

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



PRONÓSTICO DE LA DEMANDA ELÉCTRICA A CORTO PLAZO UTILIZANDO REDES NEURONALES ARTIFICIALES APLICADO AL SISTEMA INTERCONECTADO NACIONAL

TESIS

PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS CON
MENCIÓN EN SISTEMAS DE POTENCIA

ELABORADO POR

EVER RONALD FERNANDEZ QUIÑONES

ASESOR

M. Sc. ZOCIMO ÑAUPARI HUATUCO

LIMA - PERÚ

2014

DEDICATORIA

Con todo mi corazón dedico esta tesis a mis hijos Renzo, Brandon y Astrid

AGRADECIMIENTOS

Agradezco de manera muy especial a mi asesor, M.Sc. Zocimo Ñaupari Huatuco, por sus consejos y el apoyo brindado para el desarrollo de la presente tesis. Así mismo agradecer al Dr. Yuri Molina Rodriguez y al Ing. Adderly Huerta Valdivia, por su invaluable apoyo.

UNIVERSIDAD NACIONAL DE INGENIERIA
Facultad de Ingeniería Eléctrica y Electrónica
Sección de Posgrado y Segunda Especialización

TESIS DE MAESTRÍA

ACTA DE SUSTENTACIÓN

Siendo las 15:00 horas los suscritos catedráticos de la Facultad de Ingeniería Eléctrica y Electrónica de Universidad Nacional de Ingeniería, miembros del Jurado, presidida por el Señor Decano de la Facultad, nos reunimos para oír la sustentación de la tesis de maestría titulada:

“PRONÓSTICO DE LA DEMANDA ELÉCTRICA A CORTO PLAZO UTILIZANDO REDES NEURONALES ARTIFICIALES APLICADO AL SISTEMA INTERCONECTADO NACIONAL”

Presentada por el ingeniero EVER RONALD FERNÁNDEZ QUIÑONES

Luego de la presentación del candidato por el Presidente del Jurado, se realizó la exposición y se respondió a las preguntas hechas por los miembros del Jurado. La sustentación pública de la tesis terminó a las 16:20 horas reuniéndose luego los miembros del Jurado en for reservada para deliberar y acordar recomendar su distinción, a partir de (Diezochos), y el ar al Consejo Universitario la recomendación de que se otorgue el **Grado Académico de Maestro en Ciencias con mención en Sistemas de Potencia**, en merito al cual suscribieron para constancia, el correspondiente Libro de Grados N ° 1, Folio N ° 072, con fecha, Lima 21 de julio del 2014.

ÍNDICE DE CONTENIDOS

INTRODUCCION	1
CAPITULO I	2
PLANTEAMIENTO DEL PROBLEMA	2
1.1 Descripción de la realidad problemática	2
1.2 Justificación e importancia de la investigación	2
1.3 Objetivos	2
1.4 Hipótesis	3
1.5 Variables e indicadores.....	3
1.6 Periodo de análisis.....	3
1.7 Fuentes de información e instrumentos usados	3
1.8 Organización de la tesis	4
CAPITULO II	5
MARCO TEORICO: DEMANDA ELÉCTRICA A CORTO PLAZO	5
2.1 Horizontes de proyección de la demanda eléctrica	5
2.1.1 Proyección de la demanda eléctrica a largo plazo.....	5
2.1.2 Proyección de la demanda eléctrica a mediano plazo	5
2.1.3 Proyección de la demanda eléctrica a corto plazo.....	6
2.1.4 Proyección de la demanda eléctrica a cortísimo plazo	6
2.2 Importancia de la proyección de la demanda eléctrica a corto plazo ...	7
2.3 Proyección de la demanda eléctrica a corto plazo del SEIN	7
2.3.1 Consideraciones del pronóstico de la demanda del SEIN	8
2.3.2 Metodología del pronóstico diario para días ordinarios	8
2.3.3 Metodología del pronóstico diario para días feriados	11
2.4 Análisis de la metodología de la proyección diaria del SEIN	11
2.5 Características de la demanda eléctrica del SEIN a corto plazo	17

2.6 Factores que influyen en la demanda eléctrica	19
2.6.1 Factores económicos	19
2.6.2 Factores temporales	19
2.6.3 Factores climáticos	20
2.6.4 Disturbios aleatorios.....	20
2.7 Modelos de proyección de la demanda eléctrica a corto plazo	20
2.7.1 Modelo pico de carga.....	20
2.7.2 Modelo de perfil de carga.....	21
2.8 Técnicas de pronóstico de la demanda eléctrica	21
2.8.1 Regresión lineal múltiple	22
2.8.2 Alisamiento exponencial.....	22
2.8.3 Series temporales estocásticas.....	22
2.8.4 Método de espacio de estado	24
2.8.5 Modelo econométrico	25
 CAPITULO III	 26
 APLICACIÓN DE REDES NEURONALES ARTIFICIALES PARA LA PROYECCION DE LA DEMANDA ELECTRICA A CORTO PLAZO	 26
3.1 Introducción.....	26
3.2 Red neuronal biológica	26
3.3 Estructura de un sistema neuronal artificial	28
3.4 Modelo de una neurona artificial	29
3.4.1 Entradas y salidas.....	29
3.4.2 Pesos sinápticos	30
3.4.3 Regla de propagación	30
3.4.4 Función de activación.....	30
3.4.5 Función de salida	31
3.5 Arquitectura de una red neuronal artificial	31
3.6 Aprendizaje de una red neuronal artificial	33
3.6.1 Fase de entrenamiento	33
3.6.2 Fase de prueba	33
3.7 Métodos de entrenamiento de redes neuronales artificiales	34

3.7.1	Métodos de gradiente descendente	34
3.7.2	El algoritmo de backpropagation	35
CAPITULO IV		36
CASO DE ESTUDIO		36
4.1	Introducción	36
4.2	Selección de la barra a proyectar	36
4.3	Elaboración de la base de datos	37
4.3.1	Depuración de datos históricos	37
4.3.2	Ordenamiento de la base de datos	38
4.4	Características de la red neuronal artificial usado	38
4.4.1	Topología de la red neuronal artificial	38
4.4.2	Función de activación de la red neuronal artificial	39
4.4.3	Método de entrenamiento de la red neuronal artificial	39
4.4.4	Arquitectura de la red neuronal artificial	40
4.5	Indicador del pronóstico de la demanda eléctrica	41
CAPITULO V		42
SIMULACIONES Y RESULTADOS		42
5.1	Diseño y entrenamiento de la red neuronal artificial	42
5.2	Software de simulación	44
5.3	Validación de los resultados de la proyección	46
5.4	Resultados de las proyecciones diarias del SEIN	46
5.4.1	Proyección diaria usando la RNA con 2 capas ocultas	46
5.4.2	Proyección diaria usando la RNA con 3 capas ocultas	51
5.4.3	Proyección diaria usando RNA con 3 capas ocultas sin considerar días feriados ..	56
5.5	Comparación de modelos de proyecciones del COES y la red neuronal artificial	58
5.6	Reducción del costo de operación	64
5.6.1	Premisas para cuantificar los nuevos costos de operación	64
5.6.2	Resultados obtenidos de la reducción del costo de operación	65

CONCLUSIONES	67
RECOMENDACIONES	70
BIBLIOGRAFÍA	71
ANEXOS	73

ÍNDICE DE TABLAS

Tabla 2.1	Formación de la base de datos de acuerdo a la característica similar del día de la semana.....	9
Tabla 2.2	Resumen de indicadores diarias del SEIN del año 2012	12
Tabla 3.1	Funciones de activación	31
Tabla 4.1	Formato de ingreso de datos.....	38
Tabla 5.1	MAPE de las proyecciones del SEIN usando RNA con 2 capas ocultas....	47
Tabla 5.2	MAPE de las proyecciones del SEIN usando RNA con 3 capas ocultas....	52
Tabla 5.3	MAPE de las proyecciones del SEIN usando RNA con 3 capas ocultas sin considerar días feriados.....	57
Tabla 5.4	Cuadro Comparativo de MAPE (abril-mayo-junio) usando RNA y metodología del COES.....	59
Tabla 5.5	Cuadro Comparativo de MAPE (julio - agosto) usando RNA y metodología del COES.....	62
Tabla 5.6	Relación de días para cuantificar el costo de operación.....	65
Tabla 5.7	Cuantificación del ahorro obtenido por mejor proyección de la RNA	66

ÍNDICE DE ILUSTRACIONES

Fig. N° 2.1	Diagrama de intervalo de tiempo vs proyección de la demanda	6
Fig. N° 2.2	Secuencia de proyección de la demanda eléctrica diaria del SEIN	11
Fig. N° 2.3	MAPE de la proyección del SEIN de enero del 2012	13
Fig. N° 2.4	Demanda del SEIN proyectado y real del día 08.ene.2012	14
Fig. N° 2.5	MAPE de la proyección del SEIN de julio del 2012	15
Fig. N° 2.6	Demanda del SEIN proyectado y real del día 28.jul.2012	15
Fig. N° 2.7	MAPE de la proyección del SEIN de diciembre del 2012	16
Fig. N° 2.8	Demanda del SEIN proyectado y real del día 29.dic.2012	17
Fig. N° 2.9	Demanda eléctrica típica por días de la semana del SEIN	18
Fig. N° 2.10	Demanda eléctrica típica diaria del SEIN por estación del año	18
Fig. N° 2.11	Modelo de series temporales	23
Fig. N° 3.1	Estructura típica de una neurona biológica	27
Fig. N° 3.2	Modelo de una neurona artificial	29
Fig. N° 3.3	Modelo de RNA Perceptrón Simple	32
Fig. N° 4.1	Perfil de demanda eléctrica del SEIN en periodo de avenida	37
Fig. N° 4.2	Función de transferencia lineal “purelin”	39
Fig. N° 4.3	Arquitectura de la RNA sin considerar las estaciones del año	40
Fig. N° 4.4	Arquitectura de la RNA considerando las estaciones del año	41
Fig. N° 5.1	Proceso de entrenamiento de la RNA	43
Fig. N° 5.2	Módulo de entrenamiento de la RNA	44
Fig. N° 5.3	Módulo de proyección y análisis de performance de la RNA	45
Fig. N° 5.4	Módulo de proyección diaria de la demanda	45
Fig. N° 5.5	MAPE de las proyecciones del SEIN usando RNA con 2 capas ocultas	46
Fig. N° 5.6	Proyección del 23 agosto 2012 con histograma de errores	49
Fig. N° 5.7	Proyección del 24 agosto 2012 con histograma de errores	50
Fig. N° 5.8	MAPE de las proyecciones del SEIN usando RNA con 3 capas ocultas	51
Fig. N° 5.9	Proyección del 15 junio 2012 con histograma de errores	54
Fig. N° 5.10	Proyección del 08 agosto 2012 con histograma de errores	55

Fig. N° 5.11	MAPE de proyección del SEIN usando RNA con 3 capas sin considerar días feriados	56
Fig. N° 5.12	Comparativo de MAPE de abril usando RNA y método COES	60
Fig. N° 5.13	Comparativo de MAPE de mayo usando RNA y método COES	60
Fig. N° 5.14	Comparativo de MAPE de junio usando RNA y método COES	61
Fig. N° 5.15	Comparativo de MAPE de julio usando RNA y método COES	63
Fig. N° 5.16	Comparativo de MAPE de agosto usando RNA y método COES	63

RESUMEN

En la presente tesis se desarrolla una propuesta de previsión de la demanda eléctrica a corto plazo, basado en Redes Neuronales Artificiales y aplicado al Sistema Eléctrico Interconectado Nacional (SEIN), el cual es modelado como una sola barra; con el objetivo de minimizar las diferencias entre la demanda eléctrica proyectada y la demanda real. Esta nueva alternativa de previsión de demanda eléctrica, se puede generalizar para cualquier barra de un Sistema Eléctrico de Potencia.

En esta tesis se desarrolla una metodología para la construcción de una base de datos de perfiles de demanda eléctrica cada media hora, agrupados en días típicos, periodos estacionales y feriados. La validez de los resultados obtenidos se ha verificado con datos históricos reales. También se analiza el efecto que las variables climatológicas influyen en la demanda eléctrica.

ABSTRACT

In this thesis develops a proposal for forecasting electricity demand in the short term, based on artificial neural networks and applied to the National Interconnected Electric System (SEIN), which is modeled as a single bar is developed in this thesis; with the objective minimize the differences between the projected electricity demand and real. This new alternative for forecasting electricity demand, can be generalized to any bar of the National Interconnected System.

