \times 10^{-6} - 1 \times 10^{-3}$	$1,6 \times 10^{-3}$	$9,2 \times 10^{-4}$
Normalización			
Norm_B	True False	False	True
Norm_X	True False	False	False
Regularización			
dropout	0.0 - 0.3	0.183	0.011
weight decay	$0.0 - 1 \times 10^{-3}$	$7,2 \times 10^{-5}$	$5,2 \times 10^{-5}$

Tabla 6.12: Búsqueda de hiperparámetros para red IEEE118 en el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

la estrategia básica con el óptimo. En este caso, ambos predictores presentan costos levemente inferiores al óptimo. En particular, las pérdidas en FCNN varían entre 0,95 y 1 del óptimo. Observando el resumen de estos resultados en la Tabla 6.13, se tiene en primer lugar un excelente rendimiento de GNN. No solo logra tener una factibilidad del 100 % sin tolerancia, sino que logra pérdidas levemente menores a las del optimizador. Por otro lado, FCNN presenta un gran porcentaje de no

6.2. Minimización de Pérdidas

factibles, sin embargo, todos estos se resuelven teniendo una tolerancia del 1 %. A pesar de presentar un porcentaje mayor de violación de restricciones, esta red logra un promedio de pérdidas 2 % menor que el óptimo.

Por lo tanto, en el caso de GNN se logra un predictor que es mejor que el optimizador en todo sentido, aunque por un margen leve, mientras que en el caso de FCNN su desempeño es más discutible, ya que tiene un porcentaje mayor de no factibles, pero un promedio de pérdidas menores. De todas formas, se logra en ambos un desempeño muy bueno.

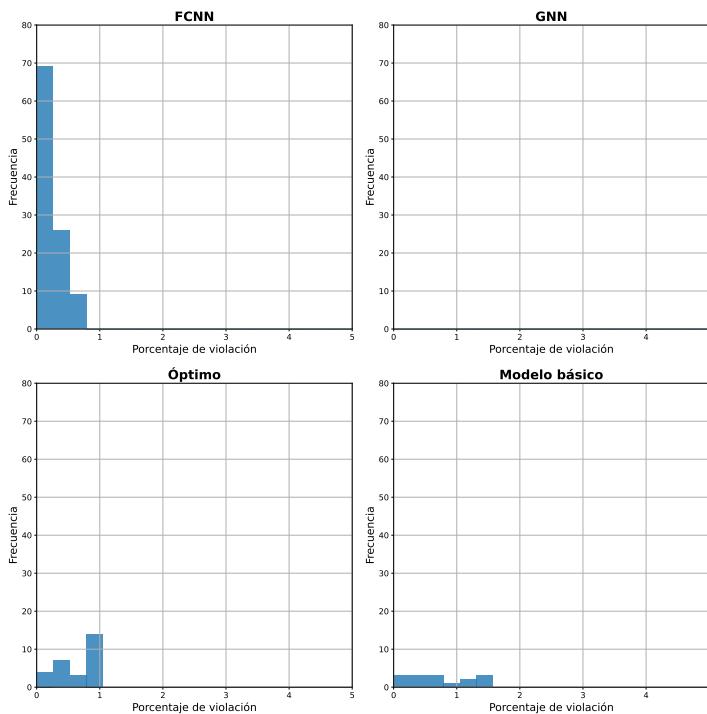


Figura 6.25: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE118 para el conjunto de datos de test, utilizando la estrategia no supervisada. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

Uruguay con Datos Sintéticos

El siguiente caso de estudio es la red uruguaya con datos sintéticos. En la Tabla 6.14 se presenta la búsqueda de hiperparámetros. En este caso, la arquitectura para GNN es $L = 2$ y $F = 1024$, mientras que para GNN es $L = 2$, $F = 512$ y $K = 4$.

En este caso, no hay figura con histograma de porcentaje de violación de restricciones. Esto se debe a que ni el óptimo, ni ninguno de los dos predictores viola alguna restricción. Respecto a las pérdidas, se puede observar en la Figura 6.27 el

Capítulo 6. Experimentos

Modelo	Pérdidas (MW)	Factibilidad	Factibilidad*(1 %)
Modelo base	96,1	96,3 %	98,6 %
Óptimo	84,9	97,0 %	100 %
FCNN	83,2	72,9 %	100 %
GNN	84,0	100 %	100 %

Tabla 6.13: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red IEEE118, utilizando la estrategia no supervisada. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,0 %.

