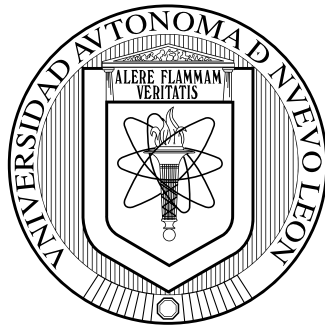


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



EVALUACIÓN TEMPRANA DE ESTABILIDAD Y SEGURIDAD DE
SISTEMAS DE POTENCIA CON BAJA INERCIA UTILIZANDO MACHINE
LEARNING

por

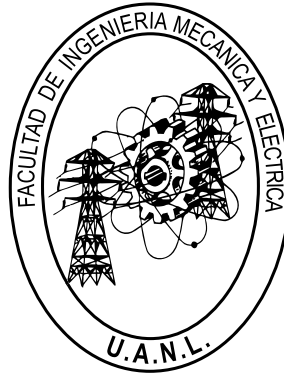
JESÚS ALEJANDRO VERDUZCO COTA

Como requisito para obtener el grado de MAESTRÍA EN
CIENCIAS DE LA INGENIERÍA ELÉCTRICA

Noviembre de 2021

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



EVALUACIÓN TEMPRANA DE ESTABILIDAD Y SEGURIDAD DE
SISTEMAS DE POTENCIA CON BAJA INERCIA UTILIZANDO MACHINE
LEARNING

por

JESÚS ALEJANDRO VERDUZCO COTA

Como requisito para obtener el grado de MAESTRÍA EN
CIENCIAS DE LA INGENIERÍA ELÉCTRICA

Noviembre de 2021

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Facultad de Ingeniería Mecánica y Eléctrica
Posgrado

Los miembros del Comité de Evaluación de Tesis recomendamos que la Tesis "Evaluación Temprana de Estabilidad y Seguridad de Sistemas de Potencia con Baja Inercia Utilizando *Machine Learning*", realizada por el estudiante Jesús Alejandro Verduzco Cota, con número de matrícula 2032338, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica.

El Comité de Evaluación de Tesis

Dr. Manuel Antonio Andrade Soto
Director

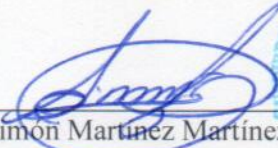
Dr. Francisco Manuel Gonzalez-Longatt
Co-director

Dr. Ernesto Vázquez Martínez
Revisor

Dra. Gina María Idárraga Ospina
Revisor

Dr. José Luis Rueda Torres
Revisor

Vo.Bo.


Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado



Institución 190001

Programa 557520

Acta Núm. 3963

Ciudad Universitaria, a 1 de noviembre de 2021.

Este trabajo es dedicado a mis padres, Oscar y Reyna, quienes me han apoyado en todo momento. Gracias por su esfuerzo y amor.

AGRADECIMIENTOS

A mis asesores el Dr. Manuel Antonio Andrade Soto y Dr. Francisco Manuel González-Longatt, por compartir sus conocimientos y guiarme constantemente en la realización de esta investigación. Agradezco enormemente su paciencia, dedicación e interés en este trabajo.

A mis padres y hermano, por su apoyo en esta etapa.

A la Universidad Autónoma de Nuevo León, la Facultad de Ingeniería Mecánica y Eléctrica, a los profesores del Posgrado en Ingeniería Eléctrica y mis compañeros de la maestría, en especial Fernanda y Alan.

Al Consejo Nacional de Ciencia y Tecnología por el apoyo financiero otorgado para cursar este posgrado.

A Alexandra Elbakyan, por facilitar el acceso al conocimiento.

RESUMEN

Jesús Alejandro Verduzco Cota

Fecha de graduación: noviembre de 2021

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Título del estudio: EVALUACIÓN TEMPRANA DE ESTABILIDAD Y SEGURIDAD DE SISTEMAS DE POTENCIA CON BAJA INERCIA UTILIZANDO MACHINE LEARNING

Número de páginas: 73

Candidato para el grado de Maestría en Ciencias de la Ingeniería Eléctrica

Propósito y método de estudio: Los sistemas eléctricos de potencia están sufriendo cambios debido a su modernización y la creciente incorporación de fuentes de energía renovable basadas en convertidores electrónicos de potencia. La sustitución de generadores síncronos con generación basada en convertidores reduce la inercia rotacional disponible para contrarrestar los cambios de frecuencia. Con una reducción de inercia, los cambios son más rápidos y drásticos, por lo que se propone desarrollar herramientas de aprendizaje automático que ayuden a tomar decisiones de forma temprana en escenarios de baja inercia.

Contribuciones y conclusiones: La metodología desarrollada para entrenamiento de redes neuronales artificiales dio como resultado dos herramientas distintas que ayudan en la evaluación temprana de estabilidad de frecuencia en sistemas eléctricos de potencia de baja inercia. Las redes neuronales demostraron capacidad para comprender satisfactoriamente las relaciones complejas de las redes eléctricas aun en condiciones de baja inercia y proporcionar información relevante de forma veloz y precisa.

Asesor: Dr. Manuel Antonio Andrade Soto.

TABLA DE CONTENIDO

Capítulo	Página
1. Introducción	1
1.1. Motivación	1
1.2. Antecedentes	2
1.3. Objetivos	4
1.4. Estructura de la tesis	5
2. Estado del arte	7
2.1. Introducción	7
2.1.1. Inercia	7
2.2. Trabajo actual	10
3. Aprendizaje automático	15
3.1. Introducción	15
3.2. Aplicaciones	16
3.3. Tipos	18
3.3.1. Aprendizaje supervisado	18
3.3.2. Aprendizaje no supervisado	19
3.3.3. Aprendizaje por refuerzo	20
3.4. Generalización, sobreajuste y subajuste	21
3.5. Técnicas	22
3.5.1. K vecinos más cercanos	22
3.5.2. Regresión lineal	23
3.5.3. Redes neuronales artificiales	24
3.5.4. Máquinas de vectores de soporte	27
4. Detección e identificación de desconexión de unidad generadora en condiciones de baja inercia usando redes neuronales	28
4.1. Introducción	28

4.2. Metodología	31
4.2.1. Origen de los datos	31
4.2.2. Organización de los datos	33
4.2.3. Creación de la RNA	35
4.2.4. Entrenamiento	36
4.3. Resultados	38
4.4. Conclusión	42
5. Predicción de frecuencia mínima después de desconexión de generador en condiciones de baja inercia	44
5.1. Introducción	44
5.2. Metodología	46
5.2.1. Origen de los datos	46
5.2.2. Organización de los datos	47
5.2.3. Creación de la RNA	49
5.2.4. Entrenamiento	50
5.3. Resultados	51
5.3.1. TDNN entrenada sin ruido	52
5.3.2. TDNN entrenada con ruido	55
5.3.3. TDNN con reducción de muestreo y entradas	58
5.4. Conclusión	63
6. Conclusión	65
6.1. Conclusiones	65
6.2. Recomendaciones para futuros trabajos	67
Referencias	68

LISTA DE FIGURAS

Figura	Página
1. Comportamiento de la frecuencia con diferentes cantidades de inercia en el sistema	9
2. Ejemplo de clasificador de figuras geométricas	18
3. Ejemplo de agrupamiento	19
4. Interacción entre agente y ambiente	20
5. Diferentes niveles de ajuste de modelo.	22
6. Ejemplo de clasificación con KNN	23
7. Ejemplo de regresión lineal con una variable dependiente y una independiente	24
8. Esquema de un perceptrón	26
9. Perceptrón multicapa	26
10. Separación de datos sin SVM y con SVM	27
11. Diagrama unifilar sistema New England	32
12. Estructura de RNA MLP para detección de desconexión de generador	36
13. Evolución del error	39
14. Gradiente y pruebas de validación durante el entrenamiento	40
15. Ejemplo de salida de la RNA entrenada al detectar desconexión del generador 1	41
16. Ejemplo de salida de la RNA entrenado al detectar desconexión del generador 4, en caso de inercia reducida	42
17. Ejemplo de salida de la RNA entrenado al detectar desconexión del generador 9, en caso de inercia reducida	42
18. Evolución de los sistemas eléctricos de potencia	44
19. Esquema de TDNN con ventana de t a $t-d$	49
20. TDNN con ventana de $t-1$ a $t-10$	50
21. Comportamiento del ECM en entrenamiento de TDNN sin ruido	52
22. Gradiente, μ y pruebas de validación durante el entrenamiento	53
23. Predicción de frecuencia mínima con TDNN entrenada sin ruido recibiendo entradas sin ruido	54
24. Ángulo original y con ruido añadido	54

25. Predicción de frecuencia mínima con TDNN entrenada sin ruido recibiendo entradas con ruido	55
26. Comportamiento del ECM en entrenamiento de TDNN con ruido	56
27. Gradiente, μ y pruebas de validación durante el entrenamiento con ruido .	56
28. Predicción de frecuencia mínima con TDNN entrenada con ruido recibiendo entradas sin ruido	57
29. Predicción de frecuencia mínima con TDNN entrenada con ruido recibiendo entradas con ruido	58
30. Desarrollo de ECM en entrenamiento de TDNN con ruido, reducción de entradas y muestreo	60
31. Gradiente, μ y pruebas de validación para TDNN con ruido, reducción de entradas y muestreo	60
32. Predicción de frecuencia mínima con TDNN entrenada con ruido y reducción de muestreo y entradas recibiendo señales sin ruido	61
33. Predicción de frecuencia mínima con TDNN entrenada con ruido y reducción de muestreo y entradas recibiendo señales con ruido	62
34. Predicción de frecuencia mínima con TDNN entrenada con ruido y reducción de muestreo y entradas, sin reducción de inercia	62

LISTA DE TABLAS

Tabla	Página
1. Límites de los criterios de entrenamiento	38
2. Límites de los criterios de entrenamiento	51
3. Error cuadrático medio para conjuntos	57
4. Resultados pruebas desconexión G1 y $H_{tot} = 90\%$	58
5. Límites de los criterios de entrenamiento TDNN con reducción de muestreo y entradas	59

CAPÍTULO 1

INTRODUCCIÓN

1.1 Motivación

La electricidad desempeña un papel fundamental en las sociedades actuales, ya que no se concibe un desarrollo económico apropiado si no se cuenta con suministro de energía eléctrica suficiente y confiable.

En los últimos años, los sistemas eléctricos de potencia (SEP), han sufrido cambios para ajustarse a la creciente integración de fuentes de energía renovable (FER) a la red. Otros motivos de la actualización de los SEP incluyen los sistemas de almacenamiento de energía, mercados eléctricos, generación distribuida, equipos de medición y comunicación, entre otros.

Las FER, las cuales en su gran mayoría están conectadas a la red eléctrica por medio de convertidores electrónicos de potencia, están desplazando a las centrales de generación basadas en generadores síncronos [1]. Esto lleva a que las redes eléctricas modernas cuenten cada vez con menos inercia rotacional, lo que significa que los cambios de frecuencia serán más rápidos y, a su vez, más drásticos [2].

Esta modificación en el cambio de la red provoca que los operadores del sistema eléctrico, antes encargados de la operación estable y confiable del SEP, se encuentren en necesidad de buscar nuevas herramientas que permitan monitorear el SEP en tiempo real y que éstas sean eficientes, rápidas y precisas.

En los últimos años, el aprendizaje automático (AA) o *machine learning*, ha demostrado gran capacidad y eficacia en diferentes áreas, incluyendo el análisis de sistemas eléctricos de potencia. En la literatura, distintos trabajos de investigación enfocados a SEP permiten observar que las herramientas de AA tienen la habilidad de aprender y entender las complejas características de las variaciones de carga, información de la red, flujo de potencia, etc. que caracterizan el comportamiento natural de los SEP modernos. Diversas técnicas de AA como redes neuronales artificiales (RNA), árboles de decisión (AD), análisis de componentes principales (ACP), máquinas de soporte vectorial (MSV), redes neuronales convolucionales (RNC) han sido propuestas en estudios relacionados con disturbios, seguridad y estabilidad del SEP [3].

1.2 Antecedentes

La reciente popularidad de las técnicas de AA ha provocado que abunden los trabajos de investigación dónde se usan dichas herramientas en el contexto de los SEP. Una revisión de la literatura disponible permite apreciar las principales áreas en que se pueden clasificar las aplicaciones del AA en los sistemas eléctricos de potencia:

El primer grupo enfoca sus esfuerzos en la detección de ataques informáticos a redes de comunicación y monitoreo de la red eléctrica. En [4], Wang *et al.* establecen un mecanismo de detección de ataques coordinados (tanto físico como informático), basado en técnicas de aprendizaje profundo (AP), donde las desviaciones considerables en la información tomada por unidades de medición fasorial (PMU, del inglés *Phasor Measurement Unit*) se inspeccionan, lo cual resulta en una rápida detección y aislamiento de ataques informáticos. [5] presenta un módulo de detección de intrusos capaz de detectar tráfico malicioso en un sistema de supervisión, control y adquisición de datos. Utiliza una MSV de una clase para esta tarea. El módulo MSV de una clase es entrenado fuera de línea y detecta anomalías en tiempo real.

El segundo grupo corresponde al área de la calidad de la energía. En [6], los autores presentan un enfoque novedoso para la detección y clasificación de señales originadas por disturbios de calidad de la energía, utilizando análisis espectral singular, transformada curvelet y RNC profundas.

La tercera clasificación de las aplicaciones del AA en los SEP está involucrada en el pronóstico de demanda y generación por parte de FER. En [7], Kong *et al.* proponen una metodología basada en RNA recurrentes de memoria de corto plazo (LSTM, del inglés *long short-term memory*) para la tarea de predecir la demanda de energía de usuarios individuales a corto plazo. Al ser comparado con otras metodologías usadas para la misma función, el esquema propuesto basado en RNA tipo LSTM supera en rendimiento a los demás métodos al predecir la demanda a corto plazo de hogares residenciales individuales. En el trabajo [8], se propone un modelo de alta precisión basado en RNA profundas para pronosticar la potencia de salida de un sistema fotovoltaico utilizando información meteorológica como temperatura, radiación solar y datos históricos de potencia de salida del sistema fotovoltaico.

El cuarto grupo corresponde a la evaluación dinámica de seguridad y estabilidad de la red eléctrica. Los autores en [9] proponen un esquema que combina el método de elipsoide característico con AD para una rápida clasificación de estabilidad transitoria después de una gran perturbación en el SEP. El esquema propuesto requiere de poca cantidad de información y demuestra alta precisión de clasificación. En [10], se propone un enfoque basado en una RNA en conjunto con el algoritmo competitivo imperialista para el monitoreo en línea de inestabilidad de voltaje a largo plazo. El enfoque propuesto usa los fasores de voltaje medidos por PMU como vectores de entrada. Se utilizó el sistema de prueba New England para demostrar la utilidad del enfoque. El presente trabajo de tesis se desenvuelve en este grupo.

El quinto grupo se centra en la detección y protección ante fallas en la red. En [11]

se presenta un enfoque de AA semisupervisado para clasificación de fallas en sistemas de transmisión y distribución. Las técnicas de AA usadas en este trabajo son AD y el método de k vecinos más cercanos (KNN, del inglés *k-nearest neighbors*). [12] muestra un nuevo enfoque que emplea una combinación de funciones wavelet y de AA para clasificación de fallas en microrredes. Se usan cuatro diferentes herramientas de clasificación (AD, KNN, MSV y Naive Bayes (NB)) para automatizar la tarea de clasificación de fallas en microrredes y sus desempeños son comparados. Concluye que la técnica KNN demostró mayor precisión de clasificación.

El tema del sexto grupo es la planificación y mercados de eléctricos. En [13], los autores proponen un método novedoso para el pronóstico de corto plazo del precio de la electricidad basado en una red híbrida de dos etapas usando mapa autoorganizado (SOM, del inglés *self-organizing map*) y MSV. El método propuesto demostró eficacia y eficiencia para aprender y pronosticar los precios de la electricidad por hora del día siguiente.

El uso del AA en los SEP está diversificado y se encuentra en crecimiento. Lo que se aprecia de esta revisión de literatura es la falta de consideración de escenarios en los que la inercia total del sistema se vea reducida, una situación cada vez más común y que afecta al comportamiento de la red.

1.3 Objetivos

El objetivo general es desarrollar herramientas de clasificación y predicción que permitan la evaluación temprana de la estabilidad de frecuencia y seguridad de sistemas eléctricos de potencia con inercia reducida utilizando técnicas de aprendizaje automático.

Los objetivos específicos son:

- Estudiar diferentes enfoques basados en AA para la evaluación de estabilidad en

escenarios previamente diseñados.

- Probar diferentes herramientas de AA para los datos disponibles.
- Implementar algoritmos y modelos en MATLAB.
- Validar funcionamiento de los modelos desarrollados.

1.4 Estructura de la tesis

El presente trabajo de tesis consta de seis capítulos, los cuales se describen a continuación:

En el Capítulo 1 se presenta la motivación esta tesis, donde se describen las principales razones que dan origen al proyecto. También se hace un resumen del trabajo de investigación relacionado con la aplicación del aprendizaje automático para solucionar problemas en sistemas eléctricos de potencia.

En el Capítulo 2 se presenta una revisión de literatura reciente para establecer el estado actual del conocimiento enfocado a la evaluación de estabilidad en redes eléctricas por medio del aprendizaje automático supervisado.

En el Capítulo 3 se establecen las bases de conocimiento del aprendizaje automático. Se incluyen conceptos clave, se describen brevemente los algoritmos más usados y se muestran las diferentes aplicaciones que se les da en la actualidad a estas herramientas.

En el Capítulo 4 se presenta la primera parte del proyecto la cual tiene como objetivo entrenar una red neuronal para detectar e identificar, de manera precisa y veloz, la desconexión de un generador en escenario de baja inercia. Se incluyen metodología, pruebas, resultados y conclusiones.

En el Capítulo 5 se describe la segunda parte del proyecto la cual tiene como finali-

dad el entrenamiento de una red neuronal para predecir la frecuencia mínima a presentarse en la red después de la desconexión de un generador, considerando un escenario de baja inercia. Se incluyen metodología, pruebas, resultados y conclusiones.

En el capítulo 6 se exponen las conclusiones originadas por el análisis general del proyecto de investigación y se hacen recomendaciones para futuros trabajos relacionados.

CAPÍTULO 2

ESTADO DEL ARTE

2.1 Introducción

La estabilidad del sistema eléctrico de potencia puede ser definida como la propiedad del sistema que le permite permanecer en un estado equilibrado de operación bajo condiciones normales y recuperar un estado de equilibrio aceptable después de haber sufrido un disturbio [14]. Una clasificación relevante dentro de la estabilidad del SEP es la llamada estabilidad transitoria, la cual es la capacidad del sistema eléctrico de mantener sincronismo entre todas sus unidades generadoras al sufrir un disturbio transitorio severo [15].

2.1.1 Inercia

La inercia es un parámetro fundamental para la estabilidad y el comportamiento de la red eléctrica. En el contexto de los SEP, inercia hace referencia a la energía cinética almacenada en los rotores que se encuentran en movimiento dentro de los generadores sincrónicos [16]. Un evento imprevisto, tal como la desconexión de un generador, provoca un desbalance entre generación y demanda. En el momento de dicho evento, la potencia disponible para entregar a la red disminuye abruptamente, sin embargo, la demanda por parte de los consumidores de electricidad no ha cambiado. Aquí es donde radica la importancia de la inercia para los sistemas eléctricos, ya que la energía cinética almacenada en las masas giratorias de los generadores convencionales es extraída para compensar temporalmente la pérdida de potencia, acción que presenta resistencia al cambio de

frecuencia en la red. Este proceso es conocido como respuesta inercial y provoca una reducción en la velocidad de giro de los rotores. Entre más baja sea la inercia en el sistema, menos amortiguamiento existirá para contrarrestar las desviaciones de frecuencia que los desbalances de potencia provocan [17].