This thesis develop a methodology for building a database of profiles of half-hourly electricity demand, grouped in typical days, seasonal periods and holidays. The validity of the results has been verified with real historical data. So also analyze e effect of climatic variables influencing electricity demand.

INTRODUCCION

Con el crecimiento de los sistemas eléctricos de potencia y el aumento de su complejidad, muchos factores influyen en la generación de energía eléctrica y el consumo (gestión de la demanda eléctrica, intercambio de energía, precios spot, productores independientes de energía, energía no convencionales, etc.). Por lo tanto, la previsión de la demanda eléctrica se ha vuelto compleja y requiere mayor precisión. La relación entre la demanda y los factores exógenos es compleja y no lineal, por lo que es difícil desarrollar un modelo a través de técnicas convencionales, como series de tiempo y análisis de regresión lineal [1].

La predicción de la demanda de electricidad se realiza para horizontes de corto, mediano y largo plazo. El corto plazo corresponde a la predicción en tiempo real para la hora siguiente y con un horizonte de hasta una semana; el pronóstico se basa en los datos históricos de demanda diaria. El mediano plazo corresponde al pronóstico mensual con un horizonte de hasta un año a partir de la demanda histórica de electricidad, y la predicción de algunas variables explicativas como el crecimiento de la economía o el clima. Finalmente, el largo plazo abarca un horizonte de uno hasta diez años; la predicción de la demanda en el largo plazo es usada para la planeación y expansión de la generación y de la transmisión; los datos usados para este pronóstico son la demanda anual, el precio de la electricidad, el precio de productos sustitutos, la evolución demográfica e indicadores económicos, entre otros.

Las Redes Neuronales Artificiales se han aplicado con éxito en varios problemas, tales como: planificación, control, análisis, protección, previsión de la demanda eléctrica, análisis de seguridad, y diagnóstico de averías, entre otras aplicaciones a los sistemas eléctricos de potencia [1]. Las capacidades de aprendizaje y procesamiento no lineal de las Redes Neuronales Artificiales permiten desarrollar un método de proyección de demanda eléctrica cuyos resultados se mostraran en la presente tesis.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción de la realidad problemática

En el Perú, el Comité de Operación Económica del Sistema Interconectado Nacional (COES) realiza el pronóstico diario de la demanda eléctrica del SEIN y también realiza el pronóstico desgregado por aéreas operativas (Norte, Centro y Sur). El cálculo del perfil diario de la demanda eléctrica se realiza con datos históricos de perfiles de demandas de días representativos. El procedimiento de proyección de la demanda eléctrica, en muchas oportunidades arroja resultados con altas desviaciones respecto a los valores reales de la demanda, como lo ocurrido el día 21.06.13 en periodo de mínima demanda, la desviación llegó hasta -4.01 % (168 MW). El día 18.06.13 en periodo de media demanda la desviación llegó hasta -4.20 % (207 MW). Esta información se encuentra publicada en el portal web del COES.

1.2 Justificación e importancia de la investigación

Dependiendo del tipo de agente del mercado, la predicción de la demanda eléctrica a corto plazo para el sistema eléctrico se justifica por las siguientes razones: Para los generadores, permite definir la cantidad de energía a generar (ofertar) para el día siguiente y programar la operación de sus unidades. Para los distribuidores y transmisores, permite realizar el monitoreo del comportamiento de la red con el fin de identificar los puntos críticos que en el futuro pueden llevar a grandes perturbaciones. Para el coordinador, es un insumo fundamental para la programación y operación de las unidades de generación, minimizando los ajustes del programa de operación en tiempo real, con el fin de evitar los sobrecostos en la producción de la electricidad.

Las consecuencias de un mal pronóstico de la demanda eléctrica provocan desviaciones al despacho de las centrales de generación, influye en la toma inexacta de decisiones del coordinador de tiempo real y hace más cara la operación diaria del sistema eléctrico de potencia.

1.3 Objetivos

El objetivo de la presente tesis es desarrollar una metodología basada en Redes Neuronales Artificiales para pronosticar la demanda eléctrica a corto plazo aplicado al

Sistema Eléctrico Interconectado Nacional, minimizando las diferencias entre la demanda eléctrica proyectada y la demanda eléctrica ejecutada. La meta es obtener errores medios porcentuales absolutos menores del 2%. Así también analizar el efecto de las diferentes variables que influyen sobre la demanda incluyendo variables climatológicas y otras variables no estacionales. La validez de la metodología propuesta se verificó con datos históricos reales.

1.4 Hipótesis

La demanda de energía eléctrica en el Sistema Interconectado Nacional (SEIN) representa un comportamiento dinámico, cuyas características y patrones de tendencia fundamentales pueden identificarse y tienen un comportamiento en un determinado periodo. Las redes neuronales artificiales permiten establecer mapeos y relaciones adaptativas no lineales y complejas entre variables de diferentes características y unidades, que permiten establecer la tendencia de la demanda eléctrica en un periodo específico.

1.5 Variables e indicadores

La variable independiente en el presente estudio es la topología de las Redes Neuronales Artificiales el cual se tiene que ir modificando hasta conseguir el objetivo y la variable dependiente es el valor de la demanda eléctrica diaria en MW.

Para validar los resultados de la proyección de la demanda se usó el indicador MAPE: Error medio porcentual absoluto, para cada día estimado. En el Capítulo IV, subtítulo 4.5 de la presente tesis, se detalla el uso de este indicador en la proyección de la demanda eléctrica.

1.6 Periodo de análisis

Se ha recopilado información histórica para la creación de la base de datos, del perfil de la demanda eléctrica diaria del SEIN de los años 2010; 2011 y hasta agosto del 2012.

Los datos históricos de los años 2010 y 2011 servirán para realizar el entrenamiento de la red neuronal, y los datos del 2012 permitirán realizar la comparación de la data histórica con los resultados de la proyección de la demanda eléctrica.

1.7 Fuentes de información e instrumentos usados

La información histórica de los perfiles de demanda eléctrica del SEIN, fueron obtenidos de la página web del Comité de Operación Económica del Sistema Interconectado Nacional (COES).

1.8 Organización de la tesis

La tesis está organizada en seis capítulos, los cuales se resumen a continuación.

El capítulo 1, detalla el objetivo, los alcances y la organización.

El capítulo 2, detalla las características de la demanda eléctrica a corto plazo, los horizontes de proyección, su importancia en la programación del despacho de un sistema de potencia, los factores que influyen en la proyección, los métodos tradicionales de proyección, así como las técnicas usadas.

El capítulo 3, describe los conceptos teóricos de las Redes Neuronales Artificiales, tales como las neuronas biológicas, su modelamiento computacional, su estructura, arquitectura, y el proceso de entrenamiento y aprendizaje.

El capítulo 4, describe el caso de estudio, aplicado al SEIN considerando como una demanda nodal, para realizar la proyección de la demanda. Se detalla el proceso para la elaboración de la base de datos. Se describe el modelo de red neuronal a utilizar, su arquitectura y la medida de precisión del pronóstico.

El capítulo 5, muestra los resultados obtenidos en las simulaciones. Se detalla el software de simulación, el diseño y proceso de entrenamiento de la red neuronal. Los resultados se validan de acuerdo al indicador de desviación porcentual de error MAPE, asimismo se hace la comparación de los resultados y se cuantifica el beneficio obtenido para algunos días del mes de mayo del 2012.

CAPITULO II

MARCO TEORICO: DEMANDA ELÉCTRICA A CORTO PLAZO

2.1 Horizontes de proyección de la demanda eléctrica

Los niveles de planificación de la operación y ejecución de un sistema eléctrico de potencia, definen los horizontes para la proyección de la demanda eléctrica, el cual es dividido en periodos de largo plazo, mediano plazo y corto plazo. Adicionalmente también podemos definir un periodo de cortísimo plazo [2] [10]. El detalle de los horizontes de la proyección de la demanda eléctrica se describe a continuación:

2.1.1 Proyección de la demanda eléctrica a largo plazo

Consiste en proyectar la demanda eléctrica en un horizonte de un año hacia adelante, para la planificación del sistema de potencia. Esta proyección es básica para elaborar el programa a largo plazo. Este programa permitirá determinar la ubicación y tamaño de los futuros centros de generación y transmisión del sistema de potencia. La proyección de demanda eléctrica a largo plazo puede ser usado como un punto de referencia para las futuras inversiones en el sector energético del país [2] [10].

La proyección de la demanda eléctrica a largo plazo, también permite analizar el nivel de consumo de energía eléctrica de un sistema de potencia, esta información complementaria, permite programar la operación de las centrales térmicas de modo que minimice sus costos de producción [2] [10].

2.1.2 Proyección de la demanda eléctrica a mediano plazo

Consiste en proyectar la demanda eléctrica en un horizonte de un mes hasta un año, este tipo de proyección sirve de información básica para determinar la descarga de los embalses, reservorios o lagunas de los centros de generación y la elaboración de los programas de mantenimiento de las instalaciones de generación y transmisión del sistema eléctrico. Esta programación permite determinar el suministro de energía eléctrica entre empresas integrantes del sistema de potencia, con el fin de satisfacer la demanda prevista. También permite establecer los costos operativos a mediano plazo. Esta previsión también es importante para la planificación de la programación del combustible que utilizan las centrales de generación térmica [2] [10].

2.1.3 Proyección de la demanda eléctrica a corto plazo

Consiste en proyectar la demanda eléctrica en un horizonte desde un día hasta un mes. La proyección de la demanda eléctrica en este horizonte es de vital importancia para los diferentes procesos que realizan los operadores de los centros de control de un sistema eléctrico, tales como: el despacho económico, la programación de transferencia de energía y el control de la operación en tiempo real.

La proyección de la demanda eléctrica diaria que realiza el operador de un sistema de potencia, normalmente está disponible a más tardar hasta las 10:00 am para su aplicación del siguiente día. Los errores significativos en este pronóstico pueden resultar en una programación excesivamente conservadora o en una programación excesivamente riesgosa causando graves consecuencias económicas.

2.1.4 Proyección de la demanda eléctrica a cortísimo plazo

La proyección de demanda eléctrica a cortísimo plazo, es también llamado proyección horaria, normalmente se realiza en un horizonte de 3 hasta 24 horas. Conocer la proyección de la demanda eléctrica cada media hora, es importante para la operación en tiempo real de los sistemas eléctricos de potencia, porque permite al operador del centro de control, tomar decisiones sobre la operación de cada generador en tiempo real.

En todo sistema eléctrico de potencia, se presentan imprevistos no pronosticados en corto plazo como son las fallas o variaciones de demandas eléctricas de usuarios libres por mantenimiento correctivos, los cuales incrementan las desviaciones de la proyección de la demanda. En un sistema eléctrico estos imprevistos que alteran la demanda eléctrica en cortísimo plazo, obligan al operador del sistema eléctrico de potencia a realizar redespachos o reprogramas de generación, para suplir el desbalance entre la demanda proyectada y la demanda real.

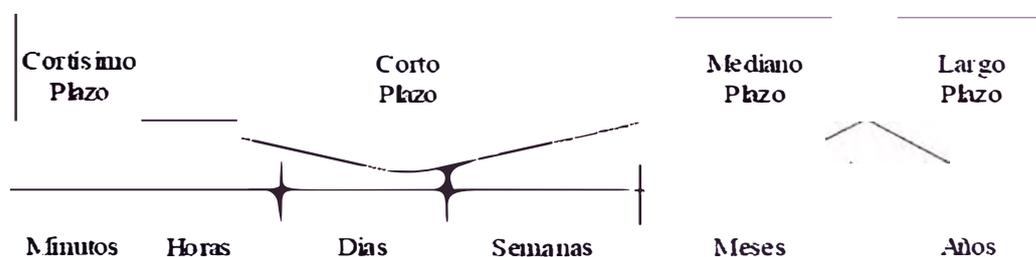


Fig. N° 2.1 Diagrama de intervalo de tiempo vs proyección de la demanda
(Fuente: Instituto de investigación FIEE-UNI)

En el sistema eléctrico peruano, este tipo de proyecciones se realiza cuando las desviaciones porcentuales entre la demanda proyectada y la demanda real es superior al 3% por más de 3 horas. Esta proyección se realiza con la finalidad de elaborar el reprograma diario por mayor o menor demanda en una respectiva área operativa del sistema eléctrico.