Hiperparámetro	Valores	FCNN	GNN
Arquitectura			
L	2 3 4	2	2
F	32 128 512 1024 2048	1024	512
K	2 4 6 8 10	-	4
Penalizaciones			
λ_{re}	0.1 - 300	2.04	41.98
λ_{im}	0.1 - 300	0.34	25.79
μ_{corr}	$1 \times 10^{-6} - 1 \times 10^{-2}$	$1,07 \times 10^{-6}$	$1,7 \times 10^{-4}$
Entrenamiento			
B	128 256	256	128
lr	$1 \times 10^{-6} - 1 \times 10^{-3}$	$2,5 \times 10^{-4}$	$3,9 \times 10^{-4}$
Normalización			
$Norm_B$	True False	False	False
$Norm_X$	True False	True	False
Regularización			
dropout	0.0 - 0.5	0.42	0.01
weight decay	$0.0 - 1 \times 10^{-3}$	$6,2 \times 10^{-6}$	$5,8 \times 10^{-6}$

Tabla 6.14: Búsqueda de hiperparámetros para la red uruguaya con datos reales para el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

6.2. Minimización de Pérdidas

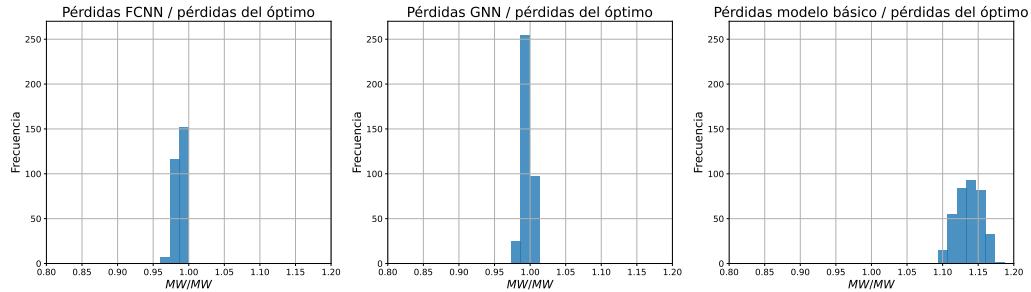


Figura 6.26: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE118 para datos de test, utilizando aprendizaje no supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y derecha con el modelo básico.

cociente entre los costos. Como se puede apreciar, en el caso de FCNN se tiene un desempeño prácticamente centrado en 1,0, mientras que para GNN está centrado cerca de 1,02.

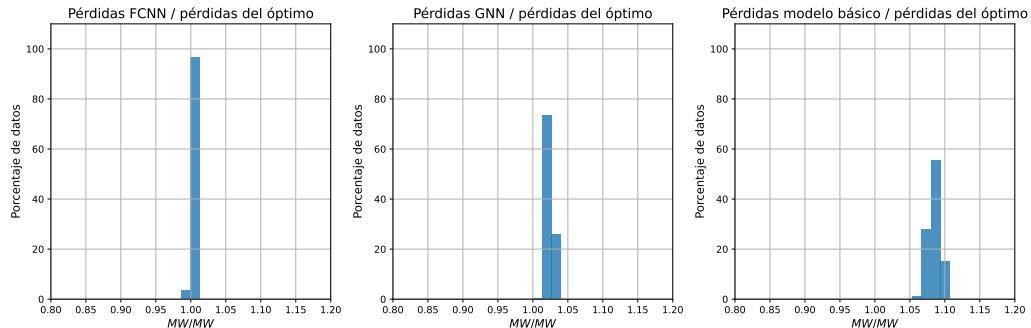


Figura 6.27: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos sintéticos para datos de test, utilizando aprendizaje no supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.

Comparando el desempeño general, en la Tabla 6.15 se observa en primer lugar el 100 % de factibilidad en los datos. A su vez, se tiene que FCNN logra costos idénticos al óptimo, mientras que GNN un 2 % mayores. De esta forma se tiene que para el caso de la red uruguaya con datos sintéticos, se logra por un lado un predictor con desempeño igual al óptimo en el caso de FCNN, mientras que para GNN el predictor final es levemente inferior.

Al resumir el desempeño de estas estrategias en casos con datos sintéticos, los resultados obtenidos son muy similares a los alcanzados al utilizar el optimizador, siendo ligeramente mejores o peores según el caso. En particular, en este contexto, el rendimiento no se corresponde con el tamaño de la red; de hecho, la red IEEE118 es la que logra el mejor desempeño. Esto es comprensible, dado que esta red posee límites de corriente más permisivos. En contraste, la red IEEE30 muestra el peor

Capítulo 6. Experimentos

	Pérdidas (MW)	Factibilidad
Modelo base (1.026)	19,0	91,1 %
Óptimo	17,5	100 %
FCNN	17,5	100 %
GNN	17,9	100 %

Tabla 6.15: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red uruguaya con datos sintéticos, utilizando la estrategia no supervisada.

rendimiento, ya que no solo tiene capacidades de corriente más restrictivas, sino que también presenta el mayor porcentaje de casos no factibles al resolver con el optimizador, indicando cierta dificultad para resolver su despacho. De este modo, se deduce que el rendimiento depende más de la dificultad para resolver el problema de optimización que del tamaño de la red. Esto es lógico, dado que se intenta resolver el problema de forma directa. De todas formas, los resultados son positivos para los tres casos, pues en general se obtiene un rendimiento comparable al del optimizador.