La respuesta de los gobernadores de los generadores sincrónicos es la acción primaria de control de frecuencia, la cual usualmente tarda entre 10 y 30 segundos en suceder. La respuesta inercial actúa antes que los demás esquemas de control de frecuencia y da tiempo para que los sistemas mecánicos en la red actúen y ordenen la aceleración o desaceleración de las máquinas.

Cuando la frecuencia en una red eléctrica cae mas allá de cierto valor predefinido, una parte de la carga es desconectada para tratar de igualar la cantidad restante de carga con la cantidad restante de generación. A este proceso se le conoce como esquema de desconexión automática de carga y, como es lógico concluir, afecta a los usuarios del servicio de electricidad.

A medida que los generadores sincrónicos son desplazados por fuentes de energía renovable, la respuesta inercial en el sistema se reduce, lo que lleva a una mayor tasa de cambio de frecuencia (ROCOF, del inglés *rate of change of frequency*) y a un menor punto mínimo de frecuencia (nadir) [1]. Estos cambios pueden provocar activaciones de protecciones como la antes mencionada u otras que usan la frecuencia para la toma de decisiones. En la Figura 1 se observa la diferencia del comportamiento de la frecuencia en un sistema con alta y baja inercia.

La dinámica de cada generador sincrónico puede ser expresado por medio de la ecuación de oscilación:

$$P_m - P_e = \frac{d(\frac{1}{2}J\omega_e^2)}{dt} \quad (1)$$

donde P_m es la potencia mecánica, P_e es la potencia eléctrica, J representa el momento de

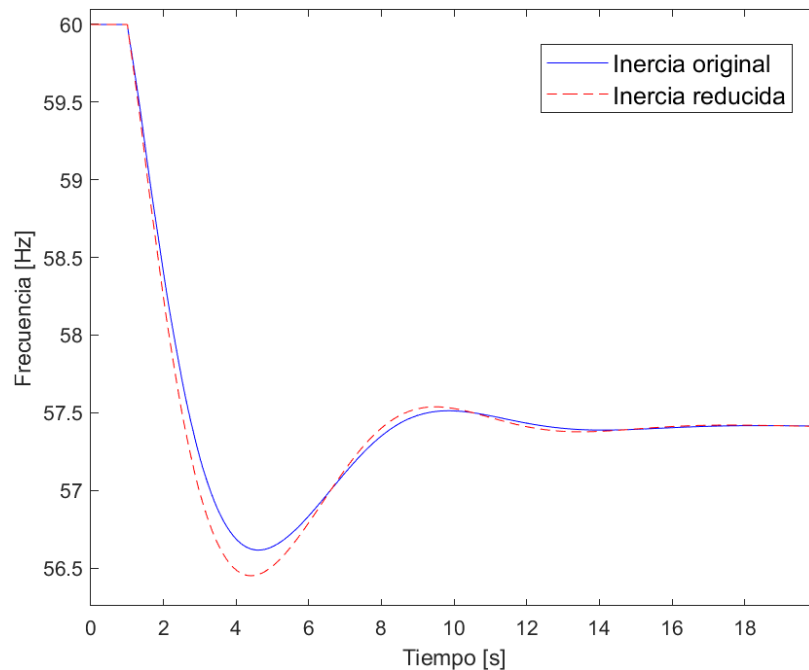


Figura 1. Comportamiento de la frecuencia con diferentes cantidades de inercia en el sistema.

inercia combinado de generador y turbina y ω_e es la frecuencia eléctrica angular. El lado derecho de (1) es la derivada de la energía cinética almacenada en turbina y generador. La constante de inercia H es la energía cinética normalizada a la potencia aparente S de cada unidad generadora y representa el tiempo en segundos que un generador puede entregar potencia nominal usando solo la energía cinética almacenada en su masa rotatoria.

$$H = \frac{\frac{1}{2}J\omega_{e,0}^2}{S} = \frac{J\omega_{e,0}^2}{2S} \quad (2)$$

siendo $\omega_{e,0}^2$ la frecuencia angular nominal del sistema. Sustituyendo (2) en (1) resulta en

$$\frac{2H}{\omega_{e,0}} \frac{d\omega_{e,0}}{dt} = \frac{P_e - P_m}{S} \quad (3)$$

Ya que la frecuencia del sistema es considerada como un parámetro global del SEP, todas las unidades generadores pueden ser combinadas en una sola unidad de la siguiente forma

$$\frac{2H_{tot}}{\omega_{e,0}} \frac{d\omega_{e,0}}{dt} = \frac{P_{gen} - P_{dem}}{S_{tot}} \quad (4)$$

donde H_{tot} es la constante de inercia de todo el sistema eléctrico, P_{gen} es la potencia total generada, P_{dem} es la potencia total demandada y S_{tot} es la capacidad total de generación conectada al sistema.

La ecuación (4) también puede ser presentada en términos de frecuencia f medida en Hz en lugar de la frecuencia angular medida en rad/s, de la siguiente manera

$$\frac{2H_{tot}}{f} \frac{df}{dt} = \frac{P_{gen} - P_{dem}}{S_{tot}} \quad (5)$$

donde $\frac{df}{dt}$ es el ROCOF del sistema. De esta forma se puede apreciar que el ROCOF aumenta a medida que H_{tot} disminuye.

2.2 Trabajo actual

El constante monitoreo y evaluación de la estabilidad del SEP es de suma importancia para asegurar una operación confiable de la red eléctrica. Las herramientas tradicionalmente usadas en la tarea de evaluación de la estabilidad son computacionalmente intensivas, por lo cual podrían no ser aptas para implementación en tiempo real, debido al gran volumen de datos y su velocidad de llegada. Es por esto que los investigadores han optado por las herramientas de aprendizaje automático (AA) para la evaluación en tiempo real en apoyo a la toma de decisiones [18].

En esta siguiente sección se presentarán los esfuerzos actuales por parte de investigadores para ofrecer soluciones de evaluación de estabilidad y parámetros relacionados a ella que podrían ayudar a las toma de decisiones de los operadores del sistema eléctrico.

En [19], los autores proponen un sistema de monitoreo en línea para estabilidad del sistema eléctrico basado en mediciones de PMU en todas las barras de generación. Dicho sistema de monitoreo usa las magnitudes y los ángulos del voltaje. Con estos se realizan mapas de calor, los cuales son una representación gráfica de datos donde cada valor en una matriz es representado con cierta intensidad de color. Los mapas de calor son las entradas que recibe una RNC, que hace una función de clasificación. De acuerdo a los autores, es la primera vez que se usan mapas de calor y técnicas de aprendizaje profundo para este problema. Los datos necesarios para entrenar la RNC fueron generados por medio de simulación detallada del sistema de potencia. Al transformar los datos de las simulaciones en mapas de calor se observan patrones que distinguen los casos estables de los inestables, lo que llevó a los autores a utilizar RNC, las cuales son actualmente la técnica más exitosa de AA para reconocimiento de patrones en imágenes. Este sistema de monitoreo en línea es puesto a prueba usando los sistemas IEEE de 118 y 145 barras. Se consigue un buen comportamiento el cual es mejor que los métodos más recientes basados en MSV. También se observa robustez ante el ruido y variaciones en el sistema.

Los autores en [20] proponen un modelo no lineal autorregresivo basado en una RNA para la predicción del valor mínimo (o nadir) de la frecuencia del sistema después de ocurrido un disturbio en un escenario de baja inercia. La serie de tiempo de la frecuencia es procesada por el modelo propuesto de predicción. Una fracción de la serie de tiempo antes del nadir de frecuencia es presentada al modelo, de esta manera, con un modelo de RNA ya entrenado, las muestras siguientes son estimadas por el modelo. Las entradas recibidas en la RNA son muestras con retardos de la serie de tiempo bajo análisis, es decir, no solo se recibe la muestra actual, sino que una ventana de datos anteriores.

En un intento de realizar un balance entre precisión y complejidad del modelo, se utilizaron diferentes longitudes de ventanas de entrada, las cuales varían de 2 a 10 muestras. Se consideraron 3 diferentes longitudes de entrenamiento: 5, 6.25 y 7.5 % de los

datos fueron utilizados durante el entrenamiento. El comportamiento de los modelos de RNA depende del número de neuronas en la capa oculta y, en este trabajo, se consideró entre 1 y 10 neuronas ocultas. Se concluyó que el escenario con la menor cantidad de datos de entrada (5 % de la serie de tiempo) durante el entrenamiento obtuvo el peor resultado de predicción con el modelo más simple (una neurona en capa oculta). El escenario donde se utiliza la mayor fracción de la serie de tiempo (7.5 %), obtuvo los mejores resultados de predicción. Se observa que el enfoque para predecir el nadir de frecuente utilizando una RNA es útil para proporcionar información relevante al operador del sistema eléctrico.

El trabajo [21] presenta una herramienta para la evaluación continua de la estabilidad transitoria. El problema es tratado como uno de clasificación con dos categorías. Se propone un novedoso algoritmo de minería de datos acompañado con una máquina de vectores de núcleo (MVN) para tratar con el problema, basándose en la información proveniente de PMUs.

Para realizar el entrenamiento de la MVN, se llevaron a cabo simulaciones fuera de línea bajo diferentes condiciones. Fueron seleccionadas 24 variables como entradas a la MVN para el entrenamiento y posterior implementación. Esta herramienta contempla hacer un nuevo entrenamiento dadas ciertas condiciones, como un cambio significativo de topología de la red. Se usa un índice de estabilidad transitoria para evaluar la estabilidad del sistema.

Se realizaron pruebas en el sistema IEEE New England de 39 barras y también en sistemas reales con el fin de evaluar la precisión y la velocidad de la herramienta. Observando los resultados se concluyó que se tiene un rendimiento óptimo para la evaluación continua de estabilidad transitoria debido a la alta precisión y velocidad alcanzada por el algoritmo propuesto.

En [18], Malbasa *et al.* presentan una estrategia de AA activo para el monitoreo de la estabilidad de voltaje en sistemas de transmisión. En este trabajo se propone una solución de aprendizaje activo para mejorar las aplicaciones actuales de AA por medio de una interacción entre la predicción en línea y el proceso de entrenamiento fuera de línea.

En lugar de depender de simulaciones exhaustivas en las que se trata de incluir todo tipo de condiciones, esta técnica identifica los puntos de operación más representativos al construir el conjunto de entrenamiento. También se buscan los puntos de operación en los que las predicciones son incorrectas. De esta forma, al crear un conjunto de entrenamiento alrededor de los puntos de operación identificados, es posible mejorar la capacidad de las herramientas de AA para pronosticar futuros estados del sistema de potencia.

El enfoque propuesto es puesto a prueba por medio de experimentos en los que utilizaron datos sintéticos obtenidos por medio de simulaciones detalladas. Se concluyó que el esquema propuesto puede mejorar los modelos de AA existentes al identificar puntos de operación donde las predicciones no se asemejan a la realidad y creado datos etiquetados alrededor de estos puntos. La herramienta de predicción que resultó mas exacta para el problema de estabilidad de voltaje fue bosques aleatorios (BA).

Los investigadores en [22] presentan una metodología basada en AA para la evaluación de la estabilidad transitoria en tiempo real usando como entrada mediciones proporcionadas por PMU con el fin de predecir el estado de estabilidad transitoria del SEP basado en el equivalente de una sola máquina del sistema. Para construir la base de datos a utilizar en el proceso de entrenamiento, se realizan simulaciones de tipo Monte Carlo evaluando las respuestas del sistema usando el método del equivalente de una sola máquina. Dichas simulaciones incluyen diferente tipos de eventos que podrían causar inestabilidad transitoria, como corto circuitos trifásicos seguidos de la desconexión de

la línea fallada. También se consideran diferentes escenarios de operación al variar las cargas.

Con esta base datos se entrena un clasificador inteligente basado en MSV para ser usado en evaluación de estabilidad en tiempo real. Además la MSV es sintonizada usando una variante del algoritmo de optimización de mapeo de media y varianza. Las pruebas realizadas demuestran la factibilidad y eficacia que se puede alcanzar en la estimación de la estabilidad transitoria con el esquema propuesto basado en clasificador de MSV.

Se puede ver que la estabilidad del sistema eléctrico es un tema que está en constante revisión y los esfuerzos por el desarrollo de herramientas útiles en toma de decisiones siguen existiendo y evolucionando a medida que se presentan nuevos algoritmos. También se puede identificar la falta de atención en la mayoría de los trabajos de situaciones en que la inercia del SEP se ve reducida, situación cada vez más común en las redes eléctricas modernas.

CAPÍTULO 3

APRENDIZAJE AUTOMÁTICO

3.1 Introducción

El aprendizaje automático (AA) o *machine learning*, es un subcampo de las ciencias de la computación el cual hace uso de la estadística y probabilidad. Dos definiciones populares del aprendizaje automático son:

1. “Campo de estudio que da a las computadoras la habilidad de aprender sin ser programadas explícitamente.” - Arthur Samuel, 1959.
2. “Se dice que un programa computacional aprende de la experiencia E con respecto a una tarea T y alguna medida de desempeño P, si su desempeño en T, medido por P, mejora con la experiencia E.” - Tom Mitchell, 1997.

El AA busca el desarrollo de técnicas que permitan a sistemas computacionales *aprender*. Se puede decir que los investigadores del aprendizaje automático buscan algoritmos que permitan utilizar muestras de datos para crear programas de computadora (también llamados modelos) con una función específica, sin tener que escribir dichos programas explícitamente. Es importante que los modelos resultantes cuenten con la capacidad de generalizar y realizar inferencias para un conjunto de datos más amplio. Un ejemplo útil para entender lo anterior, es el filtro de correos basura, que puede aprender a detectar este tipo de elementos al haber recibido ejemplos de correos útiles y basura.

Partiendo de la definición dada por Mitchell, es posible hacer un pequeño ejemplo.

Si queremos que una computadora aprenda a jugar ajedrez, la experiencia E sería jugar muchas partidas de ajedrez, la tarea T es jugar ajedrez y la medida de desempeño P es la habilidad para ganar partidas [23].

3.2 Aplicaciones

Existe una gran variedad de aplicaciones prácticas que se han beneficiado del AA en diversas áreas del conocimiento, por ejemplo:

- Conducción autónoma. Diferentes técnicas de AA se han explorado para permitir la conducción segura de vehículos sin la intervención de humanos en la toma de decisiones [24] [25].
- Procesamiento de voz. Trata el problema de conversión de secuencias de datos acústicos a una secuencia de palabras [26].
- Finanzas. Existe variedad de usos en este campo como la detección de fraudes, predicción del valor de acciones, evaluación de riesgo, retorno de inversión, entre otros [27].
- Análisis de imágenes. Esto incluye clasificación de imágenes, reconocimiento facial, detección de fallas en productos, diagnósticos médicos y una amplia variedad [28] [29].
- Clasificación de texto. Incluye problemas como detectar el tema de un texto o documento, creación de resúmenes o detección automática de contenido inapropiado [30].
- Existen muchos otros usos que se le da al AA como pronóstico de clima, recomendaciones personalizadas, publicidad dirigida, predicción de crecimiento poblacional, asistentes personales virtuales y una larga y creciente variedad [31] [32] [33].

De manera general, se puede afirmar que el aprendizaje automático es de gran utilidad para [34]:

- Problemas en los que las soluciones existentes consisten en una lista larga de reglas, un modelo de AA puede simplificar el código y tener un mejor rendimiento.
- Problemas complejos para los que no existe una solución satisfactoria usando los métodos tradicionales.
- Áreas cambiantes. Un enfoque de AA se puede adaptar continuamente a nueva información.
- Ganar entendimiento de problemas complejos y grandes cantidades de datos.

Existen tareas estándar del AA que han sido estudiadas ampliamente y cuentan con gran popularidad. Éstas son:

- Regresión: trata el problema de predecir un valor real para cada elemento. La predicción del valor de acciones u otras variables económicas son ejemplos de regresión.
- Clasificación: es el problema de asignar una categoría a cada elemento. Por ejemplo, se puede clasificar un documento en función del tema que trata como deportes, cine, política, economía.
- Agrupamiento: esta tarea tiene como objetivo separar un grupo de elementos en otros subgrupos. Los elementos de estos subgrupos cuentan características similares. Por ejemplo, hablando de redes sociales, algoritmos de agrupamiento pueden identificar comunidades dentro de grupos grandes de gente.
- Ordenamiento: este problema consiste en aprender a ordenar elementos de acuerdo con un criterio establecido. Por ejemplo, un servicio de transmisión de películas puede usar técnicas de AA para recomendar filmes basado en preferencias de cada usuario.
- Reducción de dimensionalidad: este proceso consiste en reducir el número de variables aleatorias para tener una representación de menor dimensión de la información mientras se conservan algunas propiedades de la representación original.

3.3 Tipos

Se pueden distinguir tres tipos principales de entrenamiento de modelos de aprendizaje automático, los cuales están definidos principalmente por el nivel de supervisión humana involucrada, estos son: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

3.3.1 Aprendizaje supervisado

En este tipo de aprendizaje se cuenta con datos etiquetados correctamente. El grupo de datos que serán usados para el proceso de entrenamiento se denomina conjunto de entrenamiento. Para cada variable de entrada $x^{(i)}$, también llamada *atributo*, se tiene una salida $y^{(i)}$, comúnmente llamada *objetivo*. De esta manera se entrenará al modelo de AA con entradas que tienen asociadas una respuesta correcta.

Como ejemplo, en Figura 2 se puede ver un clasificador de formas geométricas que, para cada figura que observa durante su entrenamiento, también recibe la etiqueta de esa figura. Con este procedimiento, el clasificador podrá relacionar los atributos de las formas geométricas (lados, números de esquinas, etc.) con la etiqueta correcta correspondiente,

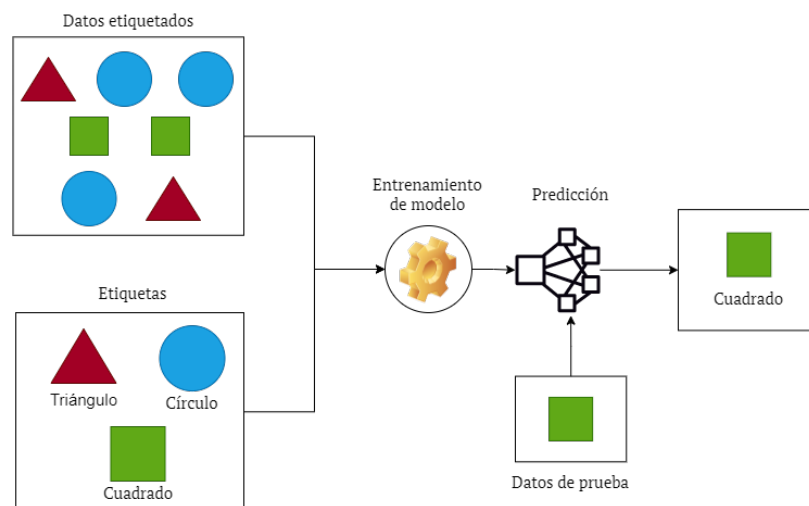


Figura 2. Ejemplo de clasificador de figuras geométricas.

para que al momento de recibir información no vista antes, la predicción de la categoría de esa información sea la acertada.

Otra tarea común es la de predecir un valor numérico, como el precio de una casa, dadas ciertas características como años de antigüedad, tamaño, número de habitaciones, etc. Para un entrenamiento exitoso, habría que dar al modelo muchos ejemplos de casas con características y su etiqueta correspondiente (precios).

3.3.2 Aprendizaje no supervisado

En esta situación, el modelo solo recibe información no etiquetada. Esto significa que no hay un objetivo conocido, únicamente se cuenta con variables de entrada. En el aprendizaje no supervisado se entrega información al modelo con el propósito de que se extraiga algún tipo de conocimiento de esa información. La tarea de agrupamiento es común en este tipo de aprendizaje. En Figura 3 se observa un ejemplo de agrupamiento: a la izquierda se encuentran los datos originales y a la derecha los mismos datos separados en grupos.