2.2 Importancia de la proyección de la demanda eléctrica a corto plazo

Para las empresas transmisoras y distribuidoras, la proyección de la demanda eléctrica a corto plazo, permite conocer de manera preliminar los flujos de potencia en las redes eléctricas de transmisión con la finalidad de identificar las instalaciones de potencia más críticas que podrían provocar fallas relevantes en el sistema eléctrico.

Para los generadores, la proyección eléctrica a corto plazo, permite definir la cantidad de energía eléctrica a generar (oferta de generación) para el día siguiente y programar la operación de sus unidades de generación.

Para el operador del sistema eléctrico, la proyección de la demanda a corto plazo, es el insumo esencial para la elaboración del programa diario de operación y es la base sobre la cual se elabora dicho programa operativo.

Una aplicación primaria de la proyección a corto plazo, es proporcionar información para la programación diaria, la cual definiría la operación de las centrales de generación más económicas para el sistema eléctrico, que cumplan con las exigencias de confiabilidad, restricciones operativas y ambientales, entre otras.

Una segunda aplicación de la proyección a corto plazo, es prever la seguridad del sistema eléctrico. La previsión de la demanda es importante para el análisis de las contingencias, que permitirá la detección de futuras condiciones sobre las cuales el sistema se vuelva vulnerable. Esta información permite al coordinador del sistema preparar la acción correctiva necesaria para operar el sistema de forma más segura.

Un mal pronóstico de la demanda a corto plazo incrementa las desviaciones entre la demanda proyectada y la demanda real, lo cual obliga al operador del sistema eléctrico a realizar reprogramas o redespachos con la finalidad de reducir las desviaciones. Así mismo puede provocar órdenes inexactas en las decisiones del operador de tiempo real, reduciendo la seguridad del sistema eléctrico.

2.3 Proyección de la demanda eléctrica a corto plazo del SEIN

La proyección de la demanda eléctrica a corto plazo del SEIN, lo realiza el COES, de acuerdo al procedimiento N° 3, denominado "Pronóstico de la demanda a corto plazo del Sistema Eléctrico Interconectado Nacional". Este procedimiento determina la metodología

de la proyección de la demanda eléctrica diaria, el cual es calculado por áreas operativas (Norte, Centro y Sur) y en su totalidad.

2.3.1 Consideraciones del pronóstico de la demanda del SEIN

La proyección de la demanda eléctrica diaria del sistema eléctrico interconectado nacional, se realiza tomando en cuenta las siguientes consideraciones:

- La proyección se elabora de acuerdo al procedimiento técnico N° 3 del COES, actualizado en la fecha 15.01.11.
- La proyección de la demanda eléctrica diaria, estima 48 valores de demanda cada media hora, agrupados en tres periodos, los cuales son: Periodo Base de 00:00 a 08:00 horas; Periodo Media de 08:00 a 18:00 horas y Periodo Punta de 18:00 a 24:00 horas.
- Los usuarios libres del SEIN, proporcionan información diaria de su demanda prevista para el día siguiente, en sus respectivas barras de conexión.
- Las empresas distribuidoras del SEIN, proporcionan información de su demanda prevista para el día siguiente, en sus respectivas barras de conexión, cuyos valores de tensiones son mayores o iguales a 60 kV.
- Las empresas distribuidoras y usuarios libres, envían las previsiones de sus respectivas demandas eléctricas en intervalos de 30 minutos antes de las 08:00 horas de cada día, la cual sirve como información base para realizar la proyección de la demanda eléctrica del siguiente día.
- La proyección de la demanda eléctrica se realiza por las áreas operativas del SEIN (Norte, Centro y Sur). El pronóstico diario final será la suma de todas las áreas.
- La demanda histórica de cada área operativa del SEIN, se determina sumando la generación ejecutada en dicha área con los flujos que interconectan el área operativa con el resto del sistema.
- La demanda histórica de un usuario libre será su demanda ejecutada en los días anteriores.
- La metodología usada para la proyección de la demanda eléctrica, desarrolla 2 métodos: para un día ordinario (no feriado) y para un día feriado.

A continuación se detalla la metodología de la proyección de la demanda eléctrica diaria que usa el COES, para los días ordinarios y días feriados.

2.3.2 Metodología del pronóstico diario para días ordinarios

a. Selección de los datos históricos

Se elige el día de la semana para el cual se realizará el pronóstico de la demanda. Luego se procede agrupar aquellos días que tienen características de perfiles de demanda eléctrica similar. De los cuales los días martes, miércoles, jueves y viernes forman un solo grupo, los demás días tienen perfiles de demanda eléctrica diferente.

La finalidad de agrupar los días de la semana de acuerdo a la característica similar de perfiles de la demanda eléctrica, es para formar una base de datos que permita de una manera resumida y sencilla, relacionar la proyección del día en evaluación con la información histórica agrupada en una base de datos.

Los días de la semana se agrupan de acuerdo al perfil de su demanda eléctrica, y se forman una base de datos, tal como se muestra en la tabla 2.1

Tabla N° 2.1 Formación de la base de datos de acuerdo a la característica similar del día de la semana

(Fuente: Comité de operación económica del sistema interconectado Nacional)

Día de pronóstico	Base de Datos
Lunes	5 últimos lunes no feriados
Martes, Miércoles Jueves y Viernes	5 últimos días que no incluye, lunes, sábados, domingos o feriados.
Sábado	5 últimos sábados no feriados
Domingo	5 últimos domingos no feriados

b. Depuración de los datos históricos

- La demanda histórica de cada área operativa, se separa en demanda vegetativa y demanda industrial, obteniéndose dos matrices de demanda histórica. Los usuarios **industriales** con una demanda menor a 10 MW se consideran como parte de la demanda vegetativa.
- Se calcula la mediana de cada fila de la matriz de la demanda vegetativa (mediana de los cinco últimos datos históricos del día en proyección).
- Se calcula el **porcentaje de diferencia** de cada matriz de demanda vegetativa con respecto a la mediana obtenida. Si el porcentaje de la diferencia es menor en 5%, el dato será considerado válido, caso contrario será reemplazado por el promedio de todos los datos válidos.

c. Cálculo del perfil diario de la demanda

- De la matriz de demandas vegetativas históricas depuradas, para cada día se calcula el valor mínimo de la demanda en periodo base y los valores máximos de demanda en periodos de media y punta.
- Para cada día, se divide cada uno de los valores de demanda con su correspondiente valor mínimo o máximo obtenido del punto anterior. Así se obtiene una matriz por unidad de demanda vegetativa.
- Para cada fila de la matriz por unidad vegetativa, se calcula la mediana de los cinco datos históricos. Así se obtiene la proyección preliminar de la demanda vegetativa por unidad.
- De la proyección preliminar, se calcula el valor mínimo para el periodo base y los valores máximos para los periodos medios y punta. Luego se divide cada valor con su correspondiente valor mínimo y máximo obtenido anteriormente. Así obtenemos el perfil de demanda vegetativa por unidad.

d. Pronóstico de las potencias mínima y máximas

Con los cinco valores de demanda obtenidos para los periodos base (demanda mínima), periodo media (demanda máxima) y periodo punta (demanda máxima), se pronostica un sexto valor asumiendo una tendencia lineal en la evolución de la demanda, para lo cual se utiliza el método de los mínimos cuadrados.

e. Pronóstico de la demanda

- El pronóstico de la demanda vegetativa se obtiene multiplicando a todos los valores de demanda vegetativa por unidad, por las magnitudes de potencia para el periodo base, media y punta.
- La demanda eléctrica pronosticada del SEIN, será la suma de la demanda vegetativa y la demanda industrial.



Fig. N° 2.2 Secuencia de proyección de la demanda eléctrica diaria del SEIN
(Fuente: Comité de operación económica del sistema interconectado Nacional)

2.3.3 Metodología del pronóstico diario para días feriados

- Se identifica el día feriado de la semana, para su proyección. Se busca información histórica de un día feriado pasado, que corresponda al día feriado a proyectarse. La semejanza se establecerá cuando ambos feriados se presenten el mismo día de la semana.
- Determinado el día feriado semejante, se separa su demanda total en demanda vegetativa y demanda industrial.
- Para el día feriado semejante (pasado) y para el día feriado a proyectar (futuro), se seleccionan las demandas vegetativas de un día previo que no haya sido feriado. Luego se procede a dividir las demandas vegetativas (futuro/pasado) y se obtienen los factores de proporción.
- Para obtener el pronóstico de la demanda vegetativa del día feriado, se multiplica los factores de proporción anteriormente obtenidos, por la demanda vegetativa del día feriado semejante (pasado).
- El pronóstico de la demanda para el día feriado será la suma de la demanda vegetativa y la demanda industrial.

2.4 Análisis de la metodología de la proyección diaria del SEIN

Se puede afirmar que la metodología que usa el COES en la proyección de la demanda eléctrica diaria posee limitaciones, algunas de las cuales se detallan a continuación:

- La metodología no detalla cual sería la proyección de la demanda eléctrica para los días feriados, cuyos días semejantes (pasados) no coincidan con el mismo periodo de estacionalidad del día en evaluación, o sea no se plantea cual sería la metodología para los casos que no se dispone de información histórica para la proyección de ese día feriado.
- La metodología no considera el patrón de tendencia representativo del perfil diario, debido a la poca información histórica que toma como base de datos (5 días). Este patrón se distorsiona principalmente en los cambios de estacionalidad, cuando la característica del perfil de demanda varía por el efecto del cambio de la temperatura.
- Esta metodología no excluye como fuente de información histórica a los días con características de demanda atípica, Los días de perfil atípico se presentan por asuntos sociales, económicos, climatológicos, periodos de pesca, entre otros. Considerar días atípicos como fuente de información traerá como consecuencia una mala predicción de demanda.
- Para conocer la precisión de las proyecciones diarias que realiza el COES, se procede a calcular las desviaciones entre la demanda proyectada y la demanda real. La desviación promedio porcentual del día (MAPE) es el indicador sobre la cual se puede concluir si la proyección diaria es aceptable.
- Para conocer la precisión de la metodología de la proyección diaria de la demanda del SEIN que realiza el COES, se ha hecho un resumen de los indicadores MAPE diarios del año 2012, los cuales se detallan en la tabla N° 2.2.

Tabla N° 2.2 Resumen de indicadores diarias del SEIN del año 2012
(Fuente: Comité de operación económica del sistema interconectado Nacional)

RESUMEN INDICADORES	ENE	FEB	MAR	ABR	MAY	JUNIO	JULIO	AGO	SET	OCT	NOV	DIC
MAPE MAYOR DEL MES	2 99	3 50	2 99	6 03	8 12	4 90	3 63	2 27	3 54	5 38	3 13	5 20
MAPE MENOR DEL MES	0 86	0 73	0 68	0 75	0 64	0 84	0 67	0 85	0 93	0 28	0 83	0 86
MAPE DEL MES	1 71	1 60	1 68	1 91	1 92	1 74	1 74	1 41	1 55	1 60	1 56	1 85

En el cuadro N° 2.2 se puede apreciar que en los meses de enero, marzo, abril, mayo, junio, julio y diciembre se realizaron los mayores errores en la proyección de la demanda diaria.

- El resumen de la desviación porcentual del mes de enero del 2012, se muestra en la figura N° 2.3. El máximo valor del MAPE fue 2.99% que corresponde al día domingo 08 de enero. El MAPE del mes fue 1.71%. El MAPE de los días 26 y 28 de enero fueron 2.64% y 2.57% respectivamente.
- La proyección de la demanda del SEIN del día 08 de enero se muestra en la figura N° 2.4. Para este día se puede observar que se efectuó una mala proyección, ya que la demanda proyectada fue en promedio más de 150 MW en comparación a la demanda real, de 06:00 a 24:00 horas. La máxima desviación fue 297 MW (7.33%) y ocurrió a las 18:30 horas. Este error de proyección radica principalmente por la variación de la estacionalidad del mes de diciembre a enero, cuya base de datos está formada por días feriados y días atípicos.