Uruguay con Datos Históricos

Finalmente, se tiene la red eléctrica uruguaya con datos reales. En la Tabla 6.16 se presenta la búsqueda de hiperparámetros, la cual resulta en una arquitectura de $L = 4$ y $F = 1024$ para FCNN y $L = 3$, $F = 512$ y $K = 2$ para GNN.

Se realiza el análisis de factibilidad de las soluciones, el cual se puede observar en la Figura 6.28. En este caso, los modelos de aprendizaje tienden a violar las restricciones en hasta un 5 %, a diferencia del 2 % del caso óptimo. Sin embargo, la cantidad de violaciones son mucho menores para los modelos aprendidos. Esto también se puede observar en la Tabla 6.17, donde los porcentajes de factibilidad son muy cercanos al 100 % aún sin tener en cuenta la tolerancia del óptimo. Aunque el desempeño en factibilidad es positivo, al observar los costos de estos predictores, el desempeño es inferior. Esto se puede observar en primer lugar en la Figura 6.29, donde los costos para FCNN son entre 1,0 y 1,5 veces mayores, mientras que para GNN este rango aumenta entre 1,0 y 1,1. Observando la potencia promedio, FCNN resulta en un 2 % más de costos, mientras que GNN un 4 % más.

Nuevamente, los resultados positivos obtenidos en la red uruguaya con datos sintéticos no se replican al utilizar datos reales. Aunque tanto GNN como FCNN muestran una mejora significativa en comparación con la estrategia base, siguen siendo notablemente inferiores al utilizar el optimizador de PandaPower. Dado que al emplear esta red con datos sintéticos se logra un mejor desempeño, la inferioridad se atribuye completamente al uso de datos reales. Las razones son similares a las del caso supervisado. En particular, la necesidad de más datos y las diferencias en las distribuciones entre los conjuntos de entrenamiento, validación y prueba son consideraciones importantes a tener en cuenta.

6.2. Minimización de Pérdidas

Hiperparámetro	Valores	FCNN	GNN
Arquitectura			
L	2 3 4	4	3
F	32 128 512 1024 2048	1024	512
K	2 4 6 8 10	-	2
Penalizaciones			
λ_{re}	0.1 - 300	296.8	0.39
λ_{im}	0.1 - 300	27.8	0.21
μ_{corr}	$1 \times 10^{-6} - 1 \times 10^{-3}$	$7,19 \times 10^{-5}$	$5,92 \times 10^{-6}$
Entrenamiento			
B	128 256	128	128
lr	$1 \times 10^{-6} - 1 \times 10^{-3}$	$2,3 \times 10^{-5}$	$8,5 \times 10^{-4}$
Normalización			
<i>Norm_B</i>	True False	False	True
<i>Norm_X</i>	True False	False	True
Regularización			
dropout	0.0 - 0.5	0.14	0.032
weight decay	$0.0 - 1 \times 10^{-2}$	$9,5 \times 10^{-3}$	$1,9 \times 10^{-6}$

Tabla 6.16: Búsqueda de hiperparámetros para la red uruguaya con datos históricos para el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.

	Pérdidas (MW)	Factibilidad	Factibilidad*(1,8 %)
Modelo base	30,7	37,1 %	51,9 %
Óptimo	28,3	79,8 %	100 %
FCNN	28,9	99,4 %	99,7 %
GNN	29,6	98,8 %	99,9 %

Tabla 6.17: Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red uruguaya con datos históricos, utilizando la estrategia no supervisada. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,8 %.

Capítulo 6. Experimentos

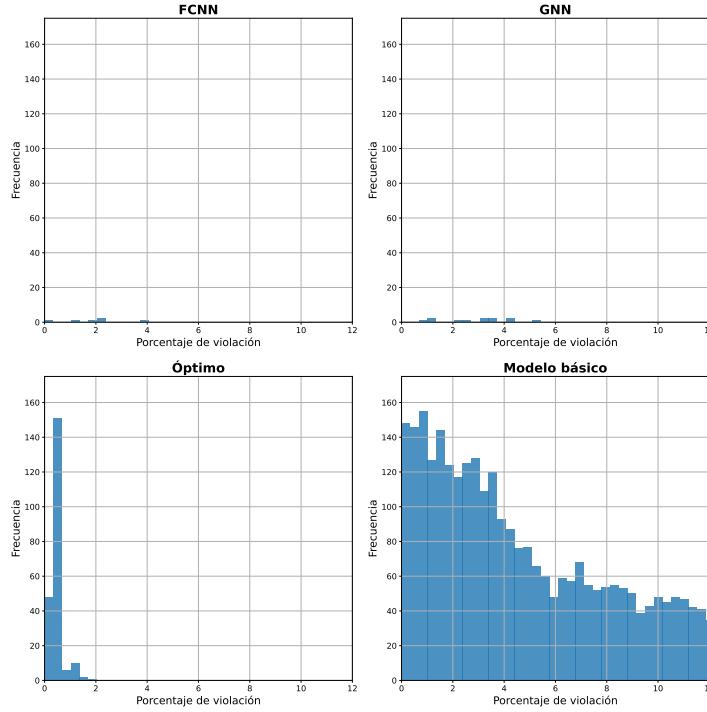


Figura 6.28: Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red uruguaya con datos históricos para el conjunto de datos de test, utilizando aprendizaje no supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.