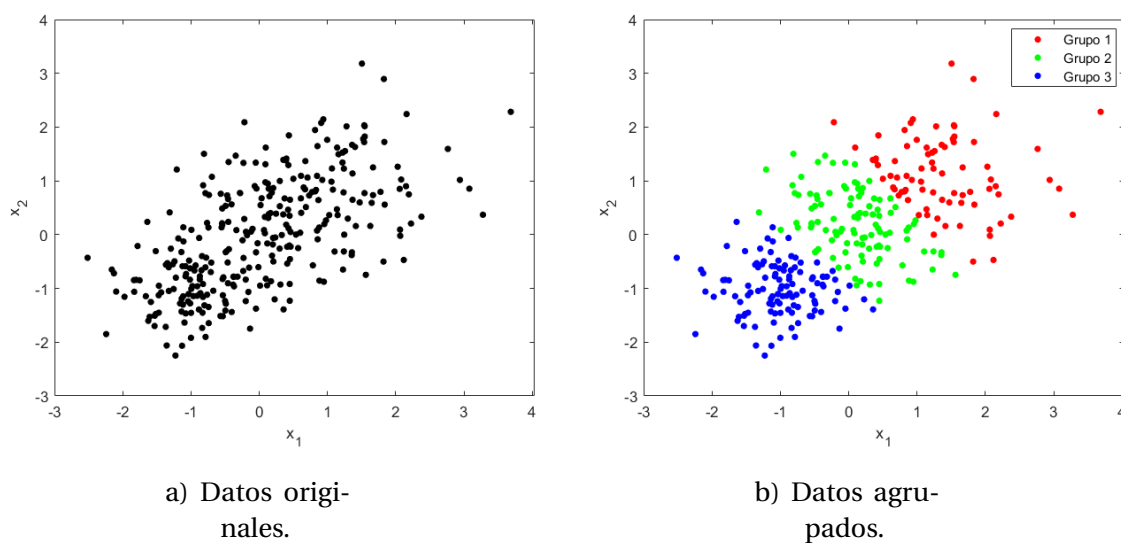


Figura 3. Ejemplo de agrupamiento.

3.3.3 Aprendizaje por refuerzo

En este tipo de aprendizaje se busca que un sistema, llamado *agente*, ubicado en un ambiente, aprenda a realizar una tarea por medio de prueba y error. Esto se lleva a cabo dando recompensas o sanciones al agente por sus decisiones y acciones. El agente debe aprender por su cuenta cuál es la mejor estrategia para obtener la mayor cantidad de recompensa. Los cuatro elementos esenciales del aprendizaje por refuerzo son:

- Agente. El programa que es entrenado para que realice una tarea específica.
- Ambiente. Esto es el entorno donde el agente realiza las acciones. Puede ser el mundo real o virtual.
- Acción. Movimiento hecho por el agente que causa un cambio en el estado del ambiente.
- Recompensa. Evaluación de la acción realizada. Puede ser negativa o positiva.

Figura 4 muestra la interacción entre agente y ambiente que da como resultado el aprendizaje de la tarea en cuestión.

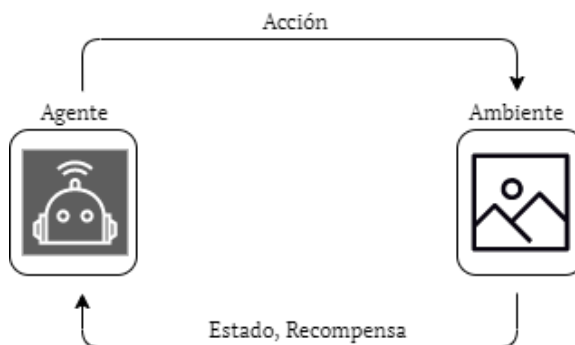


Figura 4. Interacción entre agente y ambiente.

3.4 Generalización, sobreajuste y subajuste

En el aprendizaje supervisado, se tiene como objetivo construir un modelo utilizando el conjunto de entrenamiento para después poder hacer predicciones precisas con datos nunca antes vistos (conjunto de prueba). Si el modelo es capaz de realizar predicciones acertadas con los datos nuevos, se dice que ha logrado *generalizar* de manera adecuada del conjunto de entrenamiento al conjunto de prueba. Se desea que los modelos puedan generalizar de la manera más precisa posible [35].

En la búsqueda de un modelo con una generalización adecuada a la tarea en mano, se debe evitar el *sobreajuste* y el *subajuste*. El sobreajuste sucede cuando se ajusta un modelo de forma muy cercana a las particularidades del conjunto de entrenamiento. Haciendo esto, se puede tener una muy buena medida de desempeño P en el conjunto de entrenamiento, pero un bajo desempeño en el conjunto de prueba, es decir, se pierde la capacidad de generalización. En este caso el modelo se ha vuelto demasiado complejo y puede no ser adecuado para su implementación.

Por otro lado, si el modelo es muy simple puede caer en el subajuste. Esta simpleza no permite que se entienda completamente la estructura y características de los datos para llevar a cabo predicciones satisfactorias.

Por lo tanto, se puede concluir que se busca un modelo que entienda las complejidades de los datos tratados, pero no que encaje perfectamente con cada detalle del conjunto de entrenamiento para que su generalización permita realizar predicciones sobre datos nuevos.

En Figura 5 se muestran diferentes niveles de ajuste de un modelo para el mismo conjunto de datos. Se busca que haya un balance entre precisión y generalización.

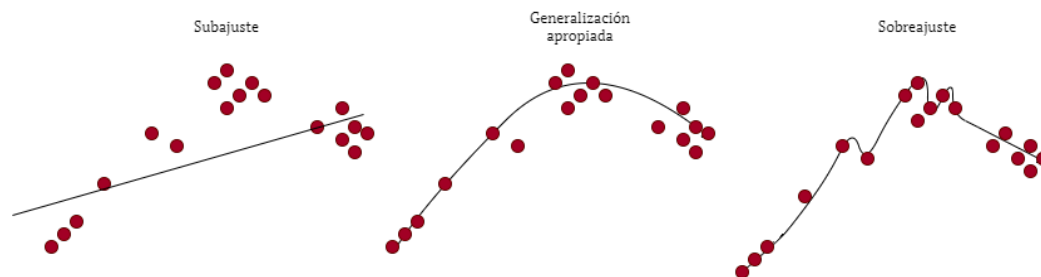


Figura 5. Diferentes niveles de ajuste de modelo..

3.5 Técnicas

Existen numerosas técnicas y algoritmos de aprendizaje automático supervisado, cada uno con ventajas y desventajas. La elección de cierta técnica dependerá de factores como los datos a utilizar y el tipo de tarea a realizar. A continuación se describen las ideas principales de algunas de las técnicas más populares actualmente.

3.5.1 K vecinos más cercanos

El algoritmo k vecinos más cercanos (KNN, del inglés k -nearest neighbors) es uno de los algoritmos más simples de aprendizaje automático. Puede utilizarse para clasificar nuevas muestras o para predicción de valores continuos (regresión). Este método se basa en observar las muestras más cercanas para clasificar el dato nuevo. Así, asigna la categoría del nuevo punto en base a la categoría de la mayoría de las muestras que le rodean.

KNN calcula la distancia entre el dato a clasificar y el resto de datos del conjunto, se seleccionan k elementos más cercanos y se elige la clasificación que tiene la mayoría de estos "vecinos".

En Figura 6 se muestra un ejemplo de clasificación de una nueva muestra usando el algoritmo KNN. Como se puede ver, en caso que se elija $k = 3$ la nueva muestra sería clasificada como triángulo. En caso $k = 5$, el dato se clasificaría como estrella. Queda claro así que la elección de la cantidad de vecinos a utilizar en esta técnica es un parámetro muy

importante que puede cambiar el comportamiento del modelo drásticamente.

3.5.2 Regresión lineal

Este es un modelo lineal usado para aproximar la relación entre una variable dependiente y , una o más variables independientes x (atributos) y un término aleatorio ϵ .

En (6) se muestra la fórmula general de predicción:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n + \epsilon \quad (6)$$

donde \hat{y} es la predicción que el modelo hace, $\theta_0, \theta_1, \theta_2, \dots, \theta_n$ son los parámetros del modelo también llamados pesos, x es un atributo, n es el número de atributos independientes y ϵ corresponde al término de residuos y engloba un conjunto grande de factores. De esta manera se modifican los pesos para ajustar el modelo de predicción.

El modelo de regresión lineal encuentra los parámetros θ y ϵ que minimizan el error cuadrático medio (ECM) entre las predicciones y los valores correctos conocidos,

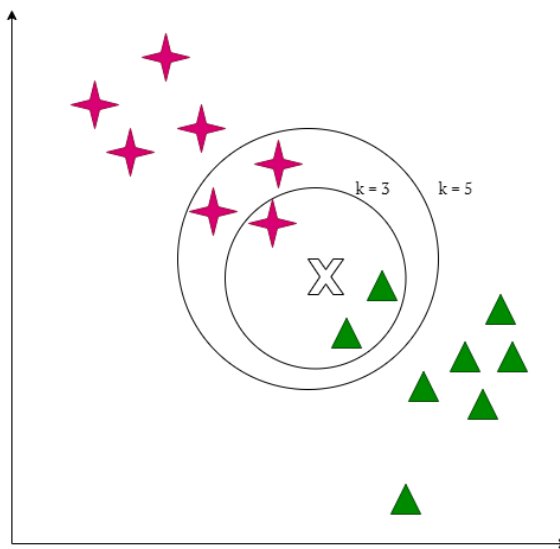


Figura 6. Ejemplo de clasificación con KNN.

para el conjunto de entrenamiento. En (7) se encuentra la fórmula del ECM

$$ECM = \frac{1}{s} \sum_{i=1}^s (y_i - \hat{y}_i)^2 \quad (7)$$

donde s es el número de muestras en el conjunto, y es el valor objetivo correcto y \hat{y} es el valor predicho.

En Figura 7 se muestra un ejemplo ilustrativo del ajuste de modelo de regresión lineal a un conjunto de datos.

3.5.3 Redes neuronales artificiales

Las redes neuronales artificiales (RNA) nacieron inspiradas en el funcionamiento del cerebro biológico. Las RNA son versátiles, poderosas y cuentan con gran capacidad lo cual las hace ideales para resolver tareas de AA de gran complejidad [34].

Para entender las RNA comenzaremos por analizar el perceptrón. El perceptrón es una de las arquitecturas mas simples de RNA, inventado por Frank Rosenblatt en 1957.

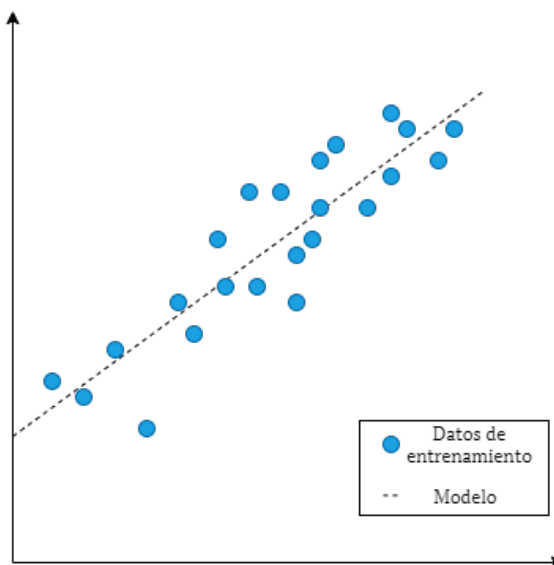


Figura 7. Ejemplo de regresión lineal con una variable dependiente y una independiente .

Cuenta con entradas y una salida. Asociada a cada entrada x , existe un peso w que regula la influencia de cada una de las entradas. Para obtener la salida \hat{y} , primero se realiza la suma ponderada de las entradas de la siguiente manera:

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n = \mathbf{w}^T \cdot \mathbf{x} \quad (8)$$

Después se verifica si z es mayor que el valor límite de la neurona.

$$\hat{y} = \begin{cases} 1 & \text{si } z > \text{valor límite} \\ 0 & \text{si } z \leq \text{valor límite} \end{cases} \quad (9)$$

Moviendo el valor límite al otro lado de la desigualdad y reemplazándolo por lo que se conoce como sesgo de la neurona (representado con b , del inglés *bias*):

$$\hat{y} = \begin{cases} 1 & \text{si } z + b > 0 \\ 0 & \text{si } z + b \leq 0 \end{cases} \quad (10)$$

Así, se puede observar que $b = -\text{valor límite}$. La acción de una neurona depende de su función de activación, que es una función que transforma los valores o define las condiciones para la salida de la neurona. Una función escalón muy utilizada en perceptrones es la función Heaviside o escalón unitario. Heaviside es una función discontinua cuyo valor es 0 para cualquier argumento negativo y 1 para cualquier argumento positivo, incluyendo el cero. Otras funciones de activación usadas ampliamente son *sigmoid*, *tanh*, *ReLU* y *softmax*.

En Figura 8 se muestra el esquema típico de un perceptrón.

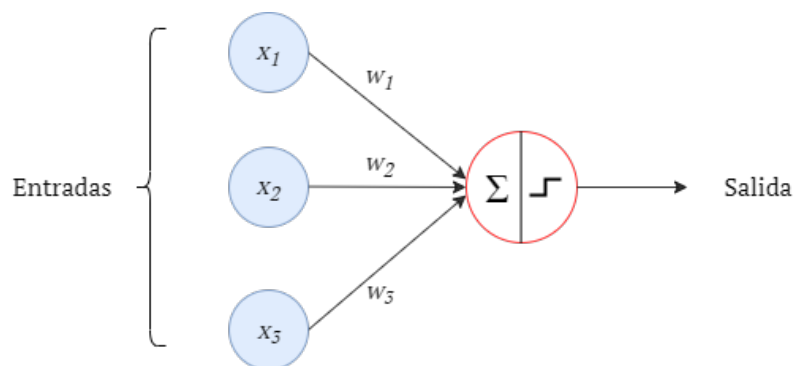


Figura 8. Esquema de un perceptrón.

El perceptrón es útil para dar una introducción sencilla a las redes neuronales, pero no es muy utilizado en la actualidad debido a sus limitaciones [36].

Las principales limitaciones de los perceptrones pueden eliminarse al combinar varios de ellos. La arquitectura de RNA resultante es conocida como perceptrón multicapa (MLP, del inglés *Multilayer Perceptron*). Un MLP se compone de una capa de entrada, una o más capas ocultas y una capa de salida, como se ilustra en Figura 9. En cada capa, con excepción de la capa de salida, se incluye una neurona de sesgo la cual siempre presenta un valor de 1 y está totalmente conectada a la siguiente capa.

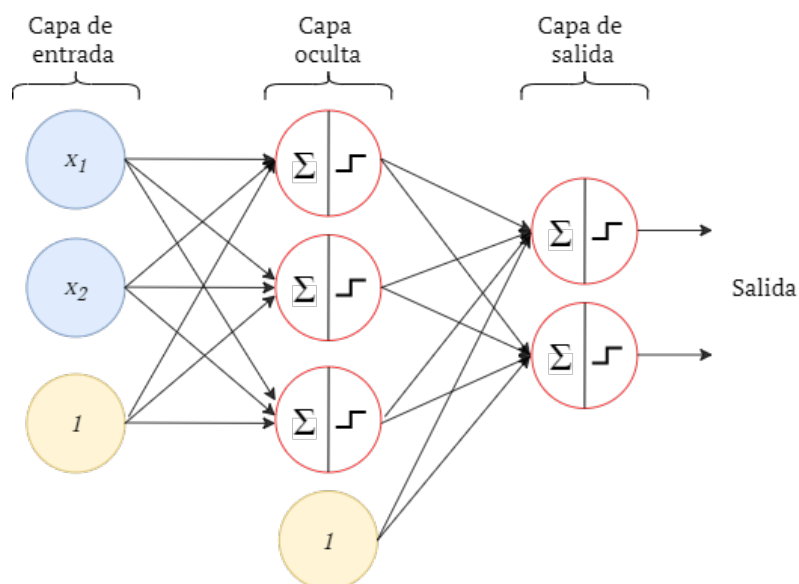


Figura 9. Perceptrón multicapa.

3.5.4 Máquinas de vectores de soporte

Las máquinas de vectores de soporte (SVM, del inglés *Support Vector Machines*), son un conjunto de algoritmos parte del aprendizaje supervisado, desarrollados por Vapnik y su equipo en la década de 1990 para clasificación binaria [37].

La idea básica detrás de SVM, es buscar el hiperplano óptimo para separar los dos conjuntos de datos, maximizando el margen entre los puntos más cercanos de las dos clases (estos puntos son llamados vectores de soporte).

Como se puede ver en la parte izquierda de la Figura 10, para el conjunto de datos linealmente separable, existen muchas diferentes formas de separar los puntos de manera satisfactoria, pero el hecho de que el hiperplano se acerque demasiado a cualquiera de los conjuntos puede llevar a problemas con la generalización. En cambio, en la parte derecha la misma figura, se observa el límite de decisión de un clasificador SVM, donde la línea separa las dos clases de datos y se encuentra lo más lejos posible de los puntos más cercanos. El hecho de maximizar el margen, proporciona una mayor capacidad de generalización y resultados correctos de clasificación.

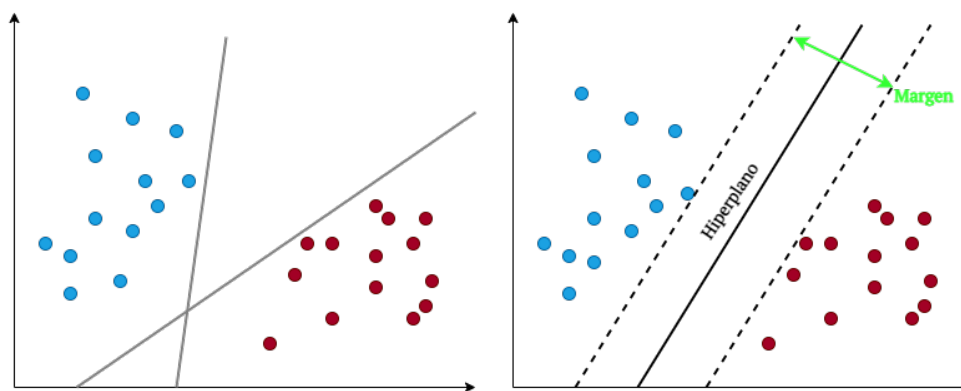


Figura 10. Separación de datos sin SVM y con SVM.

CAPÍTULO 4

DETECCIÓN E IDENTIFICACIÓN DE DESCONEXIÓN DE UNIDAD GENERADORA EN CONDICIONES DE BAJA INERCIA USANDO REDES NEURONALES

4.1 Introducción

En las últimas décadas, los sistemas eléctricos de potencia han enfrentado cambios y actualizaciones para ajustarse a la integración de energías renovables y sistemas de almacenamiento, mercados eléctricos, regulaciones, tecnologías de medición y comunicación, entre otros. Las tecnologías basadas en convertidores electrónicos de potencia están desplazando cada vez más a los generadores síncronos (GS) en los SEP modernos, lo que provoca la reducción de la inercia total del sistema. La inercia rotacional es fundamental para el sistema de potencia ya que le permite al GS un aumento instantáneo de la inyección de potencia activa cuando existe un cambio en el equilibrio entre generación y demanda, ayudando a amortiguar la variación de frecuencia. Por lo tanto, los procesos dinámicos son cada vez más rápidos y extremos en los sistemas de potencia de baja inercia.

Esta transición trae consigo nuevos desafíos para el funcionamiento, la estabilidad y la seguridad de la red [38]. Una operación estable y segura del SEP es siempre una prioridad para los gobiernos y las empresas de servicios públicos debido a la importancia de la red eléctrica en los aspectos sociales, políticos y económicos. Según el Instituto de Investigación de Energía Eléctrica (EPRI), la pérdida económica en Estados Unidos

causada por cortes de suministro eléctrico es de \$104 mil millones a \$164 mil millones de dólares estadounidenses al año [39]. Estas razones hacen que el seguimiento constante del estado del SEP sea una tarea fundamental y un tema en constante investigación.