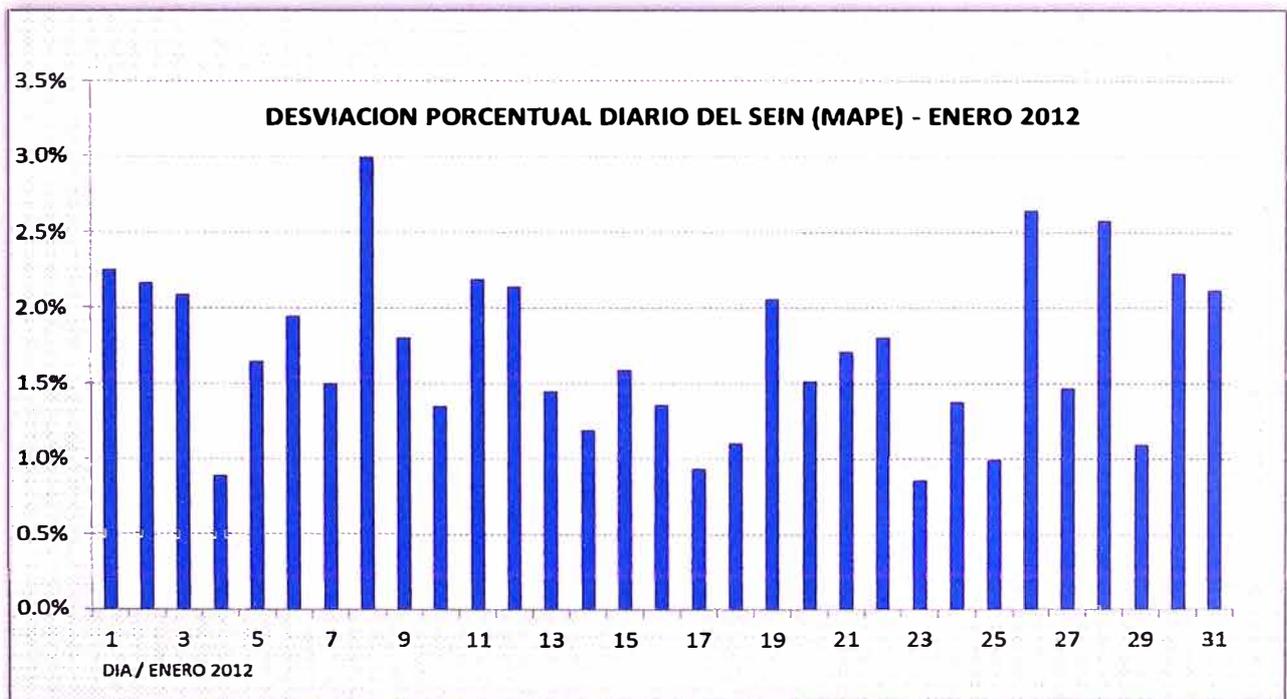


Fig. N° 2.3 MAPE de la proyección del SEIN de enero del 2012
(Fuente: elaboración propia)

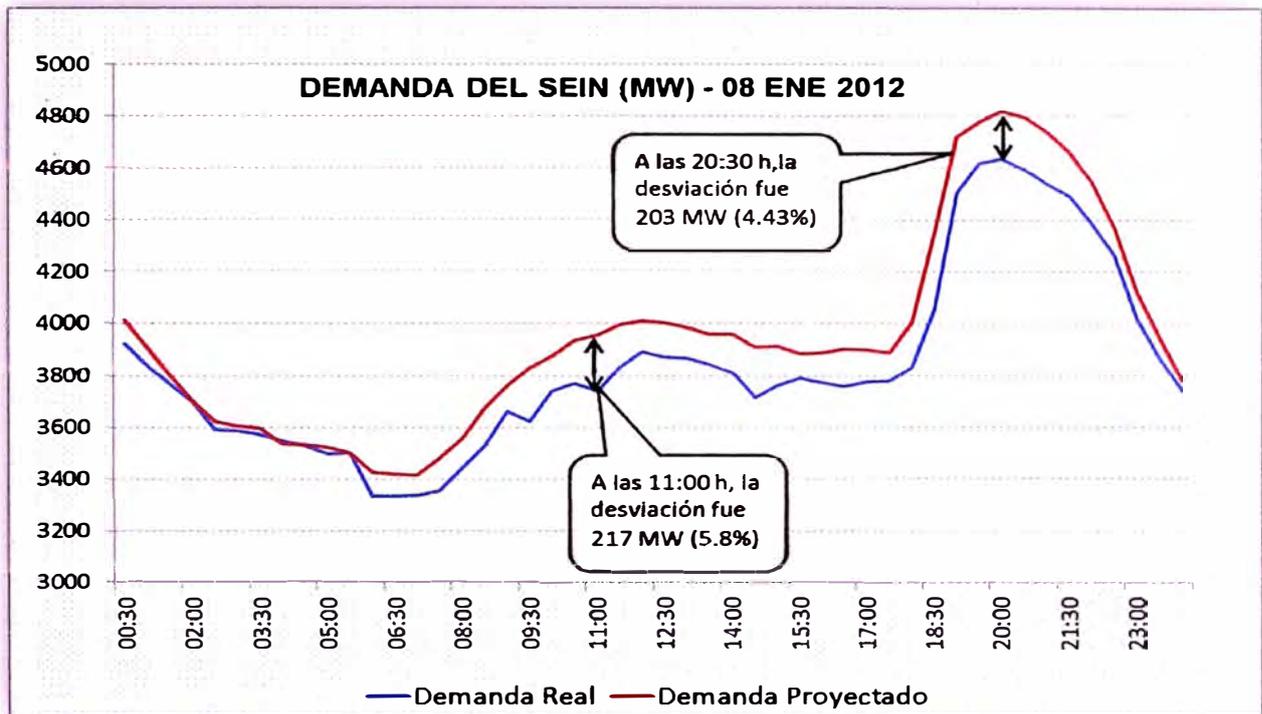


Fig. N° 2.4 Demanda del SEIN proyectado y real del día 08.ene.2012

(Fuente: elaboración propia)

- El resumen de la desviación porcentual del mes de julio del 2012, se muestra en la figura N° 2.5. El máximo valor del MAPE fue 3.63% que corresponde al día sábado 28 de julio. El MAPE del mes fue 1.74%. El MAPE del día 15 de julio fue 3.5%, esto debido principalmente al colapso del área norte del SEIN ocurrido a las 05:26 horas.
- La proyección de la demanda del SEIN del día 28 de julio se muestra en la figura N° 2.6. La proyección para ese día presentó altas desviaciones. A las 07:00 horas la desviación fue 262 MW (7.9%) y a las 10:00 horas fue 287 MW (7.78%). El error en la proyección fue por la falta de información (base de datos) para proyectar la demanda de un día sábado feriado coincidente con el 28 de julio.

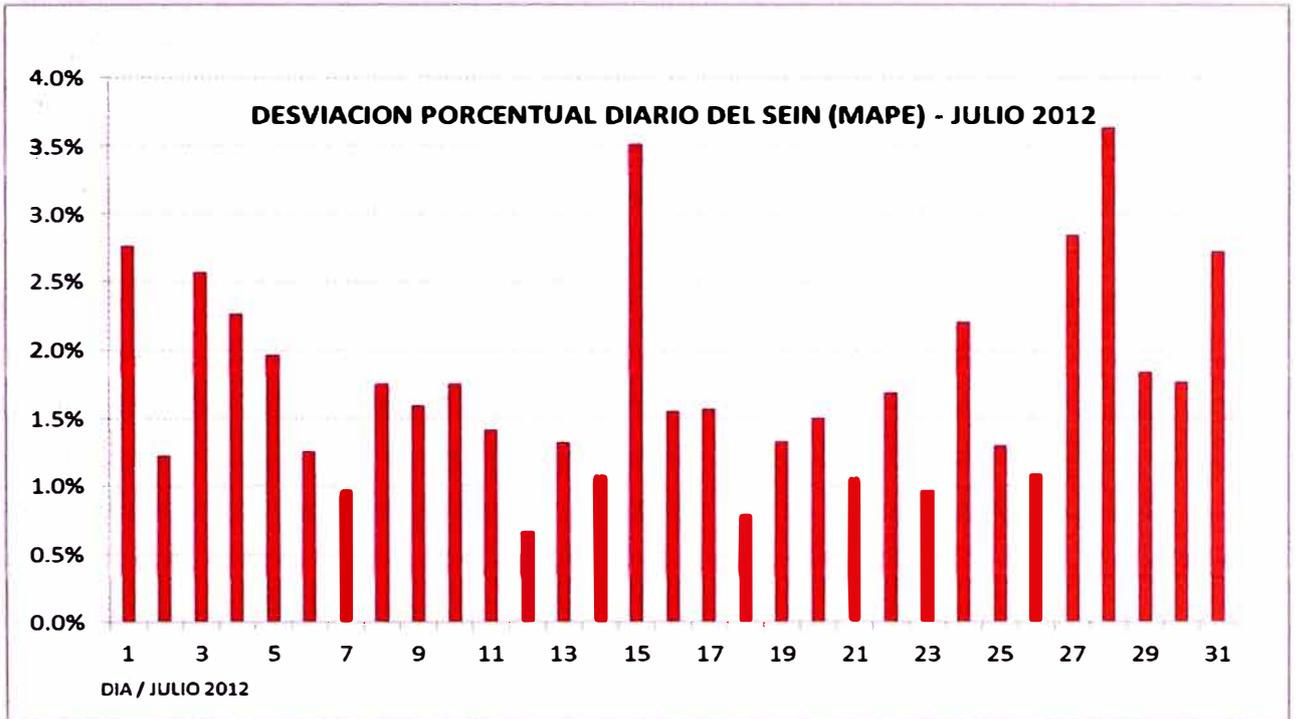


Fig. N° 2.5 MAPE de la proyección del SEIN de julio del 2012
(Fuente: elaboración propia)

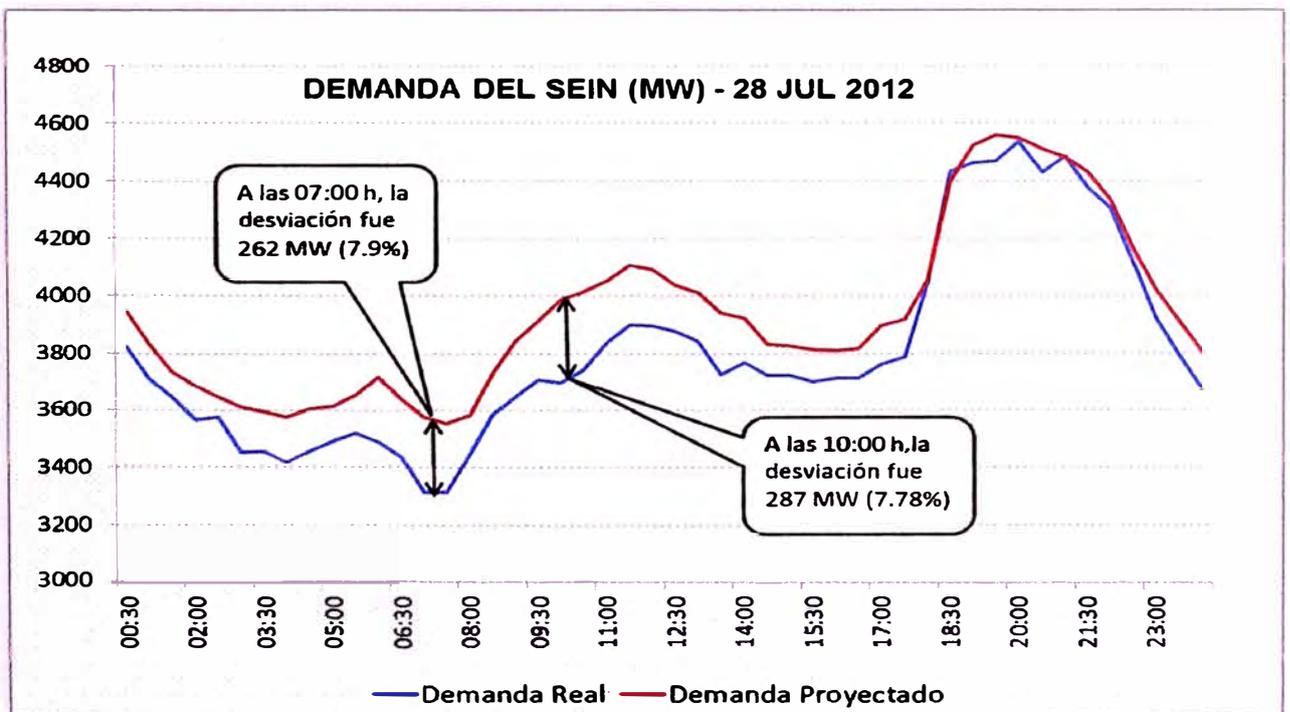


Fig. N° 2.6 Demanda del SEIN proyectado y real del día 28.jul.2012
(Fuente: elaboración propia)

- El resumen de la desviación porcentual del mes de diciembre del 2012, se muestra en la figura N° 2.7. El máximo valor del MAPE fue 5.20% que corresponde al día lunes 24 de diciembre. El MAPE del mes fue 1.85%. El MAPE del día 16 de diciembre fue 3.5%, esto debido principalmente al colapso del área sur del SEIN ocurrido a las 00:02 horas. El MAPE del día 29 de diciembre fue 3.73%,
- La proyección de la demanda del SEIN del día 29 de diciembre, se muestra en el gráfico N° 2.8. El perfil de la demanda proyectada y la demanda real es similar, con un mismo patrón de tendencia, sin embargo la demanda proyectada es siempre mayor las 24 horas. La desviación promedia diaria es 159 MW (3.73%). Esta proyección fue mal estimado y no guarda relación con su metodología, ya que la base de datos (5 días sábados pasados) usado para esta proyección, no arrojan resultados similares a lo que se pronosticó.