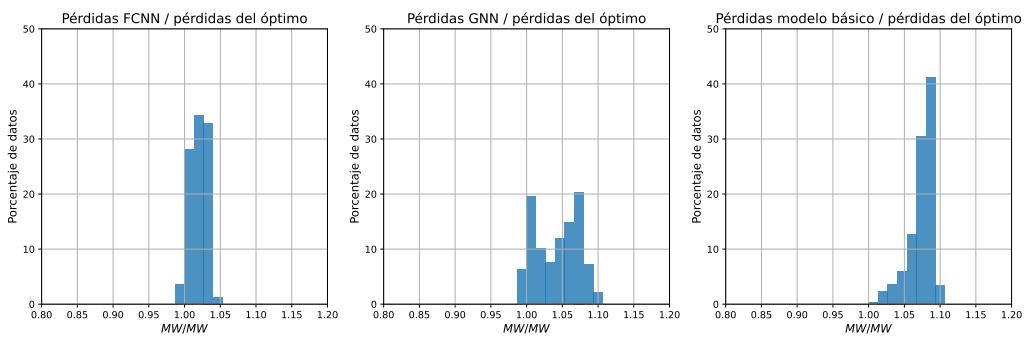


Figura 6.29: Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos históricos para datos de test, utilizando aprendizaje no supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y derecha con el modelo básico.

Capítulo 7

Conclusiones

En este trabajo, se propusieron dos estrategias de aprendizaje automático para resolver el problema de despacho óptimo de potencia reactiva en la red eléctrica uruguaya. La primera estrategia consistió en entrenar modelos para imitar el comportamiento del optimizador de punto interior tradicional, con el objetivo de replicar sus soluciones pero con tiempos de inferencia más cortos. La segunda estrategia se enfocó en minimizar directamente las pérdidas de energía reactiva utilizando una función objetivo definida específicamente para este propósito. Dentro de cada una de estas estrategias, se compararon dos arquitecturas de redes neuronales: las redes neuronales completamente conectadas y las redes neuronales sobre grafos. Estas metodologías se evaluaron tanto con datos sintéticos en redes de prueba estándar (IEEE30 e IEEE118) como en un modelo de la red real de Uruguay, utilizando tanto datos sintéticos como reales, permitiendo validar y comparar el desempeño de los modelos en distintos entornos.

En cuanto a los resultados para los casos con datos sintéticos, la estrategia supervisada demuestra un muy buen desempeño en la predicción de las variables de control del ORPD. Sin embargo, su principal limitación radica en que pequeñas imprecisiones al predecir alguna variable pueden provocar que se violen ligeramente algunas de las restricciones del sistema, especialmente aquellas que el optimizador había dejado al límite. Por lo tanto, si se desea aplicar esta estrategia, se debe tener en cuenta que permitir tiempos de ejecución mucho más cortos tiene como contrapartida que las soluciones obtenidas pueden exceder mínimamente las restricciones establecidas. Por otro lado, en el caso de la estrategia de minimización directa de pérdidas mediante aprendizaje no supervisado, se logró una solución que en todos los casos presenta un desempeño muy similar al del optimizador, teniendo donde algunas veces el desempeño es levemente superior, mientras que en otros es levemente inferior.

Sin embargo, aunque estos resultados son prometedores, su desempeño no se logra extender para el caso de los datos reales. En el caso de la estrategia supervisada, se observó un desempeño aproximadamente 10 veces peor que al utilizar datos sintéticos respecto a la capacidad de imitar las salidas del optimizador. De todas formas, las predicciones resultantes siguen siendo aceptables como solución del problema, especialmente teniendo en cuenta que resulta en tiempos de infer-

Capítulo 7. Conclusiones

cia mucho menores. Respecto a la minimización directa de pérdidas, se logró una solución que en todos los casos es superior a la estrategia básica, pero que presenta un desempeño inferior al del optimizador. Esto indica que, si bien esta estrategia mejora respecto a no realizar optimización alguna, no se alcanza el nivel óptimo que proporciona el optimizador estándar para datos reales.

Respecto a la comparación entre FCNN y GNN, los resultados obtenidos fueron similares para ambas arquitecturas, siendo GNN ligeramente inferior. Este resultado es un poco sorpresivo, ya que se esperaba que GNN, al aprovechar la estructura de grafo de la red eléctrica, pudieran superar a FCNN. Sin embargo, parte del análisis a realizar era comparar el desempeño entre estas. Este resultado sugiere que, en este caso particular, la disminución en expresividad al usar GNN de forma de permitir introducir la topología de la red no resultó beneficiosa. Es posible que se requieran problemas de mayor tamaño para que este cambio tenga un efecto positivo.