Las unidades de generación son un componente vital del SEP y la desconexión repentina de una de estas unidades tiene un impacto sumamente relevante en la estabilidad del sistema, rompiendo el equilibrio entre carga y generación y, dadas ciertas condiciones, tiene el potencial de generar un apagón.

Varios apagones relacionados con la estabilidad de frecuencia han demostrado la importancia de niveles de inercia apropiados y un control de frecuencia adecuado. Por ejemplo, en Italia, el 28 de septiembre de 2003, el SEP italiano experimentó un corte de energía eléctrica el cual afectó aproximadamente a 60 millones de personas y alcanzó un punto mínimo de frecuencia de 47.5 Hz, lo que provocó la desconexión de todos los generadores del país [40].

En Australia en 2016 ocurrió un apagón, el cual es el primer incidente de este tipo reportado que se atribuye a una alta penetración de fuentes de energía renovable. El 28 de septiembre de 2016, una fuerte tormenta golpeó el estado de Australia del Sur y dañó varias torres de transmisión, provocando que la red del estado perdiera alrededor del 52 % de su generación eólica en unos cuantos minutos.

Para compensar la pérdida de generación se comenzó a importar potencia por medio de una interconexión con un estado vecino, pero la capacidad del enlace se vio rebasada, lo que terminó por provocar el aislamiento de la red del estado del resto del sistema y su colapso poco después [41]. Alrededor de 1.7 millones de personas se vieron afectadas por este evento y se estima que las pérdidas económicas rondan los 367 millones de dólares australianos [42].

Uno de los casos más recientes fue el sucedido el 16 de junio de 2019 en Argentina

y algunos países vecinos, evento que afectó cerca de 50 millones de personas y causó grandes pérdidas monetarias [43].

El análisis de casos de grandes cortes de energía eléctrica como los presentados deja ver que para minimizar el área afectada, es de gran importancia actuar lo más rápido posible. Entre más veloces sean las acciones del operador del sistema, existe más probabilidad de evitar la propagación del fenómeno.

Saber si un generador se desconectó de SEP y específicamente qué unidad generadora ya no se encuentra aportando potencia al sistema ayudaría al operador de la red a tomar decisiones para mantener el SEP funcionando a su mejor capacidad.

Recientemente, se han utilizado numerosas técnicas de aprendizaje automático para modelar y monitorear aplicaciones complejas. Herramientas como las redes neuronales artificiales (RNA), el árbol de decisión (DT), el análisis de componentes principales (PCA), las máquinas de vectores de soporte (SVM), las redes neuronales convolucionales (CNN), entre otras, se han propuesto en estudios que involucran la seguridad y estabilidad del sistema de energía.

En la literatura, las implementaciones de técnicas de aprendizaje automático han demostrado tener la capacidad de aprender y comprender las características peculiares de los sistemas eléctricos de potencia modernos [3]. Las RNA son una de las técnicas de aprendizaje automático más utilizadas en las últimas décadas. Las RNA son sistemas de computación masivamente paralelos que constan de un gran número de procesadores simples con muchas interconexiones [44].

Las RNA aprenden la relaciones complejas entre las entradas y salidas para hacer predicciones precisas y han demostrado ser una herramienta útil en el análisis de sistemas de potencia. Por ejemplo, en [45] los autores propusieron una metodología de localización de fallas basada en RNA para un sistema de transmisión de alto voltaje de corriente

continua basado en convertidor de fuente de voltaje. En [46], se presenta un controlador regulador automático de voltaje basado en RNA para el sistema de voltaje de excitación de la máquina síncrona, donde los resultados mostraron una rápida eliminación de la oscilación que condujo a una mejora de la estabilidad transitoria del SEP. En [47] se presenta un enfoque basado en RNA para el monitoreo en línea de un margen de estabilidad de voltaje en sistemas de energía eléctrica.

Teniendo en contexto la importancia de una precisa y oportuna detección de la desconexión de un generador en sistema eléctrico con baja inercia y el reciente desarrollo e implementación del aprendizaje automático, se propone entrenar una RNA de tipo perceptrón multicapa para lograr detectar e identificar de manera acertada y veloz la desconexión de algún generador del SEP.

4.2 Metodología

En toda tarea de aprendizaje automático lo más importante son los datos con que se va a trabajar. Buenos resultados dependen en gran medida de contar con datos adecuados. Las características que deben tener los datos varían dependiendo del objetivo que se tiene. Por esta razón se comenzará con la descripción de los datos utilizados en el proceso de entrenamiento de la RNA MLP.

4.2.1 Origen de los datos

Los datos sintéticos fueron creados utilizando el programa computacional de análisis de sistemas de potencia DIgSILENT PowerFactory. Las simulaciones en el dominio del tiempo en un sistema de prueba produjeron la serie de tiempo ($\mathbf{X}_r = \delta$) usada para propósitos de entrenamiento y prueba. El sistema de prueba es el conocido sistema de potencia de 39 barras y 10 máquinas de Nueva Inglaterra [48] (figura 11).

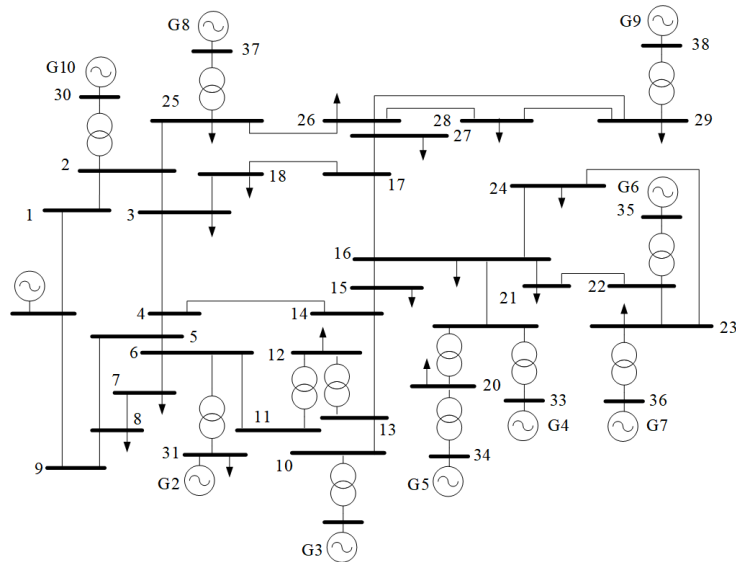


Figura 11. Diagrama unifilar sistema New England.

Los datos sintéticos se han creado considerando la desconexión repentina de un generador (contingencia única, diez casos), y luego se generan subcasos para considerar la inercia reducida. El escenario 1 considera la reducción de la inercia rotacional en todos los generadores; el k -ésimo subcaso se refiere a la reducción de la inercia total del sistema ($H_{gi} = (1 - k \times 100^{-1}) \times H_{0gi}$, donde H_{0gi} es la constante de inercia inicial del i -ésimo generador y $k = 0, 1, \dots, 10$, $i = 1, 2, \dots, 10$). El escenario 2 considera la inercia reducida localizada; el i -ésimo subcaso se refiere a la reducción de la inercia solo en el i -ésimo generador ($H_{gi} = (1 - k \times 100^{-1}) \times H_{0gi}$). El escenario 1 requirió 110 simulaciones y el escenario 2 1100 simulaciones, se produjeron 44.6 GB y 89.2 GB de datos para los escenarios 1 y 2, respectivamente. La simulación RMS cubre las variables electromecánicas del sistema de prueba durante una simulación en el dominio del tiempo de 240 segundos (paso de tiempo constante igual a 0.01 segundos). Los datos sintéticos totales se crearon en aproximadamente 15 horas (PC, Windows 10 professional, Intel i7 -8850H 2.60 GHz, 32 GB RAM).

4.2.2 Organización de los datos

Los datos creados en la sección anterior produjeron series de tiempo de varias variables electromecánicas; sin embargo, se seleccionaron los ángulos de voltaje de barra para formar la matriz de características ($\mathbf{X}_r = \delta$). El fundamento de esta decisión es que la dinámica electromecánica que provoca la desconexión del generador en un sistema de potencia multimáquina se refleja en los ángulos de tensión de barra más rápidamente que en la frecuencia en las redes inductivas, como es el caso de las redes eléctricas de alta tensión, según fue observado en los resultados de las simulaciones. Además, desde el punto de vista práctico, este proyecto de investigación analiza la implementación real del enfoque propuesto, y la disponibilidad de flujos de datos de PMU ofrece la información relacionada con los ángulos de voltaje, pero también el estimador del estado del sistema de potencia podría ayudar con los datos en el caso de la disponibilidad del flujo de datos de una PMU.

La corriente por unidad de cada generador, procedente de los datos sintéticos, se utiliza para identificar si las máquinas están conectadas o no a la red inmediatamente después de la perturbación. Esta información se utiliza para etiquetar el conjunto de datos que será usado durante el proceso de entrenamiento y para formar la matriz de objetivos (\mathbf{D}_r). A cada generador se le asigna una variable de estado binaria para indicar el estado de la conexión en cada paso de tiempo, $D_{ri} = 1$ si el i -ésimo generador está desconectado del sistema eléctrico y $D_{ri} = 0$ si está conectado. Por ejemplo, si en el tiempo t , el generador G7 se desconecta del sistema de prueba y el resto de las máquinas están conectadas, el vector de objetivos en ese instante sería $\mathbf{D}_r = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]^T$. Si el generador G3 está desconectado y el resto sigue aportando al sistema, el vector de objetivos en ese instante sería $\mathbf{D}_r = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]^T$.

El vector de características \mathbf{X}_r contiene los ángulos de voltaje de las barras en cada

instante. Organizando todos los vectores de características y objetivos en matrices, \mathbf{X}_r y \mathbf{D}_r resulta en:

$$\mathbf{X}_r = \begin{bmatrix} \delta_1^1 & \delta_1^2 & \cdots & \delta_1^s \\ \delta_2^1 & \delta_2^2 & \cdots & \delta_2^s \\ \vdots & \vdots & \ddots & \vdots \\ \delta_n^1 & \delta_n^2 & \cdots & \delta_n^s \end{bmatrix}_{s \times n} \quad (11)$$

$$\mathbf{D}_r = \begin{bmatrix} D_1^1 & D_1^2 & \cdots & D_1^s \\ D_2^1 & D_2^2 & \cdots & D_2^s \\ \vdots & \vdots & \ddots & \vdots \\ D_{n_g}^1 & D_{n_g}^2 & \cdots & D_{n_g}^s \end{bmatrix}_{s \times n_g} \quad (12)$$

donde r es el número de simulación de la que se extrae la información, n es el número de barras del sistema eléctrico, s es el número de muestras por simulación seleccionadas para el entrenamiento y n_g es el número de generadores en el sistema.

Como se mostró antes, ambas matrices comparten el mismo número de muestras s (columnas) debido a que a cada vector de características le corresponde un vector de salidas deseado (vector de objetivos).

En este trabajo, se seleccionaron 989 muestras de características y objetivos de cada simulación. Este número de muestras representa 9.86 s de cada simulación, donde se puede observar el comportamiento antes, durante y después de la perturbación. La cantidad de muestras seleccionadas fue el resultado de una búsqueda de rendimiento deseado y velocidad de entrenamiento. En consecuencia, se crearon 1210 matrices de características y objetivos. Se concatenaron todas las matrices de características, así como todas las matrices de objetivos. El resultado es una matriz de características \mathbf{X} con dimensiones de $39 \times 1,196,690$ y una matriz de objetivos \mathbf{D} con dimensiones de $10 \times$

1,196,690, teniendo ambas matrices información de todas las simulaciones RMS.

4.2.3 Creación de la RNA

Una RNA se define por el número de capas, el número de neuronas en cada capa, los enlaces de conexiones y los pesos. Los datos ingresan a la ANN a través de la capa de entrada, pasan por la capa oculta y la salida se presenta en la capa de salida. En este trabajo de investigación, se emplea una red neuronal prealimentada (*feed-forward* en inglés) con estructura perceptrón multicapa. Ninguna de las salidas de las neuronas en las redes neuronales prealimentadas se usa como entrada para las neuronas de la misma capa o capas anteriores. La señal de entrada solo se propagará hacia adelante, desde las neuronas de entrada a las neuronas de salida. Dado que si se utiliza una RNA MLP, se necesitan dos capas ocultas para una generalidad completa, se decidió utilizar dos capas ocultas para este trabajo [49].

El número de neuronas ocultas se determinó utilizando la regla de la pirámide geométrica sugerida en [50], que ha mostrado un buen comportamiento. La *regla de la pirámide geométrica* es una aproximación del número ideal de neuronas ocultas y para dos capas ocultas muestra que:

$$NHN_1 = Mr^2 \quad (13)$$

$$NHN_2 = Mr \quad (14)$$

$$r = \sqrt[3]{\frac{N}{M}} \quad (15)$$

donde N es el número de entradas, M es el número de salidas, NHN_1 es el número de neuronas en la capa oculta 1 y NHN_2 es el número de neuronas en la capa oculta 2. Estas operaciones dan como resultado $NHN_1 = 24$ y $NHN_2 = 15$, ya que están considerando 39 entradas y 10 salidas. En la Figura 12 se muestra un esquema de la arquitectura resultante de la RNA.

4.2.4 Entrenamiento

Las RNA MLP generalmente se entrenan utilizando el algoritmo de retropropagación de errores (BP, del inglés *backpropagation*). BP es un método basado en el descenso de gradientes para minimizar el error de la salida calculada por la RNA.

BP puede verse como un proceso de dos etapas: (i) etapa hacia adelante y (ii) etapa hacia atrás. En la primera etapa, se aplica un vector de entrada a los nodos de entrada de la RNA y su efecto se propaga a través de la red capa por capa. La primera etapa produce un conjunto de salidas como la respuesta real de la red. Durante la etapa de avance, los pesos de las conexiones son todos fijos. En la segunda etapa, todos los pesos se ajustan siguiendo una regla de corrección de errores: la respuesta real de la RNA se resta de la respuesta deseada (objetivo) para producir una señal de error. Esta señal de error se propaga hacia

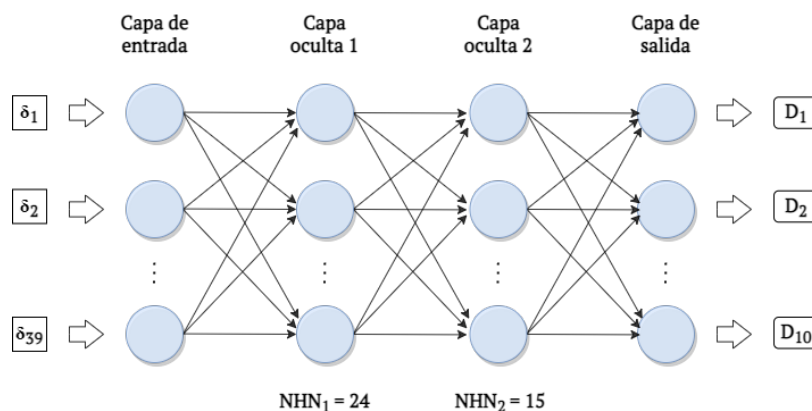


Figura 12. Estructura de RNA MLP para detección de desconexión de generador.

atrás a través de la red. Este proceso se repite capa a capa hasta completar la red neuronal artificial. Los pesos de cada neurona se reajustan para hacer que la respuesta real de la RNA se acerque más a la respuesta deseada, es decir, minimizando el error.

El algoritmo de entrenamiento elegido es BP basado en la técnica de optimización *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) *cuasi-Newton*. Este método ha resuelto con éxito sistemas de ecuaciones no lineales [51]. Los autores del método cuasi-Newton BFGS sugieren que el método “*supera sustancialmente a los métodos de gradiente conjugado conocidos en una amplia clase de problemas*” [52]. En [53] se puede encontrar una discusión detallada del método cuasi-Newton BFGS.

La función de error entropía cruzada (EC), se seleccionó como la función de error a minimizar. El error de EC mide el rendimiento de un modelo de clasificación cuya salida es una probabilidad entre 0 y 1. El error de EC aumenta a medida que la probabilidad predicha se aleja de la clasificación real. Por ejemplo, predecir una probabilidad de clase de 0.13 cuando el valor real es 1 resultaría en un valor de error EC muy alto. Minimizar la entropía cruzada conduce a buenos clasificadores. Error de EC para cada par de elementos salida-objetivo se calcula como:

$$EC = -y \log(\hat{y}) \quad (16)$$

donde y es la salida deseada y \hat{y} es la salida real de la RNA.

En este trabajo de investigación, los datos sintéticos se dividen aleatoriamente en tres partes: conjunto de entrenamiento (70 %), conjunto de validación (15 %) y conjunto de prueba (15 %), con el fin de evitar el sobreajuste de la ANN. El MLP consta de dos capas ocultas, 24 y 15 neuronas ocultas en cada capa oculta.

El proceso de entrenamiento comienza dando las matrices X y D . La Tabla 1 muestra

los parámetros utilizados para el proceso de entrenamiento. Al alcanzarse alguno de estos valores, el entrenamiento se concluye. Época es un término usado en el aprendizaje automático para definir el número de veces que todos los datos del conjunto de entrenamiento han pasado a través del algoritmo de aprendizaje. En cada época, todos los datos de entrenamiento pasan por el modelo para que este aprenda de ellos. El gradiente, en términos matemáticos, es una derivada parcial con respecto a sus entradas. Un gradiente mide el cambio en los pesos del modelo con respecto al cambio en el error. También se puede pensar en el gradiente como en la pendiente de una función. Entre mas grande el gradiente, más pronunciada la pendiente y más rápido puede aprender el modelo. Dado el caso que la pendiente sea cero, el modelo deja de aprender.

4.3 Resultados

La red neuronal propuesta se implementó en MATLAB 2020a utilizando *Deep Learning Toolbox*. El preprocesamiento de los datos, el entrenamiento y las pruebas se realizaron con el procesador Intel Xeon W-3235, 12 núcleos, 24 subprocesos, 3,3 / 4,4 GHz, 64 GB de RAM, Windows 10 profesional y cuatro tarjetas gráficas NVIDIA® GeForce® RTX 2080Ti. El uso de cuatro tarjetas gráficas y el procesamiento en paralelo permitió una aceleración sustancial en el proceso de entrenamiento, el cual se completó en aproximadamente 319 segundos.

TABLA 1

Límites de los criterios de entrenamiento.

Nombre	Valor	Significado
epochs	1000	Número máximo de épocas para entrenar
EC	0	EC objetivo
min_grad	1×10^{-6}	Gradiente de rendimiento mínimo
val_checks	12	Número máximo de épocas de incremento continuo del error

El error de EC mínimo alcanzado fue de 2.40×10^{-4} ; esta es la media de los valores de EC individuales (como se muestra en (16)). En la Figura 13 se puede ver la evolución del entrenamiento de la RNA, y como el error se reduce a través de las épocas hasta encontrar el punto óptimo.

En Figura 14 se observa el comportamiento del gradiente durante el proceso de entrenamiento y también el punto en que el error deja de reducirse por lo que se concluye con el aprendizaje.

La precisión general de la RNA entrenada es del 89.8 %, considerando los tres conjuntos. Esto significa que del total de observaciones presentadas durante el entrenamiento, la RNA clasificó correctamente el 89.8 % de las observaciones y clasificó erróneamente el 10.2 %.