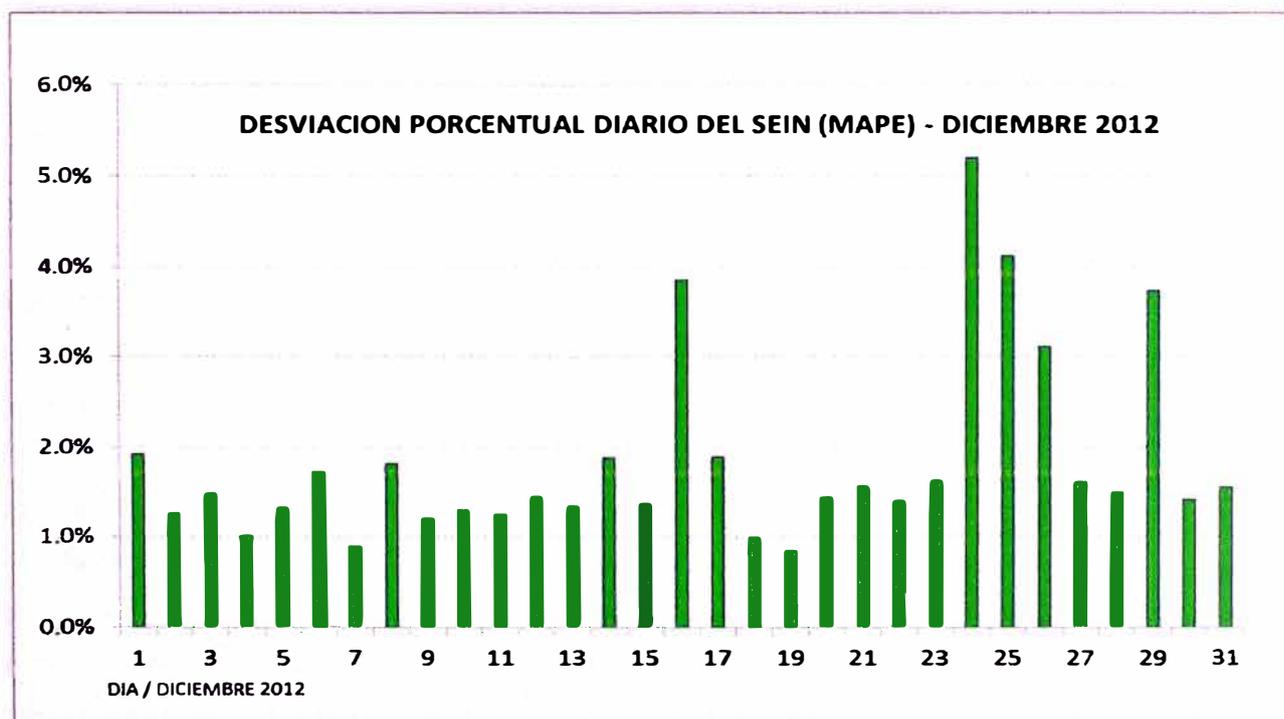


Fig. N° 2.7 MAPE de la proyección del SEIN de diciembre del 2012
(Fuente: elaboración propia)

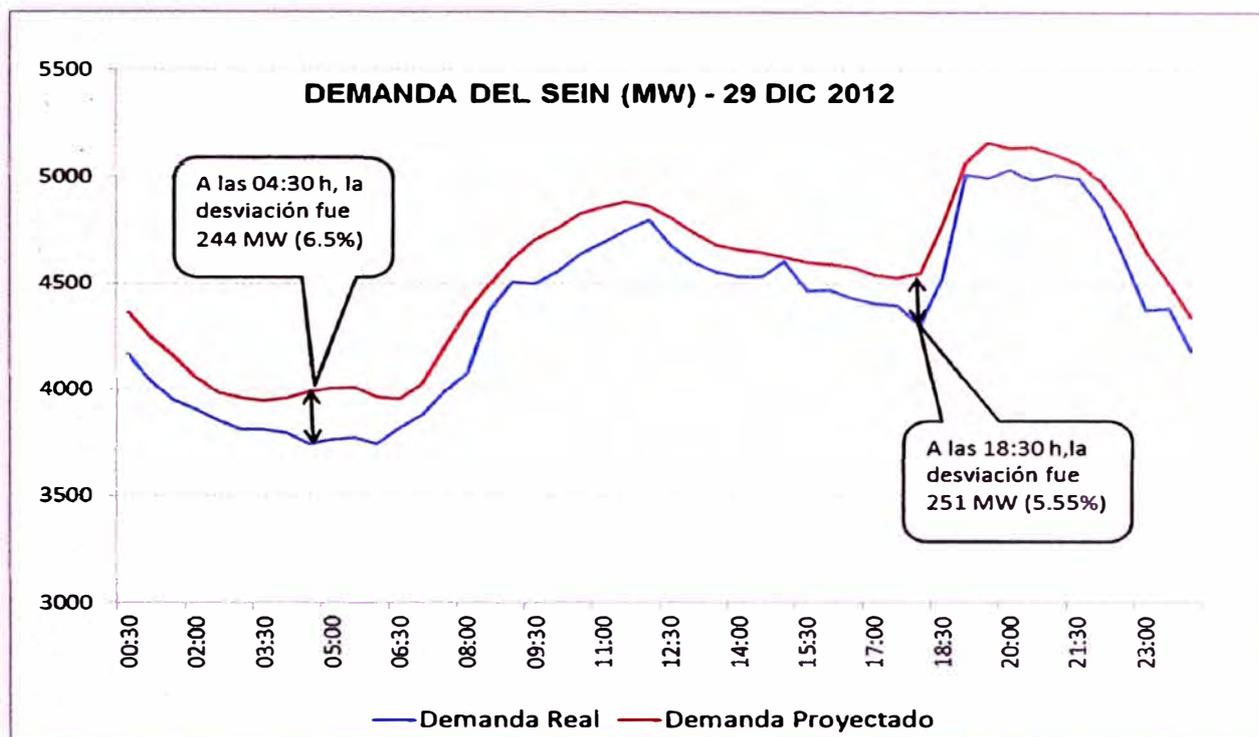


Fig. N° 2.8 Demanda del SEIN proyectado y real del día 29.dic.2012
(Fuente: elaboración propia)

En resumen el método tradicional de proyección de la demanda diaria del SEIN que utiliza el COES tiene más de 20 años de uso continuo. A través de esta metodología propuesta se deja en consideración otra alternativa de proyección para su evaluación y comparación de valores pronosticados.

2.5 Características de la demanda eléctrica del SEIN a corto plazo

De acuerdo a las características del perfil de la demanda eléctrica del SEIN, podemos agrupar los días de la semana en tres grupos: Un primer grupo para los días lunes, martes, miércoles, jueves y viernes, otro grupo para los días sábados y un tercer grupo para los días domingos. Los días festivos o declarados feriados por el estado peruano tienen diferentes características. En la figura 2.9 se muestra el perfil de demanda eléctrica para cada día de la semana.

Otra característica de relevancia que presenta el perfil de demanda eléctrica diaria del SEIN, es la estacionalidad. Si hacemos la comparación de los perfiles de demanda de un día para diferentes estaciones del año, se puede apreciar que el perfil de demanda varía, tal como se muestra en la figura N° 2.10.

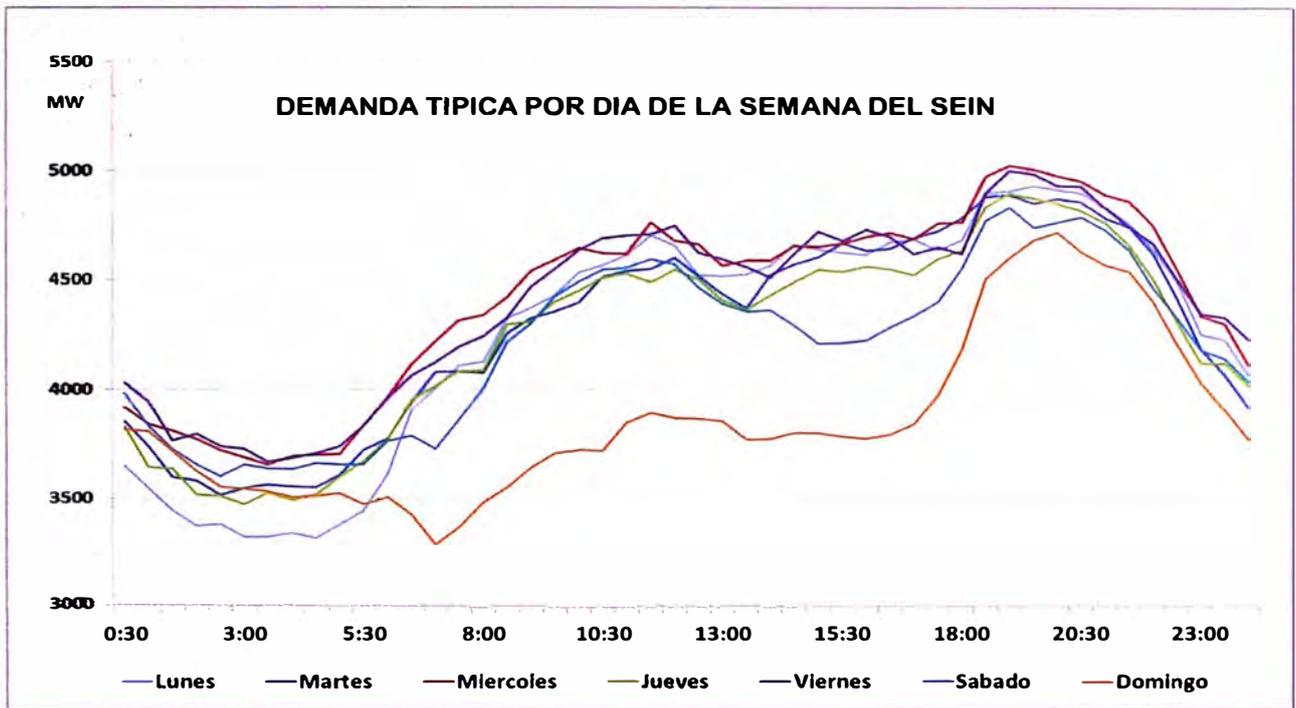


Fig. N° 2.9 Demanda eléctrica típica por días de la semana del SEIN
(Fuente: elaboración propia)