Una contribución destacada de este trabajo es el preparar y dejar público un modelo del sistema eléctrico uruguayo junto con una base de datos histórica curada. Esta base de datos proporciona casos de prueba útiles para el estudio de algoritmos y herramientas en sistemas de potencia, facilitando su desarrollo y validación. Al comparar los métodos de generación de datos sintéticos con los datos reales, se observaron grandes discrepancias en las distribuciones de datos. Esto a su vez resulta en desempeños inferiores, resaltando la importancia de utilizar datos reales para entrenar y validar modelos destinados a aplicaciones prácticas.

El principal desafío encontrado durante el desarrollo de esta tesis consistió en lograr un entendimiento profundo del problema de despacho de reactiva, junto con la obtención de la base de datos y modelo eléctrico de la red uruguaya. La falta de conocimiento inicial sobre sistemas eléctricos, junto con la falta de estructura y organización de los datos, hicieron que obtener una base de datos utilizable requiera un esfuerzo considerable. Esto resultó en una menor cantidad de dedicación a la experimentación sobre las estrategias y modelos propuestos. De todas formas, se considera que los resultados obtenidos son satisfactorios y se suman a los datos públicos como justificación para futuros trabajos en esta área. En particular, se proponen tres líneas de investigación.

Primero, implementar una metodología más eficaz para actualizar las ponderaciones de las penalizaciones en la función de pérdida utilizada en la estrategia no supervisada. Ajustar dinámicamente estos pesos, en lugar de tomarlos como hiperparámetros, puede mejorar significativamente el rendimiento del modelo al equilibrar la minimización del objetivo principal y el cumplimiento de las restricciones del sistema, acercándose así al óptimo global [9]. Segundo, se propone explorar arquitecturas más avanzadas de redes neuronales sobre grafos, incorporando mecanismos de atención como *Graph Attention Networks* [39]. Los mecanismos de atención pueden potenciar la capacidad del modelo para capturar relaciones complejas en la red eléctrica, lo que podría mejorar el desempeño en el problema del ORPD. Por último, mejorar los modelos y datos de la red eléctrica uruguaya para que sean aún más fieles a la realidad. En particular, extender el tamaño de la base de datos permitiendo mejorar el desempeño para datos reales. Esto permite

evaluar la posibilidad de integrar estos modelos en el proceso de despacho real, contribuyendo a optimizar la operación del sistema eléctrico nacional.

En conclusión, este trabajo demostró que las técnicas de aprendizaje automático pueden ser herramientas útiles para acelerar la resolución de problemas de optimización y, en particular, el problema de despacho óptimo de potencia reactiva. La capacidad de estos modelos para obtener patrones en los datos de entrada les permite imitar en gran medida las soluciones del optimizador, pero con tiempos de inferencia menores. A su vez, la disponibilidad de datos históricos reales y la aplicación de estas técnicas sirven como base para futuras investigaciones que contribuyan a mejorar la eficiencia en los sistemas eléctricos, mitigando los problemas que implica la transición hacia fuentes de energía más sostenibles.

Esta página ha sido intencionalmente dejada en blanco.

Referencias

- [1] Administración del Mercado Electrico (ADME). Generación de Electricidad por Fuente. <https://adme.com.uy/>.
- [2] Administración del Mercado Electrico (ADME). Mapa unifilar de la red de transmisión Uruguaya. https://www.adme.com.uy/mapa_uy_estaciones.php.
- [3] Siemens AG. PSS[®]e (power system simulator for engineering). <https://new.siemens.com/global/en/products/energy-energy-automation-and-smart-grid/pss-software/pss-e.html>, 2024. Accessed: 2024-07-11.
- [4] Takuya Akiba, Shu Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM, 2019.
- [5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
- [6] I. Boero, S. Díaz, and T. Vázquez. Datos doraa-uy. https://drive.google.com/drive/folders/121s67_IgW-r39hG-xwHIUI32-_QIE97X?usp=sharing, 2024. Accedido: 4-oct-2024.
- [7] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- [8] CAF - Banco de Desarrollo de América Latina. Uruguay líder en el uso de fuentes renovables en américa latina. [https://www.caf.com/es/actualidad/noticias/2021/07/uruguay-lider-en-el-uso-de-fuentes-renovables-en-americ\(latina\)/](https://www.caf.com/es/actualidad/noticias/2021/07/uruguay-lider-en-el-uso-de-fuentes-renovables-en-americ(latina)/), 2021.
- [9] Miguel Calvo-Fullana, Santiago Paternain, Luiz F. O. Chamon, and Alejandro Ribeiro. State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards. *IEEE Transactions on Automatic Control*, 69:4275–4290, 2021.