Por ejemplo, una clasificación errónea sería que la salida de la RNA MLP muestre que el generador G7 se desconectó cuando en realidad el generador G4 está fuera de línea o que muestre que el generador G3 está en línea cuando actualmente está fuera de

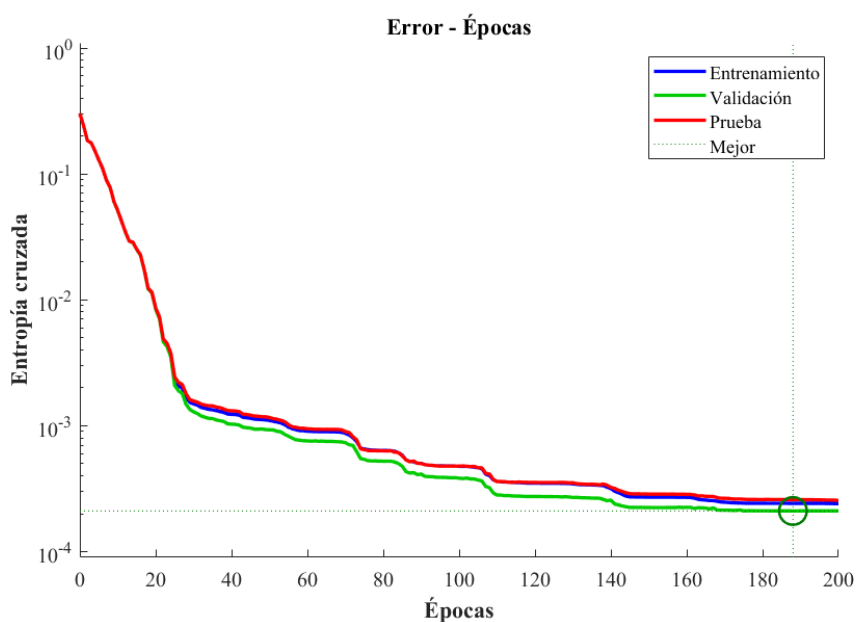


Figura 13. Evolución del error.

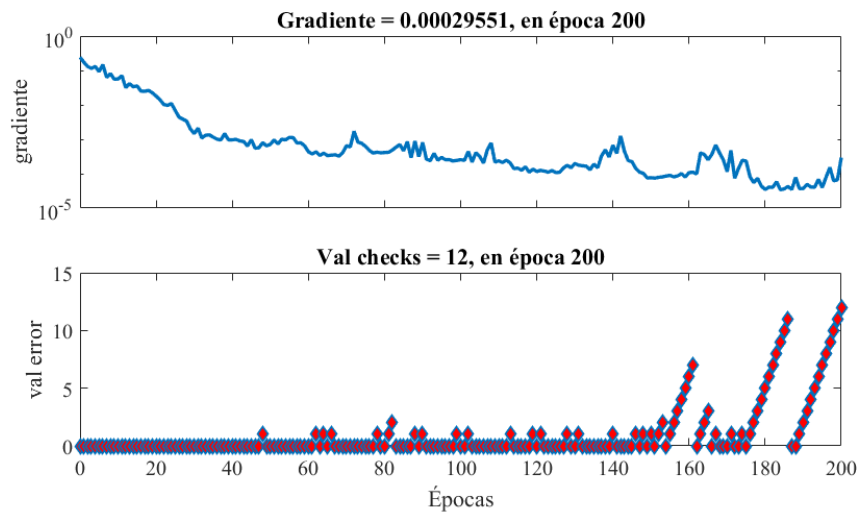


Figura 14. Gradiente y pruebas de validación durante el entrenamiento.

línea.

La Figura 15 muestra los resultados gráficos de un ejemplo ilustrativo de prueba de la RNA entrenada. La RNA recibió como entrada 15 segundos de los 39 ángulos de voltaje de las barras, datos que provienen de simulación RMS. Para este ejemplo específico, el caso de simulación considera la inercia rotacional inicial ($H_{tot} = 100\%$) de los generadores síncronos en el sistema. Luego, se presenta la interrupción del generador G1 (en $t = 1$ s).

La Figura 15 muestra el cambio del estado del generador G1 de $D_{r1} = 0$ (estado en línea) a $D_{r1} = 1$ (estado fuera de línea), mostrando que la RNA mantiene una alta confianza en que el generador G1 está desconectado durante el resto de la simulación. Los generadores restantes mantienen un estado muy cercano a $D_{r1} = 0$ ($i = 2, 3, \dots, 10$), lo que confirma la detección e identificación exitosa de la interrupción del generador.

La Figura 16 muestra el desempeño de la RNA entrenada en condiciones de baja

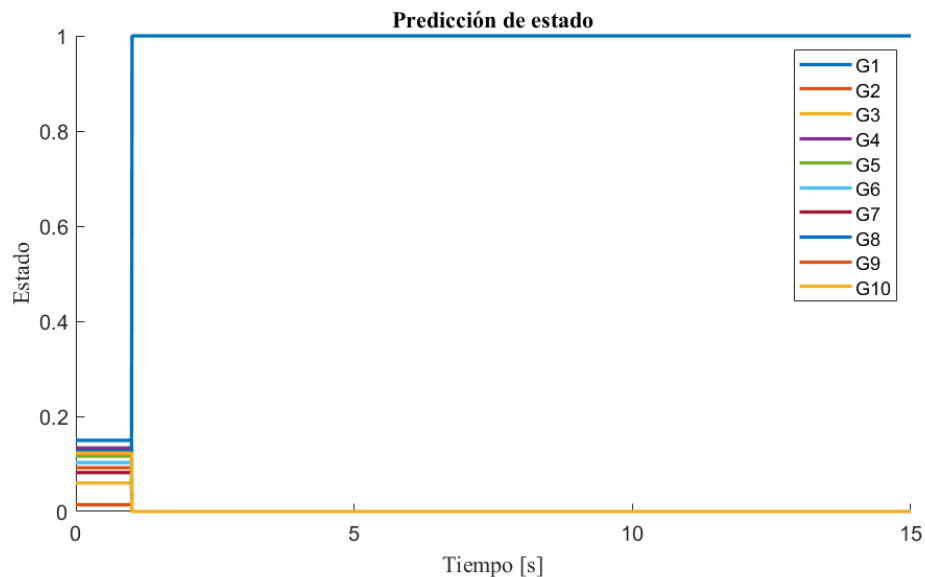


Figura 15. Ejemplo de salida de la RNA entrenada al detectar desconexión del generador 1.

inercia. La RNA recibió 15 segundos de datos de los 39 ángulos de voltaje de las barras del sistema en la capa de entrada. Los datos de entrada consideran una reducción del 10 % en la inercia de todas las máquinas ($H_{tot} = 90\%$). En la Figura 16 se puede observar que en el momento de la desconexión del generador ($t = 1.00$ s), el MLP detecta este evento instantáneamente e identifica correctamente qué máquina está ahora fuera de línea (en esta prueba, G4) modificando el estado de G4 de cero a uno. Mientras tanto, el resto de los generadores mantienen su estado en valor cero, lo que significa que la RNA identifica que los generadores aún están conectados a la red eléctrica.

En la Figura 17, se puede observar el desempeño del MLP en una situación de reducción de inercia. La inercia total del sistema se reduce quedando $H_{tot} = 95\%$ y en $t = 1$ s, el generador 9 se desconecta de la red eléctrica. Una vez más, la detección e identificación de la salida del generador de la red es instantánea y correcta, observándose que el estado de G09 cambia a 1, es decir, la máquina está separadas de la red.

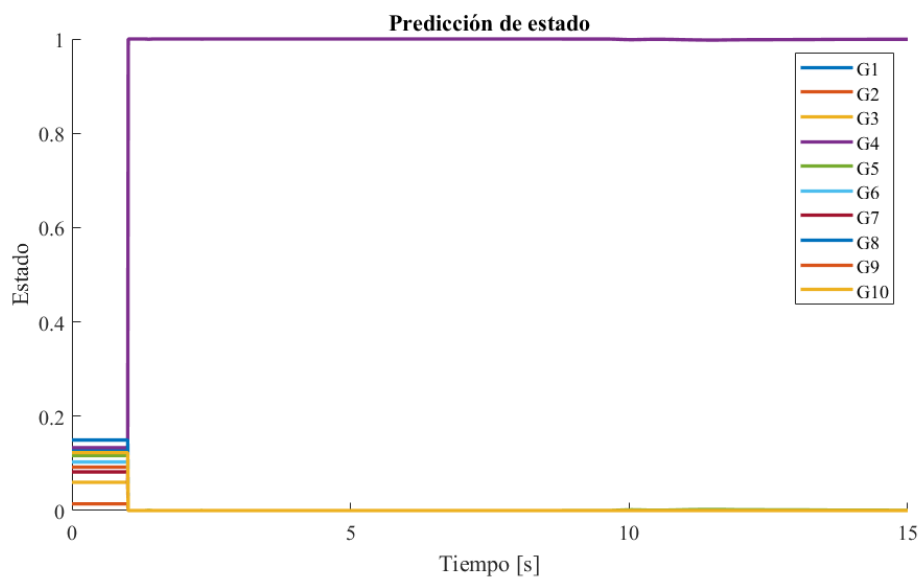


Figura 16. Ejemplo de salida de la RNA entrenado al detectar desconexión del generador 4, en caso de inercia reducida.

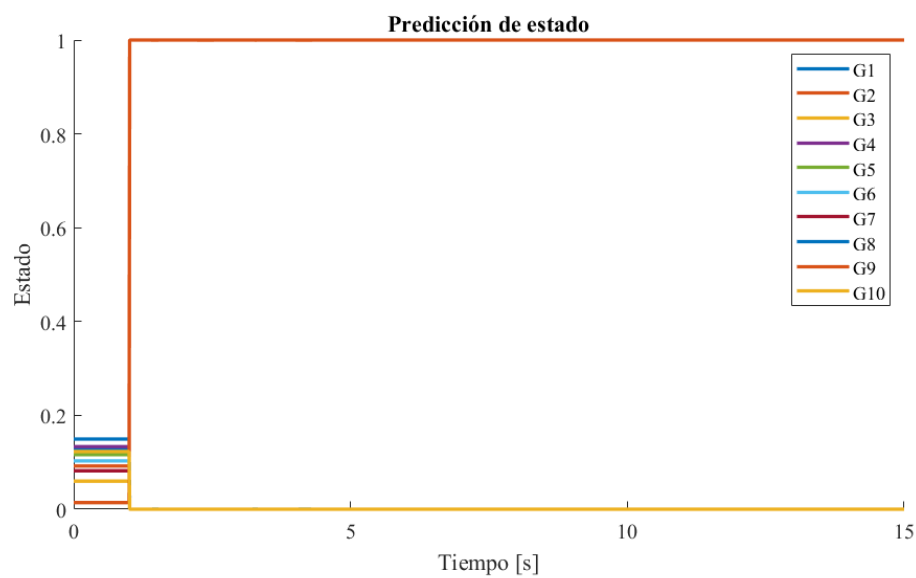


Figura 17. Ejemplo de salida de la RNA entrenado al detectar desconexión del generador 9, en caso de inercia reducida.

4.4 Conclusión

Las situaciones de reducción de inercia rotacional del sistema de potencia moderno son cada vez más comunes y están presionando al operador del sistema a tomar decisiones

en un marco de tiempo cada vez más reducido, lo que a la vez incentiva la búsqueda y desarrollo de nuevas herramientas de análisis. Las pruebas mostradas demuestran que, con un entrenamiento adecuado, una red neuronal de perceptrón multicapa puede identificar con éxito la desconexión de un generador de un sistema de potencia multimáquinas, considerando diferentes situaciones de nivel de inercia, recibiendo como entrada solo los ángulos de voltaje de las barras. Una herramienta de este tipo puede ayudar en el proceso de decisión de los operadores del sistema eléctrico de una manera muy rápida y precisa, en el caso de que se disponga de los datos correctos. Esto corrobora una vez más la capacidad de las redes neuronales de extraer patrones complejos propios de la naturaleza de los sistemas eléctricos.

Se analizaron casos con diferente inercia rotacional total en el sistema, observando la correcta y casi instantánea identificación de la desconexión de una unidad generadora de la red. Esto es especialmente importante ya que la inercia total en las redes eléctricas modernas cambia a lo largo del día dependiendo del despacho realizado y la disponibilidad de recursos naturales para las unidades generadoras renovables.

CAPÍTULO 5

PREDICCIÓN DE FRECUENCIA MÍNIMA DESPUÉS DE DESCONEXIÓN DE GENERADOR EN CONDICIONES DE BAJA INERCIA

5.1 Introducción

En la actualidad, es cada vez más común la utilización de inversores electrónicos de potencia en la red eléctrica. Esto se debe principalmente a la incorporación de fuentes de energía renovable y sistemas de almacenamiento relacionados, en respuesta a un movimiento que pretende reducir la quema de combustibles fósiles con fines de protección del medio ambiente y a una creciente demanda de energía eléctrica a nivel mundial [54].

Estos cambios se ven reflejados en una modificación de los SEP, que tradicionalmente se basaban en generadores síncronos (plantas hidroeléctricas, nucleares y de combustibles fósiles), a sistemas basados mayoritariamente en fuentes renovables y almacenamiento de energía (baterías), como se ilustra en Figura 18.

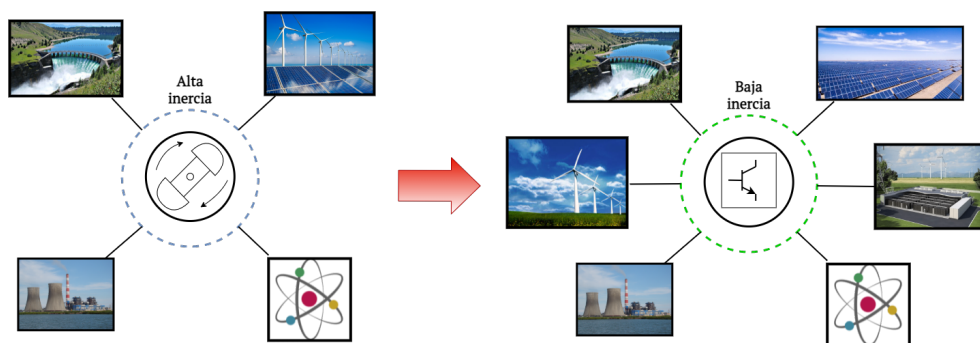


Figura 18. Evolución de los sistemas eléctricos de potencia.

Varios países han establecido metas ambiciosas para incorporar a su sistema eléctrico un porcentaje significativo de fuentes renovables, por ejemplo, China se ha puesto el objetivo de producir 35 % de la potencia eléctrica del país por medio de fuentes renovables para el año 2030 . Existen otros países que llevan la delantera en la penetración de energía renovable en su red, como Dinamarca (cerca de 53 %), Uruguay (28 %) y Alemania (26 %) [55].

La integración a gran escala de fuentes de energía renovables basadas en inversores electrónicos de potencia en la red eléctrica lleva a problemas de estabilidad de frecuencia [56]. Generalmente este tipo de fuentes tiene una baja o nula respuesta inercial. La generación fotovoltaica hace uso de inversores electrónicos de potencia para su integración a la red y no ofrecen respuesta inercial al sistema. Debido al uso de convertidores de frecuencia variable para la generación eólica, se desacopla de la red la inercia de la turbina. Por lo tanto, la inercia total del sistema eléctrico de potencia se reduce a medida que la penetración de energía renovable aumenta [57].

Una baja inercia en el sistema de potencia provoca una mayor tasa de cambio de frecuencia (ROCOF, del inglés *rate of change of frequency*) y a un menor punto mínimo de frecuencia (nadir) [58]. Desde el punto de vista de un operador del sistema eléctrico, se generan dos problemas principales. El primero es que un alto ROCOF lleva a la activación de protecciones y segundo, un nadir muy bajo puede resultar en desconexión de carga automática, lo cual afecta directamente a los usuarios y lleva a grandes consecuencias económicas.

Por los motivos expuestos, se planteó como objetivo crear una metodología para predecir el punto mínimo de frecuencia (f_{min}) en las barras de generación después de la desconexión de una unidad generadora en un sistema eléctrico de potencia con baja inercia.

Esta metodología se fundamenta principalmente en el uso de redes neuronales artificiales debido a su demostrada capacidad de formar complejas relaciones en diferentes áreas del conocimiento, incluidos los sistemas eléctricos de potencia.

5.2 Metodología

5.2.1 Origen de los datos

Los datos usados en el proceso de entrenamiento son los mismos que se describieron en el Capítulo 4. Para conveniencia del lector, se describen nuevamente.

Los datos fueron creados utilizando el programa computacional de análisis de sistemas de potencia DIGSILENT PowerFactory. El sistema de prueba es el conocido sistema de potencia de barra 39 y 10 máquinas de Nueva Inglaterra [48] (Figura 11).

Los datos sintéticos se han creado considerando la desconexión repentina de un generador (contingencia única, diez casos), y luego se generan subcasos para considerar la inercia reducida. El escenario 1 considera la reducción de la inercia rotacional en todos los generadores; el k -ésimo subcaso se refiere a la reducción de la inercia total del sistema ($H_{gi} = (1 - k \times 100^{-1}) \times H_{0gi}$, donde H_{0gi} es la constante de inercia inicial del i -ésimo generador y $k = 0, 1, \dots, 10$, $i = 1, 2, \dots, 10$).

El escenario 2 considera la inercia reducida localizada; el i -ésimo subcaso se refiere a la reducción de la inercia solo en el i -ésimo generador ($H_{gi} = (1 - k \times 100^{-1}) \times H_{0gi}$). El escenario 1 requirió 110 simulaciones y el escenario 2 1100 simulaciones, se produjeron 44.6 GB y 89.2 GB de datos para los escenarios 1 y 2, respectivamente. La simulación RMS cubre las variables electromecánicas del sistema de prueba durante una simulación en el dominio del tiempo de 240 segundos (paso de tiempo constante igual a 0.01 segundos). Los datos sintéticos totales se crearon en aproximadamente 15 horas (PC, Windows 10 professional, Intel i7 -8850H 2.60 GHz, 32 GB RAM).

5.2.2 Organización de los datos

Los datos creados en la sección anterior produjeron series de tiempo de varias variables electromecánicas; se seleccionaron los ángulos de voltaje de barra (δ) como las características, resulta en el vector:

$$\delta = \left[\delta_1 \quad \delta_2 \quad \cdots \quad \delta_n \right] \quad (17)$$

donde δ_i es el ángulo de voltaje de la i -ésima barra y n es el número de barras en la red. La decisión de utilizar los ángulos de tensión del barra se debe a que la dinámica electromecánica que provoca la desconexión de un generador en un sistema de potencia de varias máquinas se refleja en los ángulos de tensión de las barras más rápidamente que en la frecuencia en redes eléctricas inductivas, como en las de alta tensión.

El vector de objetivos se crea extrayendo el valor mínimo de cada serie de tiempo de frecuencia de barras generadores:

$$f = \left[f_1 \quad f_2 \quad \cdots \quad f_{n_g} \right] \quad (18)$$

donde f_i es la frecuencia mínima presente en la serie de tiempo de frecuencia en la barra del generador i -ésimo y n_g es el número de generadores en el sistema.

Solo se utilizaron datos del Escenario 1 para el proceso de entrenamiento y validación. De cada simulación del escenario 1 (110 simulaciones) se seleccionó el vector de ángulos desde el segundo 1 (momento del disturbio) al segundo 30.9816 (3000 vectores),

y se organizaron en una matriz de características \mathbf{X}_r :

$$\mathbf{X}_r = \begin{bmatrix} \delta_1^1 & \delta_1^2 & \cdots & \delta_1^s \\ \delta_2^1 & \delta_2^2 & \cdots & \delta_2^s \\ \vdots & \vdots & \ddots & \vdots \\ \delta_n^1 & \delta_n^2 & \cdots & \delta_n^s \end{bmatrix}_{s \times n} \quad (19)$$

donde s es el número de muestras por simulación seleccionadas para el entrenamiento.

Para crear una matriz de objetivos por simulación, el vector \mathbf{f} se replicó s veces, lo que resultó en una matriz con la misma información en cada columna:

$$\mathbf{f}_r = \begin{bmatrix} f_1^1 & f_1^2 & \cdots & f_1^s \\ f_2^1 & f_2^2 & \cdots & f_2^s \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_g}^1 & f_{n_g}^2 & \cdots & f_{n_g}^s \end{bmatrix}_{s \times n_g} \quad (20)$$

Entonces, por simulación, tenemos una matriz \mathbf{X}_r con dimensiones de 39×3000 y una matriz \mathbf{f}_r con dimensiones de 10×3000 .