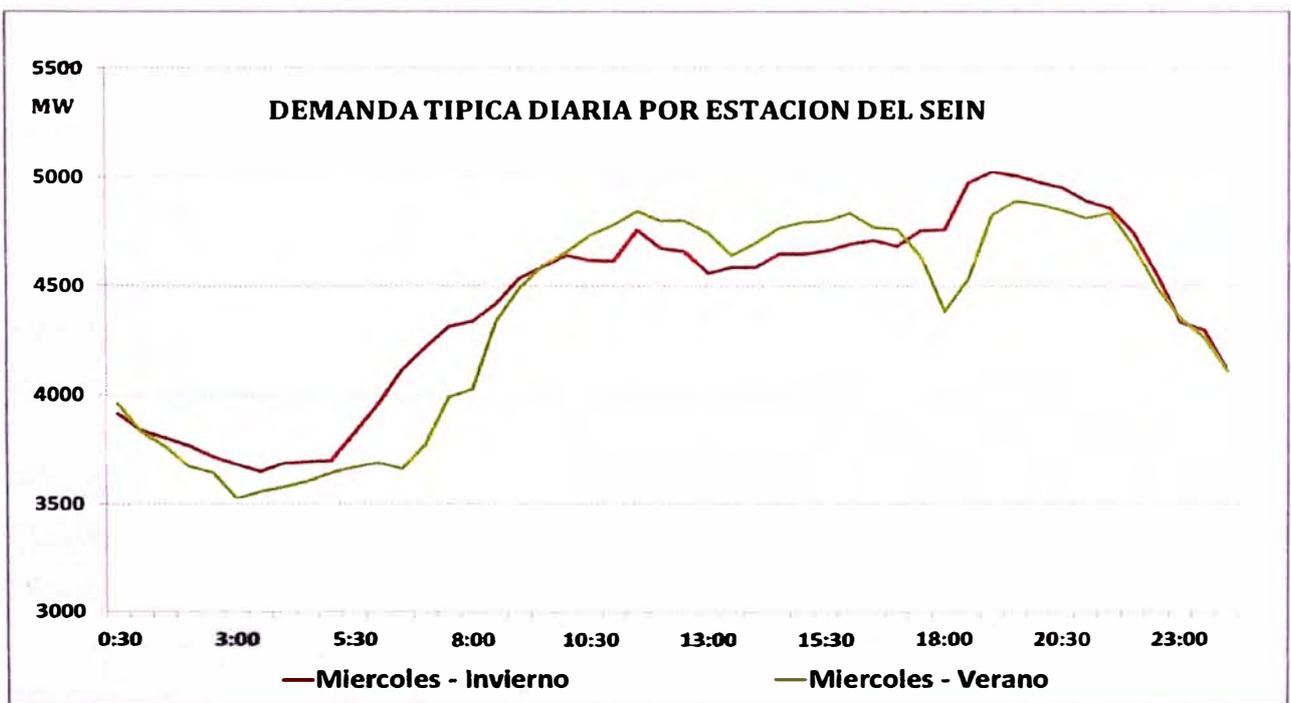


Fig. N° 2.10 Demanda eléctrica típica diaria del SEIN por estación del año
(Fuente: elaboración propia)

De estos dos últimos gráficos, podemos afirmar que la demanda eléctrica a corto plazo tiene una característica particular, que depende del ciclo general diario de cada semana y de la temperatura, ya que el comportamiento del perfil de un día de la semana es diferente para cada estación del año.

Se concluye que el consumo de electricidad varía de acuerdo a la estación del año, el día de la semana y la hora del día. Los mayores consumos se presentan durante los días laborales y en horas pico (12:00 y 19:00 horas). El mínimo consumo se presenta a las 03:00 horas, para los días laborales y días sábados, mientras que para los días domingos el mínimo consumo se presenta a las 07:00 horas.

2.6 Factores que influyen en la demanda eléctrica

La demanda eléctrica es influenciado por factores económicos, temporales, climáticos y disturbios aleatorios, los cuales se detallan a continuación:

2.6.1 Factores económicos

El sistema económico es un factor importante en el comportamiento de la demanda eléctrica. Es así que el consumo eléctrico depende del nivel de actividad industrial (pesquería, minería, agricultura, etc.). Las tendencias económicas tienen impactos significativos sobre el crecimiento o decrecimiento de la demanda eléctrica [2]. En el sistema peruano los periodos de pesca y veda tienen influencia directa en la demanda eléctrica. El periodo de pesca de la anchoveta para el uso industrial en el Perú dura aproximadamente entre cuatro y seis semanas al año.

2.6.2 Factores temporales

Se considera tres factores temporales principales que influyen en la demanda eléctrica las cuales son: las estaciones del año, los días cíclicos de la semana y los días festivos o feriados.

La demanda eléctrica del SEIN tiene implicancia directa con la estación del año, mostrándose con mayor variación entre las estaciones de verano e invierno. Una característica típica del sistema peruano es que la máxima demanda del día se inicia en el área sur, luego en el área centro y finalmente en el área norte.

Los días feriados tienen perfiles de demanda diferentes y muchas veces es difícil conocer como será el comportamiento de demanda del día festivo. Días y horas anteriores y posteriores al día festivo presentan un comportamiento diferente a un día típico normal. En el sistema peruano es muy común que se presenten estas variaciones.

El primero de enero, se presenta el menor consumo eléctrico del año en el sistema eléctrico del SEIN.

2.6.3 Factores climáticos

Las condiciones meteorológicas son responsables de variaciones significativas de la demanda eléctrica. Debido al uso de los equipos de aire acondicionado, equipos de calefacción, entre otros. En muchos sistemas de potencia la temperatura es una variable climática que puede incrementar las desviaciones al pronóstico de la demanda, lo cual obliga al operador del sistema a realizar reprogramas o redespachos diarios [2] [3].

En el sistema eléctrico peruano se presentan cambios rápidos y significativos del clima, principalmente en los periodos de cambio de estacionalidad. La presencia del fenómeno del niño en la estación de verano trae muchas consecuencias nefastas en lo económico y social del área norte del SEIN, por lo cual la demanda baja considerablemente en esta área operativa.

2.6.4 Disturbios aleatorios

Podemos agrupar en esta clasificación a una variedad de eventos aleatorios no previstos que causan variaciones en la demanda eléctrica. Por ejemplo los movimientos sísmicos, o los problemas sociales que se suelen presentar en contra de la explotación de los minerales, pueden provocar la paralización total o parcial de la producción de una empresa minera, en el sistema peruano se presentan estas situaciones.

2.7 Modelos de proyección de la demanda eléctrica a corto plazo

A continuación detallamos algunos modelos de proyección de demanda eléctrica a corto plazo:

2.7.1 Modelo pico de carga

Los modelos de pico de carga son usados como funciones de variables climáticas, las cuales adoptan la forma: *Pico de Carga = Carga Base + Componente Dependiente*. En donde la carga base es insensible al clima y la componente dependiente es sensible a las variables del clima [2] [4].

Como variables climáticas se incluye a la temperatura con datos históricos o datos previstos. También son considerados como variables del modelo la intensidad luminosa, velocidad del viento, precipitaciones, entre otros. La componente dependiente, puede ser empíricamente calculada y puede ser lineal o no lineal.

La ventaja de este modelo se encuentra en su simplicidad, lo cual requiere poca cantidad de datos, y su desventaja es que no depende del tiempo. Por lo cual estos modelos son esencialmente estáticos. Los fenómenos dinámicos como correlación entre periodos no pueden ser previstos.

2.7.2 Modelo de perfil de carga

El modelo de perfil de carga describe la demanda de manera de series de tiempo discreto sobre el intervalo de la previsión, la toma de muestras de la carga sobre intervalos de tiempo es usualmente una hora o media hora, mientras que la cantidad medida es generalmente la energía consumida sobre el intervalo en MWH [2] [4].

Básicamente, existen 2 tipos de modelo de perfil de carga: modelos del tiempo del día y modelos dinámicos. La combinación de estos 2 modelos también es posible.

a. Modelo tiempo de día:

El modelo de tiempo del día que se muestra en la ecuación (2.1) define la carga $f(t)$ en cada toma de muestra discreta de tiempo t del periodo de la previsión de duración T en una serie de tiempo.

$$\{ f(t), t = 1, 2, 3, \dots, T \} \quad (2.1)$$

En su forma más simple, el modelo de tiempo del día almacena T valores de carga basados sobre el comportamiento de cargas previamente observadas.

b. Modelo dinámico

Estos modelos, aparte de considerar a la carga como función del tiempo, consideran su comportamiento más reciente, condiciones meteorológicas y aportaciones al azar. Estos modelos se pueden dividir en: El promedio móvil auto-regresivo o modelos ARIMA y los modelos de espacio de estado.

2.8 Técnicas de pronóstico de la demanda eléctrica

Las técnicas usadas para la proyección de la demanda eléctrica a mediano, corto y cortísimo plazo, las podemos clasificar en regresión lineal múltiple, alisamiento exponencial, series temporales estocásticas, método de espacio de estado, Redes Neuronales Artificiales y modelos econométricos, los cuales se detallan a continuación:

2.8.1 Regresión lineal múltiple

Se expresan en términos de variables exógenas que pueden ser climáticas (temperatura, luminosidad, entre otros) y no climáticas (cantidad de consumidores, actividad económica, entre otros) [2] [4]. La expresión utilizada para representar la técnica está dado por:

$$Y(t) = a_0 + a_1x_1(t) + a_2x_2(t) + \dots\dots a_nx_n(t) + \mathcal{E}(t) \quad (2.2)$$

En donde:

$Y(t)$ = Carga eléctrica en el instante "t"

$x_i(t)$ = Variables exógenas

$\mathcal{E}(t)$ = Variables aleatoria con promedio cero y varianza constante

Las variables explicativas x_i son identificados a través de un análisis de correlación entre las variables y la carga. Cuando el número de variables candidatas es muy grande, se usa una técnica estadística conocida como Stepwise para seleccionar entre todas las variables las significativas. La estimación de los coeficientes es realizado generalmente por el método de mínimos cuadrados o pruebas estadísticas. Muchos trabajos utilizan esta técnica para la proyección de la demanda en un horizonte de mediano plazo.

2.8.2 Alisamiento exponencial

Estos modelos utilizan un conjunto de funciones temporales $f(t)$ para modelar la carga $Y(t)$ en un instante "t", como:

$$Y(t) = \beta^T(t).f(t) + \mathcal{E}(t) \quad (2.3)$$

En donde:

$\beta(t)$ Vector de coeficientes, estimado con el método de mínimos cuadrados ponderados para N intervalos.

"T" Operador transpuesto

$\mathcal{E}(t)$ Variable aleatoria [2] [4]

2.8.3 Series temporales estocásticas

Este modelo es el más usado en la proyección de la demanda de corto plazo. Esta técnica es modelado como la salida de un filtro lineal cuya entrada es una serie temporal estocástica $a(t)$ con promedio cero y varianza constante desconocida, la cual usualmente

es llamado ruido blanco tal como se muestra en la figura 2.11. Los modelos ajustados son conocidos como modelos ARIMA (p,d,q) (autoregresivo integrado de medias móviles) [2] [4].

En caso más general, suponiendo que la carga posee componentes estacionales estocásticas, esta técnica es ajustada con los modelos SARIMA (p,d,q)(P,D,Q)S, donde (p,P) determinan las ordenes autoregresivas y autoregresivas estacionales del modelo y (q,Q) determinan las ordenes de medias móviles y medias móviles estacionales en el periodo S.

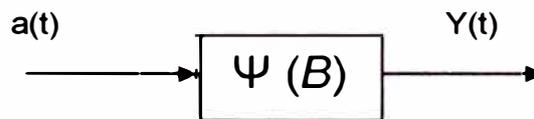


Fig. N° 2.11 Modelo de series temporales

(Fuente: Previsao de Carga no Período de Demanda de Ponta Utilizando Redes Neurais Artificiais)

En donde:

$\Psi(B)$ Función de transferencia de filtro

B Operador de retardo

Se cumple: $BY(t) = Y(t-1)$

En el caso más general la relación entre la carga $Y(t)$ y el ruido está dado por:

$$\varnothing(B) \cdot \Phi(B^S) \cdot \bar{\nabla}^d \cdot \bar{\nabla}_S^D \cdot Y(t) = \theta(B) \cdot \Theta(B^S) \cdot a(t) \quad (2.4)$$

En donde:

$\varnothing(B) \cdot \Phi(B^S)$ son polinomios de grado p y P respectivamente

$\theta(B) \cdot \Theta(B^S)$ son polinomios de grado q y Q respectivamente

$\bar{\nabla}$ es un operador diferente de orden definido por $\bar{\nabla}Y(t) = Y(t) - Y(t-1)$.

La diferencia de orden "d" está definida como $\bar{\nabla}^d Y(t) = (1-B)^d \cdot Y(t)$ donde "d" es un numero entero cualquiera.

Una definición similar es $\bar{v}_s^D = (1 - B^s)^D$. Tales operadores son necesarios cuando una serie $Y(t)$ es no estacionaria o tiene tendencial estacional.

2.8.4 Método de espacio de estado

Este modelo está representado con coeficientes variables. Es modelado como una variable de estado de un sistema lineal descrito por dos sistemas de ecuaciones. Son las ecuaciones de estado y de observación [2] [4]. tales como:

$$Y(t+1) = A(t)X(t) + W(t) \quad (2.5)$$

$$Y(t) = H(t)X(t) + V(t) \quad (2.6)$$

En donde:

$X(t)$ es un vector de estado

$A(t)$ es una matriz de transición de estado

$W(t)$ y $V(t)$ son ruidos blancos con matrices de covarianza $Q(t)$ y $R(t)$ respectivamente conocidas.