Referencias

- [10] Spyros Chatzivasileiadis. Lecture notes on optimal power flow (opf). <https://arxiv.org/abs/1811.00943>, 2018.
- [11] Carleton Coffrin, Russell Bent, Kaarthik Sundar, Yeesian Ng, and Miles Lubin. Powermodels.jl: An open-source framework for exploring power flow formulations. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–8, June 2018.
- [12] George B Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.
- [13] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Generalized louvain method for community detection in large networks. In *2011 11th International Conference on Intelligent Systems Design and Applications*. IEEE, November 2011.
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [15] Facultad de Ingeniería, Universidad de la República. Introducción al Sistema Eléctrico Uruguayo. https://eva.fing.edu.uy/pluginfile.php/182676/mod_resource/content/7/introduccion_2023.pdf.
- [16] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from IC DATA 2020 and IKE 2020*, pages 877–894, 2021.
- [17] N. Fiasorgbor. Investigating the effects of slack bus selection in load flow studies: A comparative analysis of robust and weak grids. *Journal of Power and Energy Engineering*, 11:1–14, 2023.
- [18] Mevludin Glavic and Fernando L. Alvarado. An extension of newton–raphson power flow problem. *Applied Mathematics and Computation*, 186(2):1192–1204, 2007.
- [19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [20] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 3rd edition, 2022.
- [21] IEEE Power Energy Society. Ieee 118-bus test case. <https://icseg.iti.illinois.edu/ieee-118-bus-system/>, 1993.
- [22] IEEE Power Energy Society. Ieee 30-bus test case. <https://icseg.iti.illinois.edu/ieee-30-bus-system/>, 1993.

Referencias

- [23] Savio D Immanuel and Udit Kr. Chakraborty. Genetic algorithm: An approach on optimization. In *2019 International Conference on Communication and Electronics Systems (ICCES)*, pages 701–708, 2019.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.
- [25] Richard Lincoln. Pypower. <https://github.com/rwl/PYPOWER>, 2023.
- [26] United Nations. Renewable energy. <https://www.un.org/es/climatechange/raising-ambition/renewable-energy>. Accessed: 19-09-2024.
- [27] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), February 2004.
- [28] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. Optimal power flow using graph neural networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5930–5934, 2020.
- [29] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. Unsupervised optimal power flow using graph neural networks. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6885–6889, 2024.
- [30] Pandapower. Pandapower - validaciones y pruebas. <https://www.pandapower.org/about/#analysis>, 2024.
- [31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 8024–8035, 2019.
- [32] Jeff Reback et al. pandas: Powerful data structures for data analysis, time series, and statistics. <https://pandas.pydata.org/>, 2023. Version 2.0.3.
- [33] Isaac A. Samuel, Okwechime Ngozi Marian, and Ademola Abudulkareem. Investigating the selection of a suitable slack bus: A case study of the multi-generating stations of the nigerian 330-kv power system network. *International Journal of Electrical Electronic Engineering Studies*, 2(1):1–12, September 2014. Available at www.ea-journals.org.

Referencias

- [34] Leonardo F. S. Scabini and Odemir M. Bruno. Structure and performance of fully connected neural networks: Emerging complex network properties. *CoRR*, abs/2107.14062, 2021.
- [35] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun. pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, Nov 2018.
- [36] Uruguay XXI. Informe energía renovable en uruguay. <https://www.uruguayxxi.gub.uy/uploads/informacion/79870b5679e4f9634944f6b8daca8ee6c3d45df.pdf>, 2023. Consultado el: 19 de septiembre de 2024.
- [37] UTE. Tarifario UTE 2023. <https://portal.ute.com.uy/sites/default/files/docs/Pliego%20Tarifario%20Vigente.pdf>.
- [38] UTE. UTE en Cifras 2023. https://www.ute.com.uy/sites/default/files/generico/UTE_EN_CIFRAS_2023_final%20Imprenta.pdf.
- [39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.
- [40] Liang Wang, Jingying Yang, Qing Zhang, Di Zhang, Yongqing Huang, Wei Li, and Bonian Shi. Research on coordinated reactive power and voltage control strategy for regional power grids with high penetration of renewable energy. In *2022 IEEE/IAS Industrial and Commercial Power System Asia (ICPS Asia)*, pages 1160–1165, 2022.
- [41] Jia Zhao, Li Yu, Jing-Ru Li, and Peng Zhou. Identifying influential nodes based on graph signal processing in complex networks*. *Chinese Physics B*, 24(5):058904, mar 2015.
- [42] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434, 2018.