Se crearon 110 matrices de características y objetivos. Todas las matrices de características se concatenaron juntas y todas las matrices de objetivos se concatenaron juntas. El resultado es una matriz de características \mathbf{X} con dimensiones de 39×330000 y una matriz de objetivos \mathbf{F} con dimensiones de 10×330000 , teniendo ambas matrices información de todas las simulaciones RMS del escenario 1.

Para aproximar problemas de comunicación reales, se creó una segunda matriz de características, donde se introdujo ruido gaussiano blanco aditivo (AWGN, del inglés *additive white Gaussian noise*) en la matriz de características, para que la RNA a entrenar

pueda aprender de datos ruidosos. AWGN es un modelo de ruido básico utilizado en la teoría de la información para imitar el efecto de muchos procesos aleatorios que ocurren en la naturaleza. En [59], los autores caracterizan y cuantifican el ruido en los datos registrados por PMU a 345 kV, 20 kV y 120 V. Concluyen que una relación señal/ruido (SNR) de 45 dB podría ser una buena aproximación para todos los datos de PMU. Por este motivo, el ruido incluido en la matriz de características fue de 45 dB.

5.2.3 Creación de la RNA

Una red neuronal de retardo de tiempo (TDNN, del inglés *time-delay neural network*) puede denominarse como red neuronal de retroalimentación, excepto que tiene una línea de retardo en la entrada. Una TDNN puede considerarse como una arquitectura ANN cuyo objetivo principal es trabajar con datos secuenciales.

En este tipo de redes neuronales, los retardos de tiempo d se aplican al vector de entrada y se presentan como entradas a la red. Esto significa que las neuronas de entrada reciben una ventana de valores pasados y la relacionan con las salidas deseadas. Estas entradas con retardo de tiempo proporcionan al TDNN información temporal sobre el sistema que se está examinando [60].

Para este trabajo $d = 10$ y la ventana va de 1 a d , es decir, no se considera la entrada

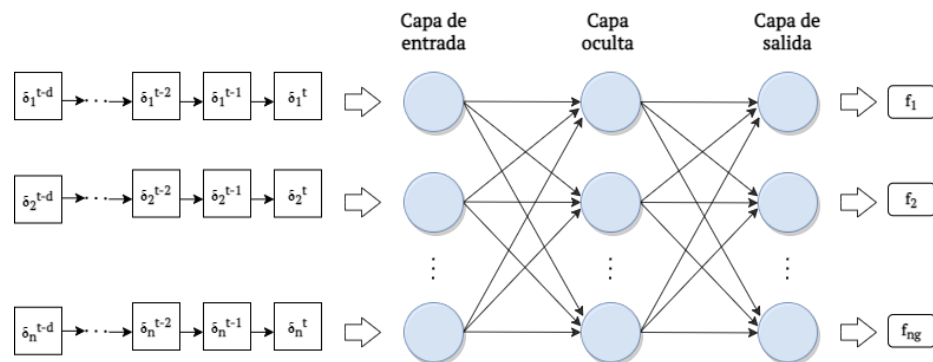


Figura 19. Esquema de TDNN con ventana de t a $t-d$.

instantánea actual en ese mismo momento, la ventana de memoria va de $t-1$ a $t-10$. La Figura 20 muestra esta configuración TDNN.

La capa de entrada tiene $n = 39$ neuronas, la capa oculta tiene diez neuronas y la capa de salida tiene $n_g = 10$ neuronas. Las neuronas de la capa oculta utilizan la función de transferencia sigmoidea tangente hiperbólica (*tansig*), y las neuronas de la capa de salida utilizan una función de transferencia lineal (*purelin*).

5.2.4 Entrenamiento

El algoritmo elegido para realizar el entrenamiento fue la retropropagación de Levenberg-Marquardt (LM), que es un algoritmo muy eficiente (en comparación con un algoritmo de gradiente conjugado y un algoritmo de tasa de aprendizaje variable) [61], y a menudo el algoritmo de retropropagación más rápido disponible en MATLAB. LM se recomienda como un algoritmo supervisado de primera elección.

El error cuadrático medio (ECM) es la función de rendimiento utilizada para evaluar el TDNN entrenado. ECM mide el promedio de los cuadrados del error entre los valores estimados y los valores reales. El ECM se calcula como:

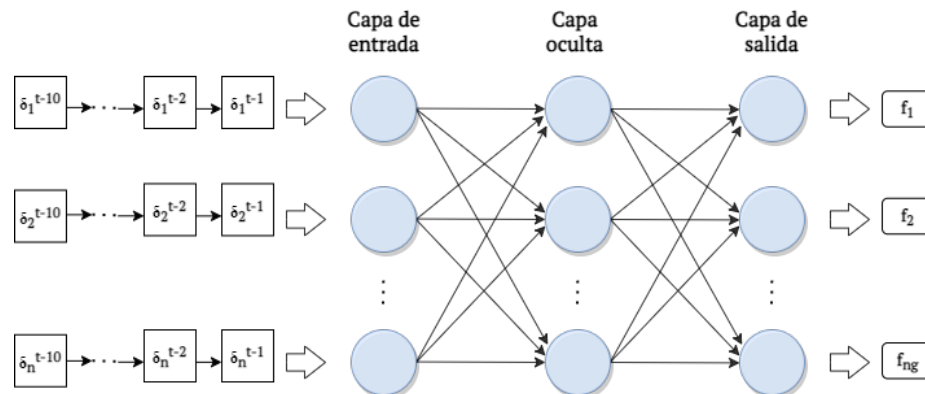


Figura 20. TDNN con ventana de $t-1$ a $t-10$.

$$ECM = \frac{1}{s} \sum_{i=1}^s (Y_i - \hat{Y}_i)^2 \quad (21)$$

donde s es el número de muestras, Y es el valor correcto y \hat{Y} es el valor predicho por el modelo.

Los datos sintéticos se dividen aleatoriamente en tres partes: conjunto de entrenamiento (70 %), conjunto de validación (15 %) y conjunto de prueba (15 %), para evitar el sobreajuste del TDNN. La Tabla 2 muestra los parámetros utilizados para el proceso de entrenamiento, junto con el significado de cada uno.

5.3 Resultados

La TDNN propuesta se implementó en MATLAB 2020a utilizando Deep Learning Toolbox y el preprocesamiento, el entrenamiento y las pruebas de datos se realizan con el procesador Intel Xeon W-3235, 12 núcleos, 24 subprocesos, 3,3 / 4,4 GHz, 64 GB de RAM, Windows 10 profesional y cuatro tarjetas gráficas NVIDIA® GeForce® RTX 2080Ti.

El entrenamiento tomó aproximadamente 3 horas con 20 minutos para el entrenamiento sin ruido y 5 horas con 57 minutos para el entrenamiento con ruido agregado. Se utilizó computación paralela con doce trabajadores para ambos entrenamientos. El entrenamiento se detuvo porque se alcanzó el criterio *val_checks* en ambos casos. En la

TABLA 2

Límites de los criterios de entrenamiento.

Nombre	Valor	Significado
epochs	1000	Número máximo de épocas para entrenar
mse	0	ECM objetivo
min_grad	1×10^{-6}	Gradiente de rendimiento mínimo
val_checks	6	Número máximo de épocas de incremento continuo del error

Figura 21 se muestra el desarrollo del ECM a través de las épocas para los conjuntos de entrenamiento, validación y prueba. Se observa como el error se reduce a medida que transcurre el entrenamiento y se toma el mejor rendimiento de la época con menor error de validación.

En Figura 22 se ve el desarrollo del gradiente, error de validación y μ durante las épocas. μ es un parámetro para controlar el algoritmo usado en entrenamiento de RNA, el cual decrece después de cada paso satisfactorio (reducción de error) y es incrementado solo cuando un paso tentativo incrementaría el rendimiento.

5.3.1 TDNN entrenada sin ruido

Como prueba para la TDNN entrenada sin ruido, esta recibió la serie temporal de ángulos de barra de una simulación perteneciente al escenario 1. En esta simulación hay una reducción de la inercia total del sistema en 10 % ($H_{tot} = 90\%$) y el generador 1 es desconectado al segundo 1.

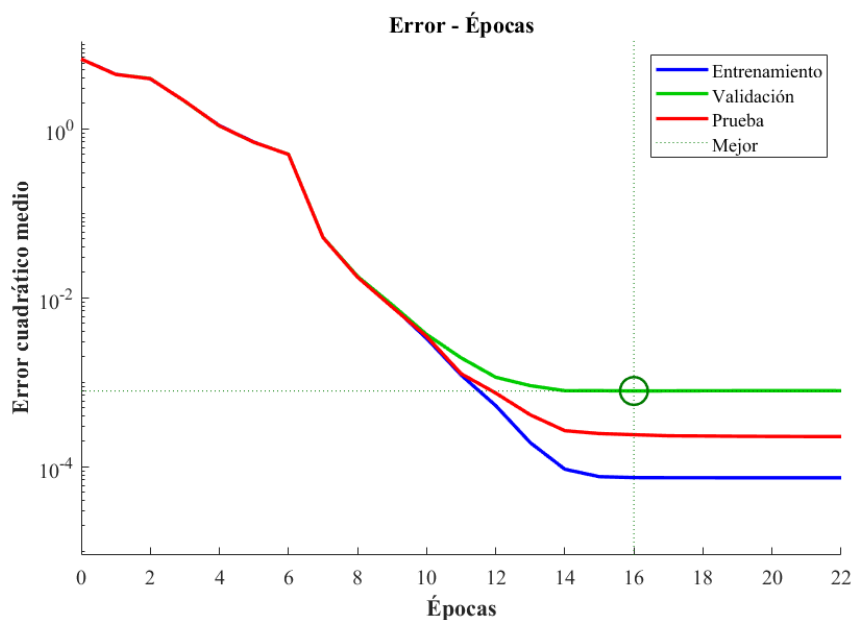


Figura 21. Comportamiento del ECM en entrenamiento de TDNN sin ruido.

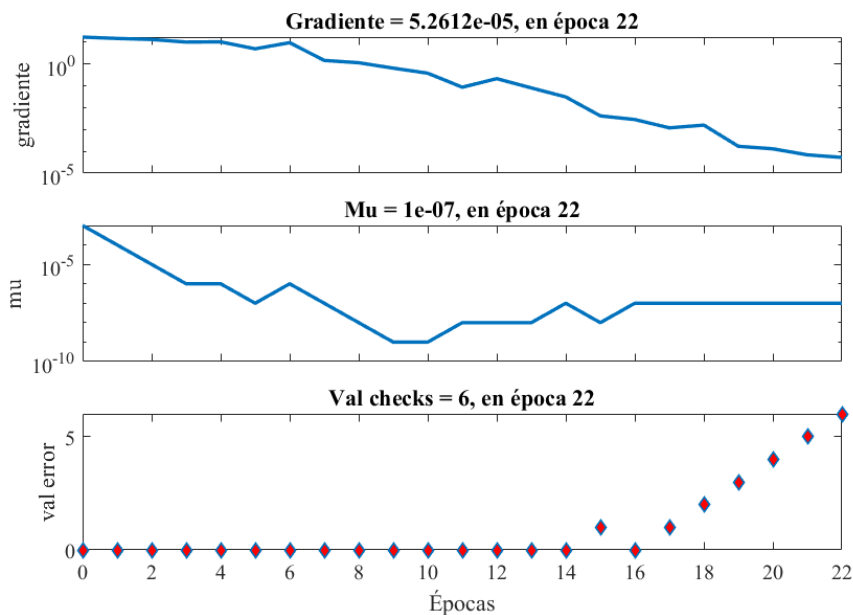


Figura 22. Gradiente, mu y pruebas de validación durante el entrenamiento.

La Figura 23 muestra el rendimiento de la TDNN, donde la línea azul es la frecuencia de la barra del generador 1 y la línea de puntos roja es la frecuencia mínima predicha. Debido a la ventana de tiempo que requiere la TDNN antes de hacer su primera predicción, es necesario esperar 10 muestras de tiempo, o sea, 0.1 segundos. Después de esta espera, se realiza una predicción cada muestra de tiempo hasta llegar al momento del valor mínimo real.

Se puede ver que la predicción es bastante precisa y se realiza mucho tiempo antes de que se alcance la frecuencia mínima. El valor mínimo real es de 55.923401 Hz y el primer valor predicho es de 55.923432 Hz. La TDNN también hizo predicciones para las otras nueve barras de generación.

Dado que analizando la serie de tiempo de la frecuencia podemos conocer el valor de frecuencia mínima real de las diez barras, podemos calcular el rendimiento de la TDNN en este ejemplo. El ECM en este caso fue 1.4800×10^{-6} .

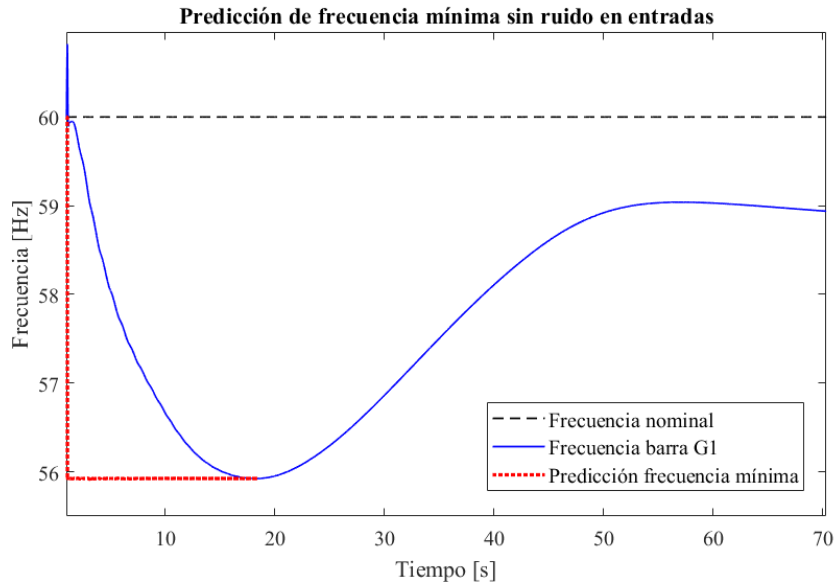


Figura 23. Predicción de frecuencia mínima con TDNN entrenada sin ruido recibiendo entradas sin ruido.

Utilizando la misma simulación del ejemplo anterior, se introdujo ruido gaussiano blanco aditivo (AWGN) en la señal de entrada. La SNR utilizada fue de 45, especificada en dB. En Figura 24 se muestra la señal del ángulo de la barra 1 con y sin ruido AWGN de 45 dB. Figura 25 presenta el comportamiento del modelo TDNN ante dicha señal ruidosa.

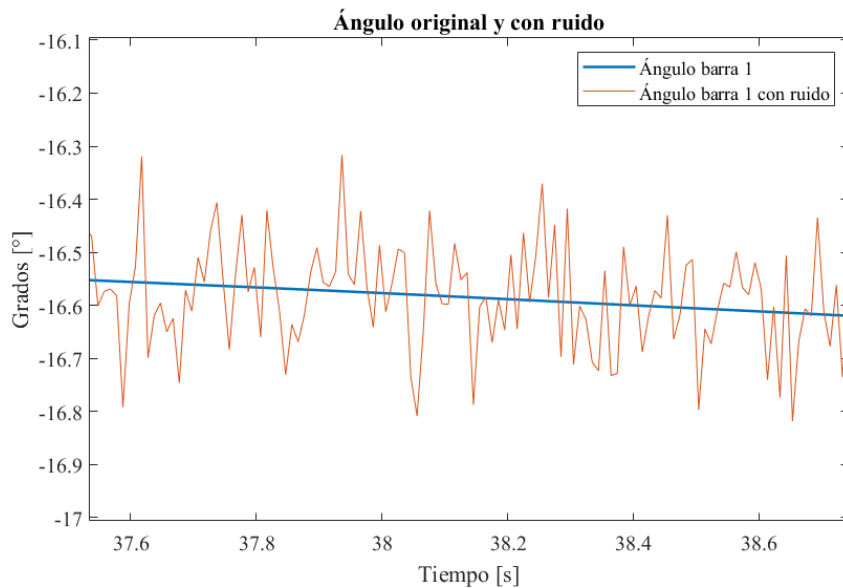


Figura 24. Ángulo original y con ruido añadido.

55.923401 y 55.929381 Hz son los valores mínimos reales y predichos, respectivamente. El ECM en este caso fue de 2.694×10^{-4} .

Comparando las figuras 23 y 25 y los resultados de ECM, es notorio que el rendimiento de la TDNN entrenada sin ruido decrece al recibir información que contiene ruido, una situación que puede presentarse en implementación real.

5.3.2 TDNN entrenada con ruido

Como se mencionó anteriormente, se creó una matriz de características con ruido añadido, la cuál se utilizó en este entrenamiento. Los parámetros de entrenamiento son los mismos de la Tabla 2. En Figura 26 se observa la reducción del ECM durante el proceso de aprendizaje, hasta encontrar el mejor resultado para el conjunto de validación en la época 34.

También en Figura 27 se ve el desarrollo del gradiente, error de validación y μ durante las épocas, hasta el momento en que se presentaron 6 épocas continuas con

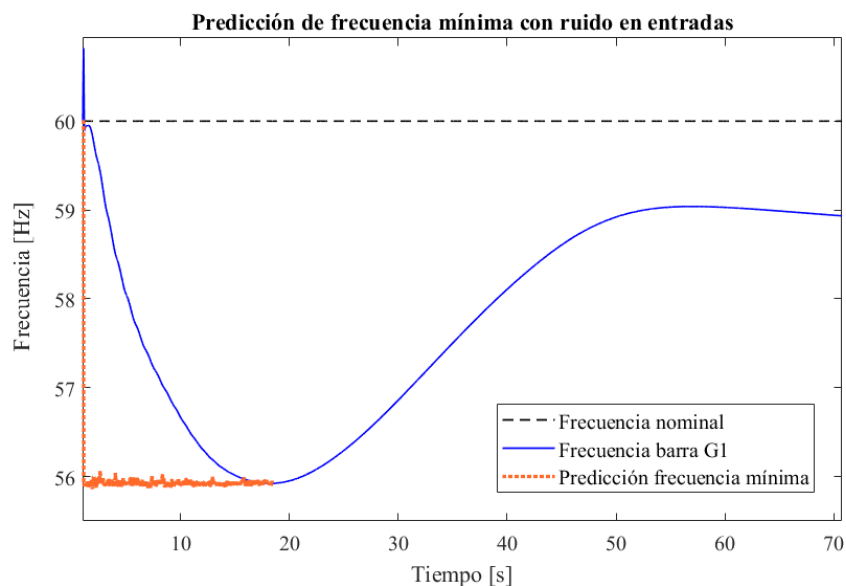


Figura 25. Predicción de frecuencia mínima con TDNN entrenada sin ruido recibiendo entradas con ruido.

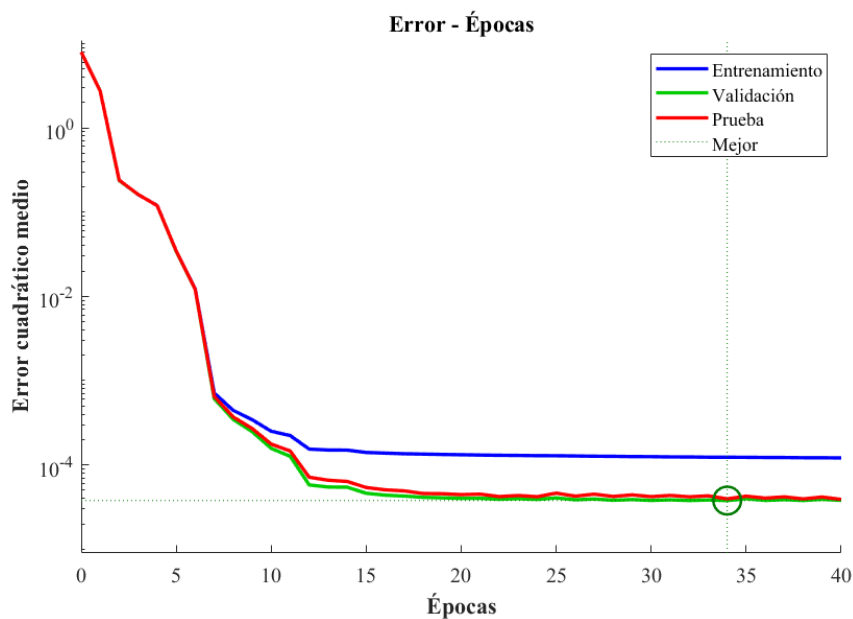


Figura 26. Comportamiento del ECM en entrenamiento de TDNN con ruido.

reducción de rendimiento.