$Y(t)$ es la carga medida en el instante "t"

$H(t)$ es una matriz que relaciona $X(t)$ con $Y(t)$.

Una estimación a posteriori del proceso está dado por:

$$X(t/t) = X(t/t-1) + K(t) [Y(t) - H(t) X(t/t-1)] \quad (2.7)$$

En donde:

$$K(t) = P(t/t-1)H(t)^T [H(t)P(t/t-1)H(t)^T + R(t)^{-1}] \quad (2.8)$$

$$P(t/t) = [1 - K(t)H(t)] P(t/t-1) \quad (2.9)$$

Una previsión a priori está dado por:

$$X(t+1/t) = A(t)X(t/t) \quad (2.10)$$

$$P(t+1/t) = A(t)P(t/t)Q(t)^T + Q(t) \quad (2.11)$$

Donde $K(t)$ es denominado ganancia del filtro de Kalman y $P(t/t-1)$ es una matriz de covarianza del vector error: $\varepsilon(t/t-1) = X(t) - X(t/t-1)$

Como principal dificultad en la aplicación de este método es identificar el sistema, en particular las matrices de covarianza $Q(t)$ y $R(t)$.

2.8.5 Modelo econométrico

De acuerdo a la estructura matemática, los modelos econométricos se pueden clasificar en: modelos econométricos de regresión uniecuacional y de regresión biecuacional (con ecuación de cointegración o de largo plazo y ecuación de corrección de error o de corto plazo)

La especificación funcional del modelo se inicia considerando que las variables socioeconómicas explicativas de las ventas vegetativas pueden ser: El producto bruto interno, la población y la tarifa eléctrica. Se analiza la estacionalidad de las variables participantes, en sus formas naturales (en niveles) o transformadas, mediante las pruebas de Dickey–Fuller Aumentado, ADF.

Se evalúa la Colinealidad en los regresores o variables explicativas. El análisis de colinealidad permite determinar si cada regresor, inicialmente considerado, guarda independencia lineal o no con los demás regresores. El requisito fundamental para que un modelo sea apropiado para predicciones es la independencia lineal entre variables explicativas.

El método de los mínimos cuadrados, o también se conoce como teoría de la regresión lineal es uno de los modelos más utilizados. Se parte de representar las relaciones entre una variable económica endógena y una o más variables exógenas de forma lineal, de la siguiente manera:

$$Y = a_1 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots \beta_n X_n \quad (2.12)$$

"Y" es la variable endógena, cuyo valor es determinado por las exógenas X_1 , hasta X_n . Cuales son las variables elegidas depende de la teoría económica que se tenga en mente, y también de análisis estadísticos y económicos previos. El objetivo buscado sería obtener los valores de los parámetros desde a_1 hasta β_n .

CAPITULO III

APLICACIÓN DE REDES NEURONALES ARTIFICIALES PARA LA PROYECCION DE LA DEMANDA ELECTRICA A CORTO PLAZO

3.1 Introducción

Las Redes Neuronales Artificiales (RNA) son modelos matemáticos que intentan reproducir el funcionamiento del sistema de las redes neuronales biológicas. Como todo modelo, realizan una simplificación del sistema real que simulan y toman las características principales del mismo para la resolución de una tarea determinada.

En este capítulo se define el comportamiento de una Red Neuronal Artificial, su arquitectura, el tipo de entrenamiento que se realiza y la forma como se selecciona la red con menores errores.

Con las Redes Neuronales Artificiales (RNA) se busca la solución de problemas complejos, no como una secuencia de pasos, sino como la evolución de unos sistemas de computación inspirados en el cerebro humano, y dotados por tanto de cierta "inteligencia", los cuales no son sino la combinación de elementos simples de procesos interconectados, que operan en forma paralela en varios estilos, y consiguen resolver problemas relacionados con el reconocimiento de formas o patrones, predicción, codificación, control y optimización entre otras aplicaciones.

3.2 Red neuronal biológica

El cerebro es el elemento principal del sistema nervioso humano y está compuesto por un tipo especial de célula llamada neurona. Una neurona es una célula viva y como tal posee todos los elementos comunes de las células biológicas. A su vez, las neuronas tienen características propias que le permiten comunicarse entre ellas, lo que las diferencia del resto de las células biológicas [5] [6] [7] [11].

En la figura 3.1 se observa que la neurona biológica está compuesta por un cuerpo celular o soma, del cual se desprende un árbol de ramificaciones llamado árbol dendrítico, compuesto por las dendritas. Del soma también parte una fibra tubular, llamada axón, el cual suele ramificarse cerca de su extremo. Las dendritas actúan como un canal de entrada de señales provenientes desde el exterior hacia la neurona, mientras que el axón actúa como un canal de salida. El espacio entre dos neuronas vecinas se

denomina sinapsis. En el córtex cerebral se observa una organización horizontal en capas, así como también una organización vertical en columnas de neuronas.

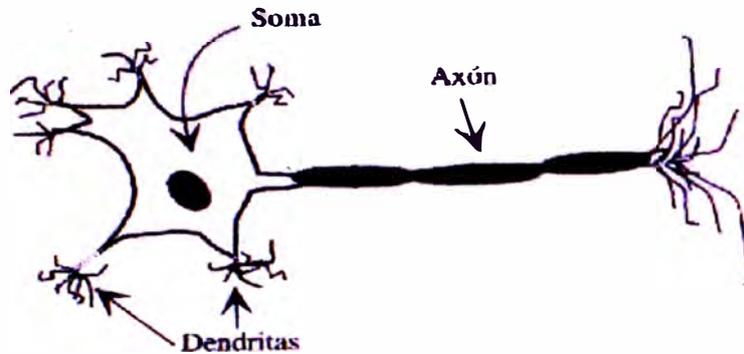


Fig. N° 3.1 Estructura típica de una neurona biológica

(Fuente: Entrenamiento de redes neuronales basado en algoritmos evolutivos)

La intensidad de una sinapsis no es fija, sino que puede ser modificada en base a la información proveniente del medio. De esta manera la estructura del cerebro no permanece fija sino que se va modificando por la formación de nuevas conexiones, ya sean excitadoras o inhibitoras, la destrucción de conexiones, la modificación de la intensidad de la sinapsis, o incluso por muerte neuronal [11].

Como se mencionó antes, una de las características principales de las neuronas, y que la distinguen del resto de las células, es su capacidad de comunicarse. Las señales nerviosas pueden ser eléctricas o químicas. La transmisión química se da principalmente en la comunicación entre neuronas, mientras que la eléctrica se produce dentro de una neurona. En general, una neurona recibe información de cientos de neuronas vecinas y la transmite a otras tantas neuronas [11].

La comunicación entre neuronas se lleva a cabo de la siguiente manera: en el soma de las neuronas transmisoras o pre sinápticas se genera un pulso eléctrico llamado potencial de acción. El pulso eléctrico se propaga a través del axón en dirección a las sinapsis. La información se transmite a las neuronas vecinas utilizando un proceso químico, mediante la liberación de neurotransmisores. Estos neurotransmisores se transmiten a través de la sinapsis hacia la neurona receptora. La neurona receptora o post sináptica toma la señal enviada por cientos de neuronas a través de las dendritas y la transmite al cuerpo celular. Estas señales pueden ser excitadoras (positivas) o inhibitoras (negativas). El soma es el encargado de integrar la información proveniente

de las distintas neuronas. Si la señal resultante supera un determinado umbral (umbral de disparo) el soma emite un pulso que se transmite a lo largo del axón dando lugar a la transmisión eléctrica a lo largo de la neurona. Al llegar la señal al extremo del axón se liberan neurotransmisores que permiten transmitir la señal a las neuronas vecinas [5] [6] [7].

3.3 Estructura de un sistema neuronal artificial

Las Redes Neuronales Artificiales son modelos matemáticos que intentan reproducir el comportamiento del cerebro humano. El principal objetivo de este modelo es la construcción de sistemas capaces de presentar un comportamiento inteligente. Esto implica la capacidad de aprender a realizar una determinada tarea [5] [6] [7].

Entre las principales capacidades de las Redes Neuronales Artificiales, podemos mencionar las siguientes:

- Capacidad de procesamiento paralelo, esta noción se sitúa en la base de la arquitectura de las Redes Neuronales Artificiales, consideradas como conjunto de entidades elementales que trabajan simultáneamente.
- Capacidad adaptativa por su proceso de aprendizaje, lo cual le permite tomar en cuenta nuevas restricciones y/o datos.
- Capacidad asociativa, proceso a través del cual puede recordarse de informaciones complejas, utilizando fragmentos de estas.
- Capacidad de auto-organización, esta propiedad asegura su estabilidad como sistema dinámico, al poder reorganizarse al adquirir conocimiento a través del proceso de aprendizaje.
- Capacidad de generalización, clasificación, extracción y optimización.

El verdadero poder de este modelo radica en el procesamiento paralelo realizado por las neuronas artificiales. La neurona artificial es un elemento de procesamiento simple y constituye el elemento principal de un sistema neuronal artificial.

Estas neuronas artificiales se combinan en estructuras denominadas capas. Una red neuronal artificial está compuesta por un conjunto de capas. De esta manera, la información se encuentra distribuida a lo largo de las sinapsis de la red, dándole a este sistema cierta tolerancia a fallas.

A su vez, las Redes Neuronales Artificiales son capaces de adaptar su funcionamiento a distintos entornos modificando sus conexiones entre neuronas. De esta manera pueden aprender de la experiencia y generalizar conceptos [5] [6] [7] [11].

3.4 Modelo de una neurona artificial

La neurona artificial es un elemento de procesamiento simple que a partir de un vector de entradas, produce una única salida. En general podemos encontrar tres tipos de neuronas artificiales, donde cada una de las cuales tiene su contraparte en el sistema nervioso, las cuales se detallan a continuación: [5] [6] [7] [11]

- Neuronas de entrada, reciben información directamente desde el exterior,
- Neuronas ocultas, reciben información desde otras neuronas artificiales. En las sinapsis de estas neuronas, se realiza la representación de la información almacenada.
- Neuronas de salida, reciben la información procesada y las devuelven al exterior.

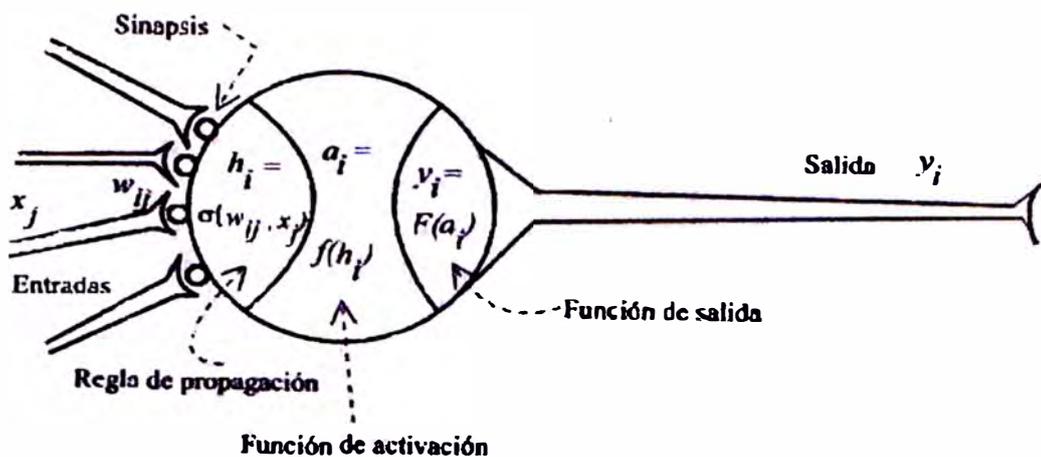


Fig. N° 3.2 Modelo de una neurona artificial

(Fuente: Entrenamiento de redes neuronales basado en algoritmos evolutivos)

En la figura 3.2 se muestran los elementos que componen una neurona artificial. Las partes que conforman la neurona artificial la detallamos a continuación:

3.4.1 Entradas y salidas

Las entradas y salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias y continuas. Las neuronas binarias (digitales) sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan dos alfabetos $\{0,1\}$ o $\{-1,1\}$. Por su parte, las neuronas continuas (analógicas) admiten valores dentro de un determinado rango, que en general suele definirse como $[-1,1]$. La selección del tipo de neurona a utilizar depende de la aplicación y del modelo a construir [5] [6] [7] [11].