Índice de tablas

6.1. Búsqueda de hiperparámetros para la red IEEE30 para los entrenamientos de aprendizaje supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	98
6.2. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos de aprendizaje supervisado para la red IEEE30. El caso factibilidad* corresponde al porcentaje de soluciones factibles, relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a 3,6 %.	102
6.3. Búsqueda de hiperparámetros para la red IEEE118 para los entrenamientos de aprendizaje supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	103
6.4. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos para la red IEEE118, utilizando el método supervisado. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,0 %.	107
6.5. Búsqueda de hiperparámetros para la red uruguaya con datos sintéticos para los modelos de aprendizaje supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	109
6.6. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos de aprendizaje supervisado para la red uruguaya con datos sintéticos.	109

Índice de tablas

6.7. Búsqueda de hiperparámetros para la red uruguaya estrategia supervisada con datos históricos. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	113
6.8. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos para la red uruguaya con datos históricos, utilizando el método supervisado. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,8 %.	118
6.9. Comparación de tiempos que demoran FCNN, GNN y IPOPT en inferir la solución en las tres redes de prueba. Se evalúa por un lado el promedio de tiempo que demoran en predecir un único dato de salida, y por otro, el tiempo que demoran en inferir la salida para todo el conjunto de datos de test. La cantidad de datos en el conjunto de test es 328 para la red IEEE30, 409 para la red IEEE118 y 1032 para la red uruguaya.	119
6.10. Búsqueda de hiperparámetros para la red IEEE30 en el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	124
6.11. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red IEEE30, utilizando la estrategia no supervisada. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 3,6 %.	125
6.12. Búsqueda de hiperparámetros para red IEEE118 en el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente. . . .	126
6.13. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red IEEE118, utilizando la estrategia no supervisada. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,0 %.	128

Índice de tablas

6.14. Búsqueda de hiperparámetros para la red uruguaya con datos reales para el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	128
6.15. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red uruguaya con datos sintéticos, utilizando la estrategia no supervisada.	130
6.16. Búsqueda de hiperparámetros para la red uruguaya con datos históricos para el caso no supervisado. La primera columna muestra el hiperparámetro, mientras que la segunda especifica el rango de valores explorados. Las columnas tercera y cuarta presentan los valores obtenidos para el mejor modelo en las arquitecturas FCNN y GNN, respectivamente.	131
6.17. Comparación de pérdidas en la red y factibilidad de los resultados para los distintos modelos, para la red uruguaya con datos históricos, utilizando la estrategia no supervisada. El caso factibilidad* corresponde al porcentaje de soluciones factibles relajando las restricciones lo necesario para que las soluciones del optimizador sean 100 % factibles, que en este caso corresponde a un 1,8%.	131

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

2.1. Modelo π de las líneas y flujo de potencias aparentes.	12
2.2. Modelo π de los transformadores.	13
2.3. Subred que se usará de ejemplo ilustrativo en esta sección. Consiste en cuatro barras, un generador de voltaje, tres cargas y un generador controlador de voltaje.	14
3.1. Esquema del flujo de entrenamiento de un algoritmo de aprendizaje automático.	32
3.2. Esquema de un perceptrón.	34
3.3. Esquema de interconexión de las distintas capas en una red neuronal.	36
3.4. Esquema de 3 capas anidadas de una red neuronal en grafos.	45
3.5. Permutación de un grafo.	47
4.1. Esquema unifilar de la red eléctrica IEEE-30 [22].	57
4.2. Esquema de la red eléctrica IEEE-118 [21].	58
4.3. Diagrama en PSSE de la red eléctrica uruguaya.	65
5.1. Histograma de $P_{GEN_{TOT}} - P_{LOAD_{TOT}}$	70
5.2. Histograma de generación de potencia activa en 3 generadores de la red uruguaya con datos sintéticos.	71
5.3. Histograma de generación de potencia activa de los 4 generadores eólicos más influyentes de la red.	71
5.4. Histograma de generación de potencia activa de los 4 generadores hidráulicos más influyentes de la red.	72
5.5. Histograma de generación de potencia activa de los 4 parques fotovoltaicos más influyentes de la red.	73
5.6. Histograma de generación de potencia activa de los 4 generadores biomasa más influyentes de la red.	73
5.7. Histograma de generación de potencia activa de los 4 generadores térmicos más influyentes de la red.	74
5.8. Generación de potencia activa promedio por año, estación y hora para cada generador.	75