En la Tabla 3, se puede encontrar la información sobre el ECM en conjunto de

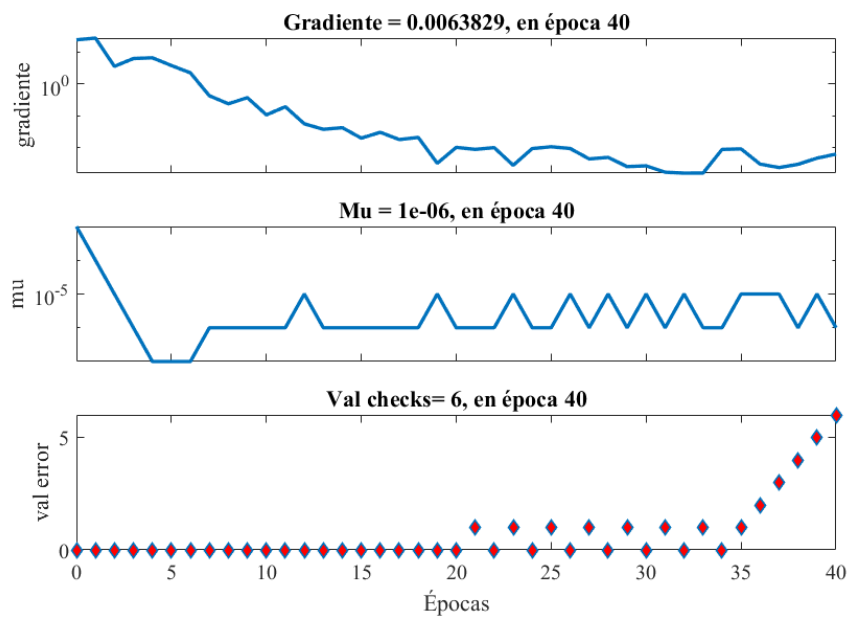


Figura 27. Gradiente, mu y pruebas de validación durante el entrenamiento con ruido.

entrenamiento, validación, prueba y general (los tres conjuntos) para ambos entrenamientos.

Se realizaron las mismas pruebas para la TDNN entrenada con ruido. La Figura 28 y la Figura 29 muestran el desempeño de la TDNN entrenada con ruido cuando recibió datos de entrada sin y con ruido, respectivamente. El ECM cuando se recibieron datos limpios fue 4.4017×10^{-6} . Cuando se presentaron datos ruidosos como entrada, el ECM fue 8.8988×10^{-6} . Se puede apreciar que ésta TDNN se comporta con más precisión a la hora de contar con ruido en las señales de entrada.

TABLA 3

Error cuadrático medio para conjuntos.

Conjunto	ECM entrenamiento sin ruido	ECM entrenamiento con ruido
General	2.07×10^{-4}	9.79×10^{-5}
Entrenamiento	7.46×10^{-5}	1.23×10^{-4}
Validación	7.90×10^{-4}	3.77×10^{-5}
Prueba	2.40×10^{-4}	3.95×10^{-5}

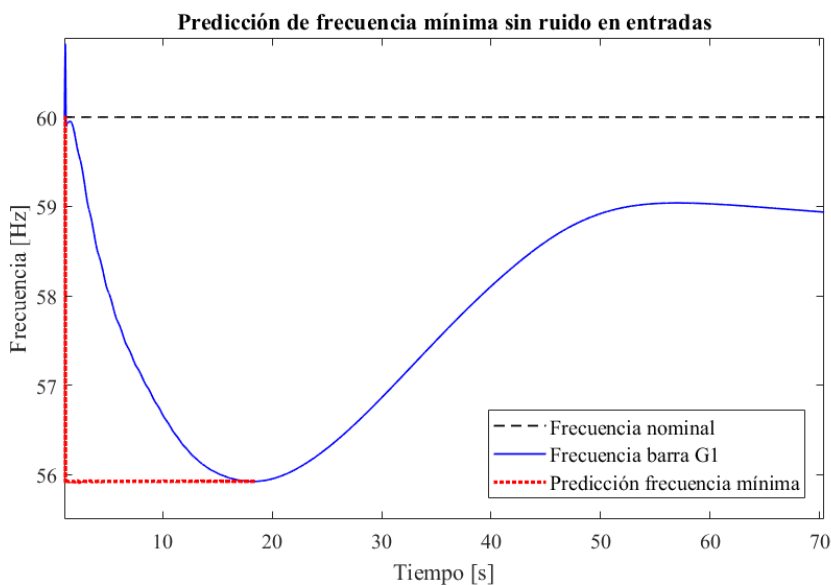


Figura 28. Predicción de frecuencia mínima con TDNN entrenada con ruido recibiendo entradas sin ruido.

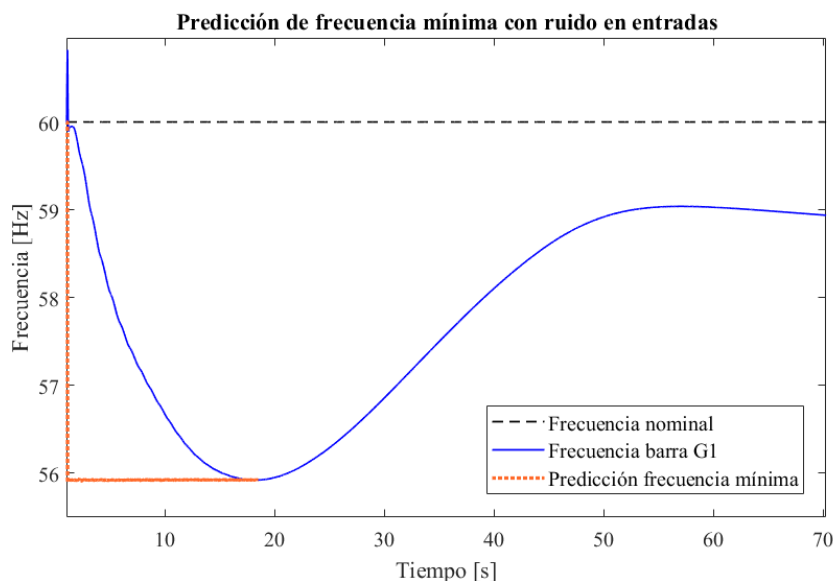


Figura 29. Predicción de frecuencia mínima con TDNN entrenada con ruido recibiendo entradas con ruido.

En la Tabla 4 se muestra un resumen de las pruebas realizadas con las dos redes neuronales entrenadas hasta este punto. Como se señaló, si la TDNN tuvo ruido en las señales de entrada durante su entrenamiento, tiene un comportamiento satisfactorio tanto si hay ruido como si no lo hay.

5.3.3 TDNN con reducción de muestreo y entradas

Con los resultados obtenidos hasta este punto y con el objetivo de eficientar el proceso de entrenamiento de la red neuronal usada, se aplican dos acciones principales a los datos. La primera es cambiar la tasa de muestreo a 50 muestras por segundo (el muestreo original es de 100 muestras por segundo). La segunda acción es reducir el

TABLA 4

Resultados pruebas desconexión G1 y $H_{tot} = 90\%$.

Entradas	ECM TDNN entrenada sin ruido	ECM TDNN entrenada con ruido
Sin ruido	1.48×10^{-6}	4.4017×10^{-6}
Con ruido	2.694×10^{-4}	8.8988×10^{-6}

número de entradas de la red neuronal, con el fin de tener menos datos que procesar. Esto se hizo tomando en cuenta [62], dónde se propone un método para la ubicación óptima de PMU para completa observabilidad del sistema de potencia.

Probando el método, se sugiere una selección de barras de colocación de PMU para el sistema de 39 barras New England. Las ubicaciones propuestas para posiciones de PMU son las barras 2, 3, 6, 8, 10, 12, 14, 16, 17, 19, 20, 22, 23, 25, 26, 29 y 39. Como se puede ver se redujo la cantidad de entradas de 39 a 17. De esta manera se procede a realizar un entrenamiento con ruido en entradas, 50 muestras por segundo y sólo 17 entradas.

Los criterios límite de entrenamiento se muestran en la Tabla 5. El entrenamiento concluye en 1 hora y 7 minutos alcanzando el límite de *val_checks*.

La Figura 30 muestra el desarrollo del error a medida que avanza el proceso de aprendizaje, hasta llegar al valor considerado óptimo. En Figura 31 se puede observar el desarrollo del gradiente, μ y las validaciones de error durante el entrenamiento.

Para verificar el apropiado funcionamiento de la última RNA entrenada, se realizan las mismas pruebas hechas a las otras redes en este capítulo. De una simulación donde hay un 10 % de reducción de inercia total en el sistema ($H_{tot} = 90\%$) y el G1 es desconectado al segundo 1, se toman las mediciones de ángulos de las 17 barras antes mencionadas. En la Figura 32 se puede ver la prueba de la TDNN cuando recibe las señales de entradas

TABLA 5

Límites de los criterios de entrenamiento TDNN con reducción de muestreo y entradas.

Nombre	Valor	Significado
epochs	1000	Número máximo de épocas para entrenar
mse	0	ECM objetivo
min_grad	1×10^{-6}	Gradiente de rendimiento mínimo
val_checks	6	Número máximo de épocas de incremento continuo del error

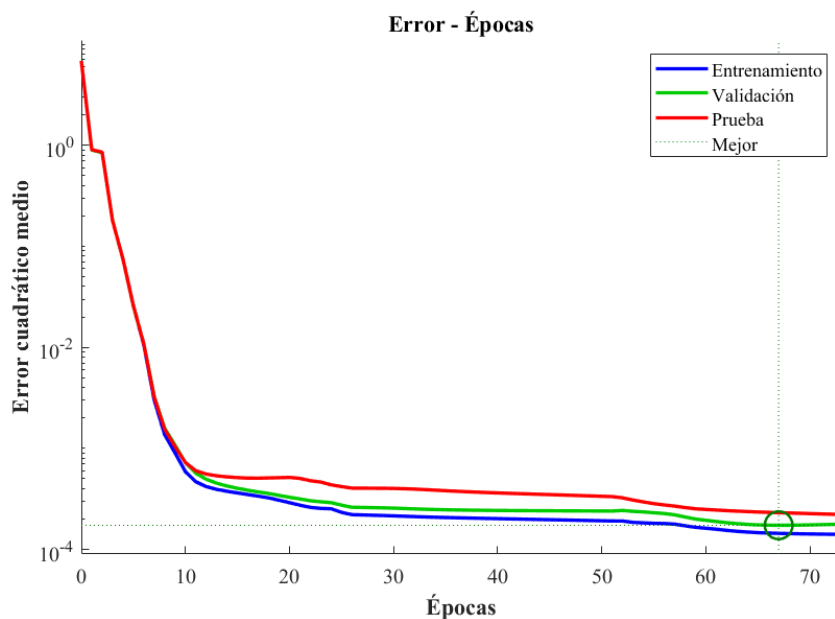


Figura 30. Desarrollo de ECM en entrenamiento de TDNN con ruido, reducción de entradas y muestreo.

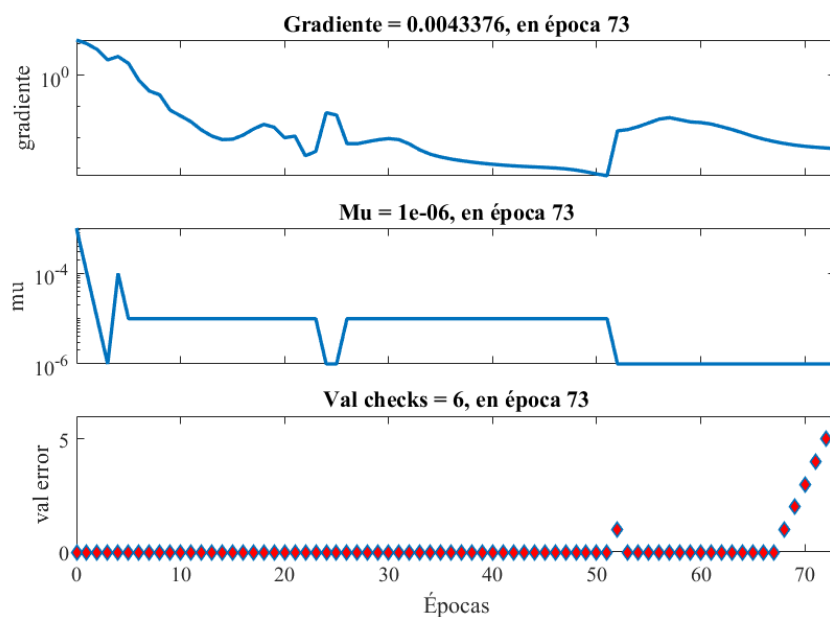


Figura 31. Gradiente, mu y pruebas de validación para TDNN con ruido, reducción de entradas y muestreo.

originales. En la barra del G1 la frecuencia mínima real es de 55.9234 Hz y el primer valor de frecuencia mínima predicha es de 55.9427 Hz. Si se promedian las primeras 10 predicciones resulta en un valor de 55.9257 Hz.

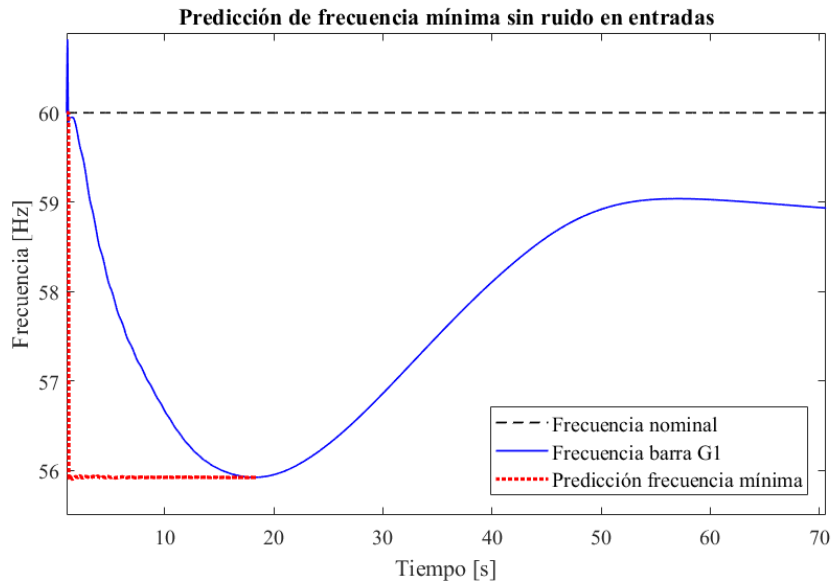


Figura 32. Predicción de frecuencia mínima con TDNN entrenada con ruido y reducción de muestreo y entradas recibiendo señales sin ruido.

En la Figura 33 se muestra el rendimiento para el mismo caso pero con ruido añadido en las entradas. El valor real de la frecuencia mínima es 55.9234 Hz, la primera predicción hecha da como resultado 55.9417 Hz y el promedio de las primeras 10 predicciones resulta en 55.927 Hz.

Por último, se presenta en Figura 34 una prueba sin reducción de inercia ($H_{tot} = 100\%$), para corroborar el funcionamiento en casos distintos. Aquí, el G2 es desconectado al segundo 1 provocando una caída en la frecuencia en el sistema que cuenta con toda su inercia original. En esta prueba la frecuencia mínima real es de 58.56 Hz, la primera predicción es de 58.5447 Hz y el promedio de las primeras 10 predicciones es de 58.5594 Hz.

A partir de este análisis se puede concluir que se debe tomar una decisión entre precisión y velocidad con esta última TDNN entrenada ya que la primera predicción es menos cercana al valor real que el promedio de las primeras 10 predicciones. Esperar este valor promediado supondría un retardo adicional de 0.1 segundos para obtener el valor

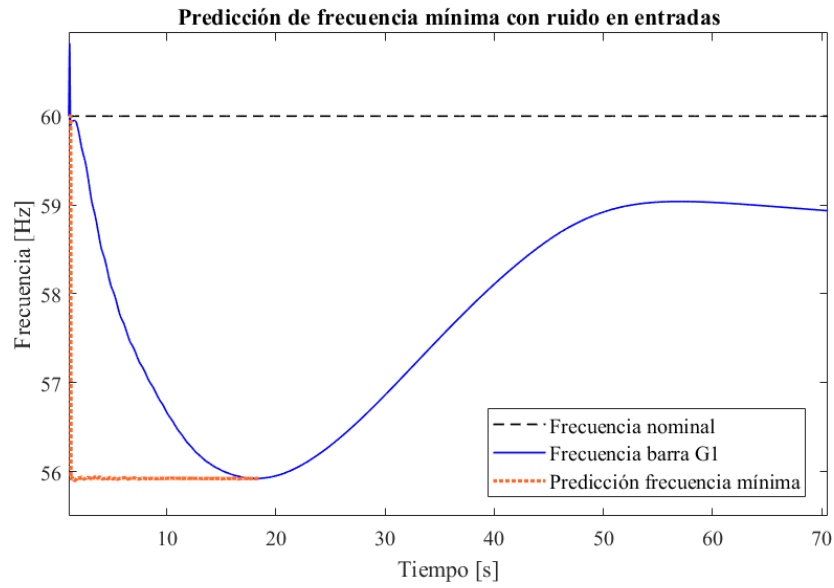


Figura 33. Predicción de frecuencia mínima con TDNN entrenada con ruido y reducción de muestreo y entradas recibiendo señales con ruido.

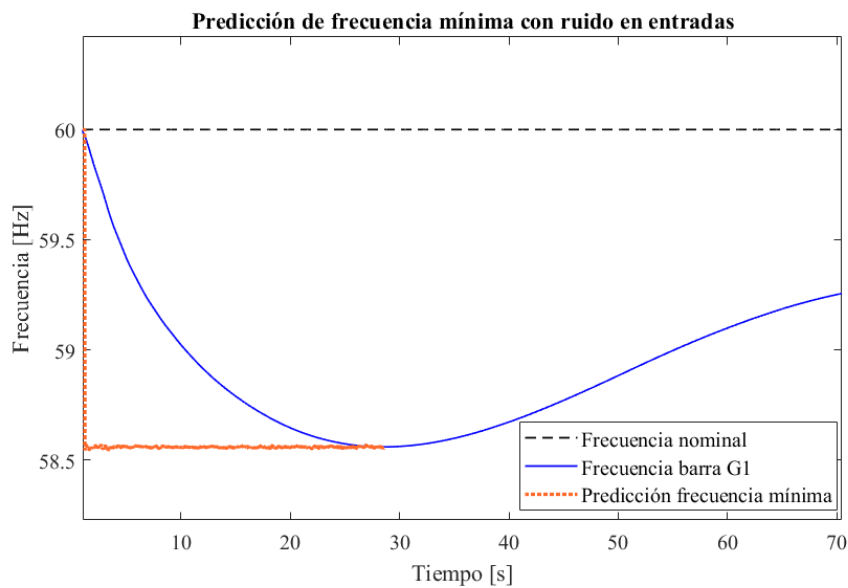


Figura 34. Predicción de frecuencia mínima con TDNN entrenada con ruido y reducción de muestreo y entradas, sin reducción de inercia.

deseado, que sumado a la espera de la ventana de entradas de la TDNN, resulta en un total de 0.2 segundos después de ocurrida la perturbación del sistema.

5.4 Conclusión

La incorporación de un creciente número de generadores basados en convertidores electrónicos de potencia a las redes eléctricas está provocando una disminución de la inercia rotacional total disponible para frenar las desviaciones de frecuencia en la red.

Esta inercia reducida en el sistema de potencia provoca un incremento en la tasa de cambio de frecuencia y un menor punto mínimo de frecuencia después de un disturbio. Dicho cambio de las dinámicas del sistema puede provocar activación no deseada de protecciones, alterando aun más el equilibrio del sistema.

En este capítulo se mostró el procedimiento para entrenar satisfactoriamente redes neuronales con retardo de tiempo con el fin de predecir el punto mínimo de frecuencia al que caerá después de un disturbio en sistemas de potencia con baja inercia.

Con el fin aumentar el realismo, también se hicieron pruebas con datos a los que se les añadió ruido. Se observó que si la TDNN no había recibido datos con ruido durante su entrenamiento, le resultaba difícil realizar una predicción precisa al darle como entradas datos con ruido.

A su vez se entrenó una TDNN con datos que incluían ruido y se identificó un comportamiento de predicción satisfactorio para datos limpios y ruidosos. Se continuó por entrenar solo modelos que involucren datos con ruido en su aprendizaje.

Con el fin de reducir el tiempo que tomaba el entrenamiento, se redujo el número de entradas, recibiendo datos solo de 17 de las 39 barras del sistema de prueba. Además se cambió la tasa de muestreo de 100 a 50 muestras por segundo. A partir de esto, hubo una considerable reducción de tiempo de entrenamiento.

Se presentaron pruebas donde se desconectan diferentes generadores de la red eléctrica y se cuenta con distintas cantidades de inercia. Los resultados demuestran un correcto funcionamiento del modelo entrenado para el propósito planteado. Se observa que en este último modelo, es conveniente hacer un promedio de las primeras 10 predicciones si se quiere mayor exactitud numérica, ya que el promedio mencionado se acerca más al valor real que la primera predicción hecha por la TDNN.

Las redes neuronales con retardo de tiempo demostraron ser una herramienta adecuada para tener un entendimiento temporal de los datos y ser capaces de proveer información de suma importancia en los sistemas eléctricos de potencia actuales.

CAPÍTULO 6

CONCLUSIÓN

6.1 Conclusiones

Debido al cambio continuo en los sistemas eléctricos de potencia modernos, la cantidad de inercia rotacional en ellos se está viendo disminuida. Principalmente la integración de fuentes de energía renovables basadas en convertidores electrónicos de potencia y la eliminación de centrales con generadores síncronos han causado la reducción en la inercia disponible para contrarrestar cambios de frecuencia originados por disturbios. Esto ha provocado que los procesos dinámicos sean cada vez más rápidos y extremos en SEPs con baja inercia.

Los operadores del sistema eléctrico, antes encargados de asegurar la estabilidad y confiabilidad del sistema, se están viendo obligados a operar en condiciones más extremas y a tomar decisiones en un marco de tiempo menor. De aquí nace la necesidad de buscar herramientas precisas y veloces que ayuden con el proceso de toma de decisiones.

El aprendizaje automático ha demostrado una gran capacidad de abstracción en diversas áreas del conocimiento, incluyendo los sistemas eléctricos de potencia. De esta forma se puso como objetivo corroborar la utilidad del aprendizaje automático y redes neuronales artificiales para análisis de SEP con baja inercia.

Como primer enfoque se utilizó la estructura de red neuronal perceptrón multi-

capa para detectar la desconexión de una unidad generadora del sistema de prueba en condiciones variables de inercia. Este MLP hizo la función de clasificador, con 10 posibles clases, en las que cada clase representa un generador. Las pruebas muestran el correcto e inmediato funcionamiento del modelo entrenado, detectando la salida del sistema de una unidad de generación e identificando cuál es esta unidad. De las 1210 simulaciones que componen los conjuntos de entrenamiento, validación y prueba, el modelo clasificó correctamente 89.8 %.

En la segunda parte del trabajo, se propuso entrenar una red neuronal con retardo de tiempo para predecir el valor mínimo al que cae la frecuencia del sistema después de ocurrido un disturbio. La estructura de red con retardo de tiempo demostró gran precisión en su predicción dando un amplio margen de tiempo antes de que se presente dicho valor mínimo en la red. También se probaron modelos que recibían entradas con ruido y los resultados fueron satisfactorios, en caso de haber tenido un entrenamiento donde los datos incluyan ruido. Los modelos se probaron también en diferentes condiciones de inercia total para corroborar su capacidad en distintas situaciones, siempre dando un valor numérico muy cercano al real.

Así se pudo comprobar un satisfactorio funcionamiento de la técnica de aprendizaje automático, las redes neuronales artificiales, para la evaluación temprana de estabilidad en sistemas eléctricos de potencia con baja inercia. Para esto es necesario contar con datos precisos provenientes de simulaciones fieles a los sistemas reales y que incluyan los eventos que son de mayor interés o preocupación.

6.2 Recomendaciones para futuros trabajos

Por limitaciones de tiempo, no se pudo concluir en este trabajo las pruebas en tiempo real de los modelos entrenados. Se recomienda realizar estas pruebas en simuladores de tiempo real como OPAL-RT y, de ser posible, usar la técnica *hardware-in-the-loop*.

También se recomienda, la inclusión de escenarios no contemplados en las simulaciones RMS, como desconexión de carga y apertura de líneas de transmisión, así como añadir en el modelado de la red convertidores electrónicos de potencia.

Considero conveniente la exploración de otras técnicas de aprendizaje automático (fuera de las RNA) en sistemas eléctricos de potencia con baja inercia.

REFERENCIAS

1. U. Tamrakar, D. Shrestha, M. Maharjan, B. Bhattarai, T. Hansen y R. Tonkoski, *Virtual Inertia: Current Trends and Future Directions*. Applied Sciences, 7(7):654, Junio 2017, ISSN 2076-3417.
2. U. Markovic, O. Stanojev, E. Vrettos, P. Aristidou y G. Hug, *Understanding Stability of Low-Inertia Systems*. Preprint, engrXiv, Febrero 2019.
3. O. A. Alimi, K. Ouahada y A. M. Abu-Mahfouz, *A Review of Machine Learning Approaches to Power System Security and Stability*. IEEE Acces, 8:20, 2020.
4. Z. Wang, H. He, Z. Wan y Y. Sun, *Coordinated Topology Attacks in Smart Grid Using Deep Reinforcement Learning*. IEEE Transactions on Industrial Informatics, 17(2):1407–1415, Febrero 2021, ISSN 1941-0050.
5. L. A. Maglaras y J. Jiang, *Intrusion Detection in SCADA Systems Using Machine Learning Techniques*. En *2014 Science and Information Conference*, páginas 626–631, Agosto 2014.
6. H. Liu, F. Hussain, Y. Shen, S. Arif, A. Nazir y M. Abubakar, *Complex Power Quality Disturbances Classification via Curvelet Transform and Deep Learning*. Electric Power Systems Research, 163:1–9, Octubre 2018, ISSN 03787796.
7. W. Kong, Z. Dong, Y. Jia, D. J. Hill, Y. Xu y Y. Zhang, *Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network*. IEEE Transactions on Smart Grid, 10(1):841–851, Enero 2019, ISSN 1949-3053, 1949-3061.
8. C. Huang y P. Kuo, *Multiple-Input Deep Convolutional Neural Network Model for Short-Term Photovoltaic Power Forecasting*. IEEE Access, 7:74822–74834, 2019, ISSN 2169-3536.
9. Y. Yang, Y. Huang, J. Liu, Y. Liu, T. Liu y Y. Xiang, *Measurement-Based Cell-DT Method for Power System Transient Stability Classification*. CSEE Journal of Power and Energy Systems, 3(3):278–285, Septiembre 2017, ISSN 2096-0042.
10. K. S. Sajan, V. Kumar y B. Tyagi, *ICA Based Artificial Neural Network Model for Voltage Stability Monitoring*. En *TENCON 2015 - 2015 IEEE Region 10 Conference*, páginas 1–3, Noviembre 2015.
11. T. S. Abdelgayed, W. G. Morsi y T. S. Sidhu, *Fault Detection and Classification Based on Co-Training of Semisupervised Machine Learning*. IEEE Transactions on Industrial Electronics, 65(2):1595–1605, Febrero 2018, ISSN 1557-9948.

12. T. S. Abdelgayed, W. G. Morsi y T. S. Sidhu, *A New Approach for Fault Classification in Microgrids Using Optimal Wavelet Functions Matching Pursuit*. IEEE Transactions on Smart Grid, 9(5):4838–4846, Septiembre 2018, ISSN 1949-3061.
13. S. Fan, C. Mao y L. Chen, *Next-Day Electricity-Price Forecasting Using a Hybrid Network*. Generation, Transmission & Distribution, IET, 1:176–182, Febrero 2007.
14. P. Kundur, *Power System Stability And Control*. McGraw-Hill, 1994, ISBN 978-0-07-063515-9.
15. C. Hsiao-Dong, F. Wu y P. Varaiya, *Foundations of Direct Methods for Power System Transient Stability Analysis*. IEEE Transactions on Circuits and Systems, 34(2):160–173, Febrero 1987, ISSN 0098-4094.
16. P. Denholm, T. Mai, B. Kroposki, R. Kenyon y M. O'Malley, *Inertia and the Power Grid: A Guide Without the Spin*. Informe técnico NREL/TP-6A20-73856, 1659820, MainId:6231, Mayo 2020.
17. P. Tielens y D. Van Hertem, *The Relevance of Inertia in Power Systems*. Renewable and Sustainable Energy Reviews, 55:999–1009, Marzo 2016, ISSN 13640321.
18. V. Malbasa, C. Zheng, P. Chen, T. Popovic y M. Kezunovic, *Voltage Stability Prediction Using Active Machine Learning*. IEEE Transactions on Smart Grid, 8(6):3117–3124, Noviembre 2017, ISSN 1949-3061.
19. A. Gupta, G. Gurralla y P. S. Sastry, *An Online Power System Stability Monitoring System Using Convolutional Neural Networks*. IEEE Transactions on Power Systems, 34(2):864–872, Marzo 2019, ISSN 1558-0679.
20. H. R. Chamorro, A. D. Orjuela-Canon, D. Ganger, M. Persson, F. Gonzalez-Longatt, V. K. Sood y W. Martinez, *Nadir Frequency Estimation in Low-Inertia Power Systems*. En *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, páginas 918–922, Delft, Netherlands, Junio 2020. IEEE, ISBN 978-1-72815-635-4.
21. B. Wang, B. Fang, Y. Wang, H. Liu y Y. Liu, *Power System Transient Stability Assessment Based on Big Data and the Core Vector Machine*. IEEE Transactions on Smart Grid, 7(5):2561–2570, Septiembre 2016, ISSN 1949-3053, 1949-3061.
22. J. D. Echeverria, C. J. Cepeda y D. G. Colome, *Real-Time Transient Stability Assessment of Electric Power Systems Using Predictive-SIME Based on Machine Learning*. En *2017 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, páginas 1–6, Quito, Septiembre 2017. IEEE, ISBN 978-1-5386-3312-0.
23. T. Mitchell, *Machine Learning*. McGraw-Hill, 1997, ISBN 978-0-07-115467-3.
24. Ü. Dogan, J. Edelbrunner y Ioannis Iossifidis, *Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior*. En *2011 IEEE International Conference on Robotics and Biomimetics*, páginas 1837–1843, Diciembre 2011.

25. S. Shalev-Shwartz, S. Shammah y A. Shashua, *Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving*. arXiv:1610.03295 [cs, stat], Octubre 2016.
26. L. Deng y X. Li, *Machine Learning Paradigms for Speech Recognition: An Overview*. IEEE Transactions on Audio, Speech, and Language Processing, 21(5):1060–1089, Mayo 2013, ISSN 1558-7916, 1558-7924.
27. S. Emerson, R. Kennedy, L. O’Shea y J. O’Brien, *Trends and Applications of Machine Learning in Quantitative Finance*. SSRN Scholarly Paper ID 3397005, Social Science Research Network, Rochester, NY, Mayo 2019.
28. S. Loussaief y A. Abdelkrim, *Machine Learning Framework for Image Classification*. En *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, páginas 58–61, Diciembre 2016.
29. M. de Bruijne, *Machine Learning Approaches in Medical Image Analysis: From Detection to Diagnosis*. Medical Image Analysis, 33:94–97, Octubre 2016, ISSN 1361-8415.
30. E. Ikonomakis, S. Kotsiantis y V. Tampakas, *Text Classification Using Machine Learning Techniques*. WSEAS transactions on computers, 4:966–974, Agosto 2005, ISSN 1109-2750.
31. S. Scher y G. Messori, *Predicting Weather Forecast Uncertainty with Machine Learning*. Quarterly Journal of the Royal Meteorological Society, 144(717):2830–2841, 2018, ISSN 1477-870X.
32. S. S. Khanal, P.W.C. Prasad, A. Alsadoon y A. Maag, *A Systematic Review: Machine Learning Based Recommendation Systems for e-Learning*. Education and Information Technologies, 25(4):2635–2664, Julio 2020, ISSN 1360-2357, 1573-7608.
33. J. A. Gómez, J. E. Patiño, J. C. Duque y S. Passos, *Spatiotemporal Modeling of Urban Growth Using Machine Learning*. Remote Sensing, 12(1):109, Enero 2020.
34. A. Géron, *Hands-On Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Beijing ; Boston, Abril 2017, ISBN 978-1-4919-6229-9.
35. A. C. Müller y S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. Octubre 2016.
36. J. Krohn, G. Beyleveld y A. Bassens, *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence*. Boston, MA, Septiembre 2019, ISBN 978-0-13-511669-2.
37. D. Meyer, *Support Vector Machines*. Informe técnico, FH Technikum Wien.
38. O. A. Alimi, K. Ouahada y A. M. Abu-Mahfouz, *Real Time Security Assessment of the Power System Using a Hybrid Support Vector Machine and Multilayer Perceptron Neural Network Algorithms*. Sustainability, 11(13):3586, Enero 2019.

39. D. Lineweber y S. McNulty, *The Cost of Power Disturbances to Industrial & Digital Economy Companies*. Informe técnico, Consortium for Electrical Infrastructure to Support a Digital Society.
40. S. Corsi y C. Sabelli, *General Blackout in Italy Sunday September 28, 2003, h. 03:28:00*. En *IEEE Power Engineering Society General Meeting, 2004.*, páginas 1691–1702, Junio 2004.
41. R. Yan, N. Masood, T. Kumar Saha, F. Bai y H. Gu, *The Anatomy of the 2016 South Australia Blackout: A Catastrophic Event in a High Renewable Network*. *IEEE Transactions on Power Systems*, 33(5):5374–5388, Septiembre 2018, ISSN 1558-0679.
42. N. Harmsen, *SA Blackout Costs Revealed with Fears of More Outages*. ABC News, Diciembre 2016.
43. P. Yuan, Q. Zhang, T. Zhang, C. Chi, X. Zhang, P. Li y X. Gong, *Analysis and Enlightenment of the Blackouts in Argentina and New York*. En *2019 Chinese Automation Congress (CAC)*, páginas 5879–5884, Noviembre 2019.
44. A.K. Jain, J. Mao y K.M. Mohiuddin, *Artificial Neural Networks: A Tutorial*. *Computer*, 29(3):31–44, Marzo 1996, ISSN 1558-0814.
45. P. Páramo Balsa, J. Roldan Fernandez, F. Gonzalez Longatt, M. Burgos Payan y J. Riquelme Santos, *Fault Location In A VSC-HVDC Link Using Neural Networks*. *DYNA*, 95(1):668–673, Noviembre 2020, ISSN 19891490.
46. A. P. Memon, A. S. Memon, A. Akhund y R. H. Memon, *Multilayer Perceptrons Neural Network Automatic Voltage Regulator With Applicability And Improvement In Power System Transient Stability*. 9:9, 2013.
47. A. R. Bahmanyar y A. Karami, *Power System Voltage Stability Monitoring Using Artificial Neural Networks with a Reduced Set of Inputs*. *International Journal of Electrical Power & Energy Systems*, 58:246–256, Junio 2014, ISSN 0142-0615.
48. M. A. Pai, *Energy Function Analysis for Power System Stability*. Springer Science & Business Media, Diciembre 2012, ISBN 978-1-4613-1635-0.
49. E. D. Sontag, *Feedback Stabilization Using Two-Hidden-Layer Nets*. En *1991 American Control Conference*, páginas 815–820, Junio 1991.
50. T. Masters, *Practical Neural Network Recipes in C++*. Morgan Kaufmann, 1993, ISBN 978-0-12-479040-7.
51. J. E. Dennis y R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, Enero 1987, ISBN 978-0-89871-364-0.
52. D. F. Shanno, *Conjugate Gradient Methods with Inexact Searches*. *Mathematics of Operations Research*, 3(3):244–256, Agosto 1978, ISSN 0364-765X.

53. P. E. Gill, W. Murray y M. H. Wright, *Practical Optimization*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Enero 2019, ISBN 978-1-61197-559-8.
54. A. Ulbig, T. S. Borsche y G. Andersson, *Impact of Low Rotational Inertia on Power System Stability and Operation*. IFAC Proceedings Volumes, 47(3):7290–7297, Enero 2014, ISSN 1474-6670.
55. K. S. Ratnam, K. Palanisamy y G. Yang, *Future Low-Inertia Power Systems: Requirements, Issues, and Solutions - A Review*. Renewable and Sustainable Energy Reviews, 124:109773, Mayo 2020, ISSN 13640321.
56. H. Bevrani, A. Ghosh y G. Ledwich, *Renewable Energy Sources and Frequency Regulation: Survey and New Perspectives*. IET Renewable Power Generation, 4(5):438–457, Septiembre 2010, ISSN 1752-1424.
57. N. Soni, S. Doolla y M. C. Chandorkar, *Improvement of Transient Response in Microgrids Using Virtual Inertia*. IEEE Transactions on Power Delivery, 28(3):1830–1838, Julio 2013, ISSN 1937-4208.
58. C. Seneviratne y C. Ozansoy, *Frequency Response Due to a Large Generator Loss with the Increasing Penetration of Wind/PV Generation – A Literature Review*. Renewable and Sustainable Energy Reviews, 57:659–668, Mayo 2016, ISSN 1364-0321.
59. M. Brown, M. Biswal, S. Brahma, S. J. Ranade y H. Cao, *Characterizing and Quantifying Noise in PMU Data*. En *2016 IEEE Power and Energy Society General Meeting (PESGM)*, páginas 1–5, Julio 2016.
60. D. Molina, J. Liang, R. Harley y G. K. Venayagamoorthy, *Comparison of TDNN and RNN Performances for Neuro-Identification on Small to Medium-Sized Power Systems*. En *2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, páginas 1–8, Abril 2011.
61. M.T. Hagan y M.B. Menhaj, *Training Feedforward Networks with the Marquardt Algorithm*. IEEE Transactions on Neural Networks, 5(6):989–993, Noviembre 1994, ISSN 1941-0093.
62. S. Chakrabarti y E. Kyriakides, *Optimal Placement of Phasor Measurement Units for Power System Observability*. IEEE Transactions on Power Systems, 23(3):1433–1440, Agosto 2008, ISSN 1558-0679.

RESUMEN AUTOBIOGRÁFICO

Jesús Alejandro Verduzco Cota

Candidato para el grado de Maestría en Ciencias de la Ingeniería Eléctrica

Tesis: EVALUACIÓN TEMPRANA DE ESTABILIDAD Y SEGURIDAD DE SISTEMAS DE POTENCIA CON BAJA INERCIA UTILIZANDO MACHINE LEARNING

Biografía: Nacido en Hermosillo, Sonora, México, el 15 de octubre de 1994, hijo de Oscar Rosalío Verduzco Soto y María Reyna Cota Vásquez.

Educación: Asistió al Instituto Tecnológico de Hermosillo donde estudió la carrera de Ingeniería Eléctrica con especialidad en Sistemas Eléctricos de Potencia. En 2018, se graduó con mención honorífica.