Índice de figuras

5.9. Distribución de las 9 cargas con mayor consumo en la red uruguaya. Estas corresponden, en orden de izquierda a derecha y descendiente, a: Central de Montevideo E, Bifuracación, Central Norte, Central de Montevideo F, Central de Montevideo C, Solymar, Central de Montevideo H, Pando y Maldonado.	77
5.10. Carga total por año, estación y hora.	77
5.11. Histogramas de generación o consumo óptimo en los compensadores de reactiva. Estas corresponden, de izquierda a derecha y en orden descendente, a: Salto Grande Uruguay, Melo, Central Montevideo A, Central Montevideo I, Palmar y San Carlos.	78
5.12. Histogramas de voltajes óptimos para 9 generadores de la red eléctrica uruguaya	79
5.13. Histogramas de voltajes óptimos para 9 generadores de la red uruguaya con datos sintéticos. Estas corresponden, de izquierda a derecha y en orden descendente, a: Palmar, Juan P. Terra, Central Montevideo L, Bifuración, Rocha, Baygorria, Bonete, Bonete B y Pintado B.	80
5.14. Mapa del Uruguay con indicador de centralidad de grado en cada nodo de la red. Cuanto más intenso el color del nodo, implica una mayor centralidad. Notar que la medida de centralidad se encuentra normalizada, tomando valores entre 0 y 1.	82
5.15. Mapa del Uruguay con indicador de centralidad <i>betweenness</i> . La escala de colores es similar al caso de centralidad de grado.	83
5.16. Mapa del Uruguay con indicador de centralidad de cercanía. La escala de colores es similar al caso de centralidad de grado.	85
5.17. Indicador de centralidad de vector propio mediante escala de colores.	86
5.18. Mapa del Uruguay con indicador de centralidad utilizando la señal de potencia. La escala de colores es similar al caso de centralidad de grado.	87
5.19. Comunidades detectadas mediante el algoritmo de Girvan-Newman.	90
5.20. Detección de comunidades mediante el método de Louvain.	91
 6.1. Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red IEEE30 en los datos de test, entrenada mediante aprendizaje supervisado. Se muestran 4 de los 6 generadores totales elegidos al azar.	99
6.2. Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red IEEE30 en los datos de test en aprendizaje supervisado. Se muestran 4 de los 6 generadores totales elegidos al azar.	100

Índice de figuras

6.3. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE30 para el conjunto de datos de test, utilizando el método supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	101
6.4. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE30 para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.	102
6.5. Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red IEEE118 en los datos de test. Se muestran 4 de los generadores totales elegidos al azar.	104
6.6. Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red IEEE118 en los datos de test. Se muestran 4 de los generadores totales elegidos al azar.	105
6.7. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE118 método supervisado para el conjunto de datos de test. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	106
6.8. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE118 para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.	107
6.9. Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red uruguaya con datos sintéticos en los datos de test, mediante aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.	110
6.10. Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red uruguaya con datos sintéticos en los datos de test entrenada con aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.	111
6.11. Comparación entre las potencias de los compensadores óptimas (naranja) y predichas por FCNN (azul) para la red uruguaya con datos sintéticos en los datos de test, entrenadas mediante aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.	112

Índice de figuras

6.12. Comparación entre la potencia de los compensadores óptimas (naranja) y predichas por GNN (azul) para la red uruguaya con datos sintéticos en los datos de test, para entrenamientos de aprendizaje supervisado. Se muestran 6 de los generadores totales elegidos al azar.	113
6.13. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red uruguaya con datos sintéticos para el conjunto de datos de test, utilizando el método supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	114
6.14. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos sintéticos para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.	114
6.15. Comparación entre voltajes óptimos (naranja) y predichos por FCNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.	115
6.16. Comparación entre voltajes óptimos (naranja) y predichos por GNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.	115
6.17. Comparación entre las potencias en los compensadores óptima (naranja) y predicha por FCNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.	116
6.18. Comparación entre las potencia en los compensadores óptima (naranja) y predicha por GNN (azul) para la red eléctrica uruguaya con datos históricos en los datos de test. Se muestran 6 de los generadores totales elegidos al azar.	116
6.19. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red eléctrica uruguaya con datos históricos para el conjunto de datos de test, utilizando el método supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	117
6.20. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos históricos para datos de test, utilizando el método supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.	117

Índice de figuras

6.21. Función de activación utilizada a la salida de los modelos no supervisados para controlar que las predicciones no violen las restricciones.	121
6.22. Gráfico de la función $\psi(u) = -\log_s(-u)$ para $s = 0,1$.	122
6.23. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE30 para el conjunto de datos de test, utilizando la estrategia no supervisada. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	125
6.24. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE30 para datos de test, para la estrategia no supervisada. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.	126
6.25. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red IEEE118 para el conjunto de datos de test, utilizando la estrategia no supervisada. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	127
6.26. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red IEEE118 para datos de test, utilizando aprendizaje no supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y derecha con el modelo básico.	129
6.27. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos sintéticos para datos de test, utilizando aprendizaje no supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y a la derecha con el modelo básico.	129
6.28. Histograma de porcentaje de violación de las restricciones para las distintas estrategias en la red uruguaya con datos históricos para el conjunto de datos de test, utilizando aprendizaje no supervisado. Arriba a la izquierda para FCNN y a la derecha para GNN, y debajo a la izquierda para el óptimo y a la derecha para el modelo básico. El eje x corresponde al porcentaje de violación mientras que el eje y corresponde a la cantidad de restricciones que son violadas en tal porcentaje.	132
6.29. Histograma del cociente de pérdidas entre los distintos modelos obtenidos contra las pérdidas del óptimo en la red uruguaya con datos históricos para datos de test, utilizando aprendizaje no supervisado. A la izquierda se compara el óptimo con FCNN, en el medio con GNN y derecha con el modelo básico.	132

Esta es la última página.
Compilado el lunes 14 octubre, 2024.
<http://iie.fing.edu.uy/>