3.4.2 Pesos sinápticos

El peso sináptico w_{ij} define la fuerza de una conexión sináptica entre dos neuronas, la neurona pre sináptica i y la neurona post sináptica j . Los pesos sinápticos pueden tomar valores positivos, negativos o cero. En caso de una entrada positiva, un peso positivo actúa como excitador, mientras que un peso negativo actúa como inhibidor. En caso de que el peso sea cero, no existe comunicación entre el par de neuronas [5] [6] [7] [11].

Mediante el ajuste de los pesos sinápticos la red es capaz de adaptarse a cualquier entorno y realizar una determinada tarea.

3.4.3 Regla de propagación

La regla de propagación determina el potencial resultante de la interacción de la neurona i con las N neuronas vecinas. El potencial resultante h_i se puede expresar de la siguiente manera:

$$h_i(t) = \sigma_i(w_{ij}, x_j(t)) \quad (3.1)$$

La regla de propagación más simple y utilizada consiste en realizar la suma de entradas ponderadas con sus pesos sinápticos correspondientes:

$$h_i(t) = \sum w_{ij} \cdot x_j(t) \quad (3.2)$$

3.4.4 Función de activación

La función de activación determina el estado de activación actual de la neurona en base al potencial resultante h_i y al estado de activación anterior de la neurona $a_i(t-1)$. El estado de activación de la neurona para un determinado instante de tiempo t puede ser expresado de la siguiente manera:

$$a_i(t) = f_i(a_i(t-1), h_i(t)) \quad (3.3)$$

Sin embargo, en la mayoría de los modelos se suele ignorar el estado anterior de la neurona, definiéndose el estado de activación en función del potencial resultante h_i :

$$a_i(t) = f_i(h_i(t)) \quad (3.4)$$

La tabla 3.1 muestra un listado de las funciones de activación más utilizadas en los distintos modelos de Redes Neuronales Artificiales.

Tabla N° 3.1 Funciones de activación

(Fuente: Entrenamiento de redes neuronales basado en algoritmos evolutivos)

FUNCION	FORMULA	RANGO
Identidad	$y = x$	$[-\infty, \infty]$
Escalón	$y = +1$ si $x \geq 0$ $y = 0$ si $x < 0$	$[0, 1]$
	$y = +1$ si $x \geq 0$ $y = -1$ si $x < 0$	$[-1, 1]$
Lineal a tramos	$y = x$ si $-1 \leq x \leq 1$	$[-1, 1]$
	$y = +1$ si $x > +1$	
	$y = -1$ si $x < -1$	
Sigmoidea	$y = (1 - e^{-x})^{-1}$	$[0, 1]$
	$y = \tanh(x)$	$[-1, 1]$
Sinusoidal	$y = \text{sen}(wx + \emptyset)$	$[-1, 1]$

3.4.5 Función de salida

La función de salida proporciona el valor de salida de la neurona, en base al estado de activación de la neurona. En general se utiliza la función identidad o aproximadamente lineal, es decir:

$$y_i(t) = F_i(a_i(t)) = a_i(t) \quad (3.5)$$

3.5 Arquitectura de una red neuronal artificial

Una vez definida el tipo de neurona que se utilizará en un modelo de Redes Neuronales Artificiales es necesario definir la topología de la misma.

La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

Las neuronas suelen agruparse en unidades funcionales denominadas capas. Se denomina capa de entrada a aquella que está compuesta por neuronas de entradas y por lo tanto recibe información procedente desde el exterior. Análogamente, se denomina capa oculta y capa de salida a aquellas capas que están compuestas por neuronas

ocultas y de salida respectivamente. Una red neuronal artificial está compuesta por una o más capas, las cuales se encuentran interconectadas entre sí [5] [6] [7] [11].

Entre un par de neuronas de la red neuronal artificial pueden existir conexiones. Estas conexiones son las sinapsis, tienen asociadas un peso sináptico, y son direccionales. Cuando la conexión se establece entre dos neuronas de una misma capa hablamos de conexiones laterales o conexiones intra-capas. Por el contrario, si la conexión se establece entre neuronas de distintas capas se la denomina conexión inter-capas. Si la conexión se produce en el sentido inverso al de entrada-salida la conexión se llama recurrente o realimentada [9].

Una red puede estar formada por una única capa de neuronas. En este caso hablamos de redes monocapas, y las neuronas que conforman dicha capa cumplen la función de neuronas de entrada y salida simultáneamente. Cuando la red está compuesta por dos o más capas hablamos de redes multicapas [9] [11].

A su vez, hablamos de Redes Neuronales Artificiales con conexión hacia delante (redes feedforward) cuando las conexiones entre las distintas neuronas de la red siguen un único sentido, desde la entrada de la red hacia la salida de la misma.

Cuando las conexiones son hacia delante o hacia atrás, hablamos de redes recurrentes (redes feedback).

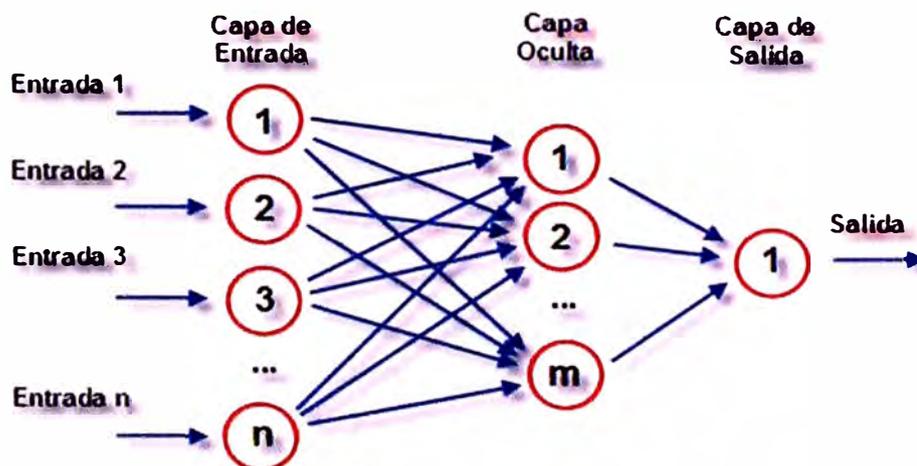


Fig. N° 3.3 Modelo de RNA Perceptrón Multicapa

(Fuente: http://es.wikipedia.org/wiki/Perceptr%C3%B3n_multicapa)

En la figura 3.3 se muestra el modelo perceptrón simple multicapa con n neuronas de entrada, m neuronas en su capa oculta y una neurona de salida.

3.6 Aprendizaje de una red neuronal artificial

Durante la operación de una red neuronal podemos distinguir claramente dos fases o modos de operación: la fase de aprendizaje o entrenamiento, y la fase de operación o ejecución.

Durante la primera fase, la fase de aprendizaje, la red es entrenada para realizar un determinado tipo de procesamiento. Una vez alcanzado un nivel de entrenamiento adecuado, se pasa a la fase de operación, donde la red es utilizada para llevar a cabo la tarea para la cual fue entrenada [5] [6] [7] [11].

3.6.1 Fase de entrenamiento

Una vez seleccionado el tipo de neurona artificial que se utilizará en una red neuronal y determinado su topología es necesario entrenarla para que la red pueda ser utilizada. Partiendo de un conjunto de pesos sinápticos aleatorio, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red desarrollar correctamente una determinada tarea. Durante el proceso de aprendizaje se va refinando iterativamente la solución hasta alcanzar un nivel de operación suficientemente bueno.

El proceso de aprendizaje se puede dividir en tres grandes grupos de acuerdo a sus características:

- **Aprendizaje supervisado.** Se presenta a la red un conjunto de patrones de entrada junto con la salida esperada. Los pesos se van modificando de manera proporcional al error que se produce entre la salida real de la red y la salida esperada.
- **Aprendizaje no supervisado.** Se presenta a la red un conjunto de patrones de entrada. No hay información disponible sobre la salida esperada. El proceso de entrenamiento en este caso deberá ajustar sus pesos en base a la correlación existente entre los datos de entrada.
- **Aprendizaje por refuerzo.** Este tipo de aprendizaje se ubica en medio de los dos anteriores. Se le presenta a la red un conjunto de patrones de entrada y se le indica a la red si la salida obtenida es o no correcta. Sin embargo, no se le proporciona el valor de la salida esperada. Este tipo de aprendizaje es muy útil en aquellos casos en que se desconoce cual es la salida exacta que debe proporcionar la red.

3.6.2 Fase de prueba

Una vez finalizada la fase de aprendizaje, la red puede ser utilizada para realizar la tarea para la que fue entrenada. Una de las principales ventajas que posee este modelo es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de

generalizar conceptos. De esta manera, una red neuronal puede tratar con información que no le fue presentada durante la fase de entrenamiento.

3.7 Métodos de entrenamiento de redes neuronales artificiales

En el contexto de las Redes Neuronales Artificiales el aprendizaje puede ser visto como el proceso de ajuste de los parámetros libres de la red. Partiendo de un conjunto de pesos sinápticos aleatorios, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red desarrollar correctamente una determinada tarea. El proceso de aprendizaje es un proceso iterativo, en el cual se va refinando la solución hasta alcanzar un nivel de operación suficientemente bueno.

La mayoría de los métodos de entrenamiento utilizados en las Redes Neuronales Artificiales con conexión hacia adelante, consisten en proponer una función de error que mida el rendimiento actual de la red en función de los pesos sinápticos. El objetivo del método de entrenamiento es encontrar el conjunto de pesos sinápticos que minimizan (o maximizan) la función. El método de optimización proporciona una regla de actualización de los pesos que en función de los patrones de entrada modifica iterativamente los pesos hasta alcanzar el punto óptimo de la red neuronal [5] [6] [7] [11].

3.7.1 Métodos de gradiente descendente

El método de entrenamiento más utilizado para entrenar a una red neuronal, es el método del gradiente descendente. Este método define una función $E(W)$ que proporciona el error que comete la red en función del conjunto de pesos sinápticos W . El objetivo del aprendizaje será encontrar la configuración de pesos que corresponda al mínimo global de la función de error, aunque en muchos casos es suficiente encontrar un mínimo local lo suficientemente bueno.

El principio general del método es el siguiente: dado un conjunto de pesos $W(0)$ para el instante de tiempo $t=0$, se calcula la dirección de máxima variación del error. La dirección de máximo crecimiento de la función $E(W)$ en $W(0)$ viene dado por el gradiente $\nabla E(W)$. Luego, se actualizan los pesos siguiendo el sentido contrario al indicado por el gradiente $\nabla E(W)$, dirección que indica el sentido de máximo decrecimiento. De este modo se va produciendo un descenso por la superficie de error hasta alcanzar un mínimo local.

$$W(t+1) = W(t) - \alpha \nabla E(W) \quad (3.6)$$

En donde α indica el tamaño del peso tomado en cada iteración, pudiendo ser diferente para cada peso e idealmente debería ser infinitesimal. El tamaño del paso es un

factor importante a la hora de diseñar un método de estas características. Si se toma un paso muy chico el proceso de entrenamiento resulta muy lento, mientras que si el tamaño del paso es muy grande se producen oscilaciones en torno al punto mínimo.

3.7.2 El algoritmo de backpropagation

El algoritmo backpropagation es el método de entrenamiento más utilizado en redes con conexión hacia delante. Es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases. Primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias [Fritsch, 1996]. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los errores de los pesos sinápticos de cada neurona.

CAPITULO IV

CASO DE ESTUDIO

4.1 Introducción

La proyección de la demanda a corto plazo usando Redes Neuronales Artificiales, propone una metodología que permite realizar este pronóstico para el SEIN, las cuales también pueden aplicarse en barras de sistemas de distribución, para lo cual se debe obtener información histórica de la demanda eléctrica con más de 2 años de data histórica.

Como se ha detallado la temperatura juega un papel importante en la demanda eléctrica, sin embargo la variabilidad de la temperatura del sistema eléctrico peruano hace que este sea complejo de predecir. Por esta razón la temperatura no se ha considerado en la elaboración de la arquitectura o modelo de la red neuronal.

4.2 Selección de la barra a proyectar

Las proyecciones de las demandas eléctricas diarias usando Redes Neuronales Artificiales, se aplicaron al SEIN, considerando a esta como una barra con demanda eléctrica nodal.

Para la creación de la base de datos, se ha obtenido como información histórica el perfil de la demanda eléctrica diaria del SEIN de los años 2010; 2011 y hasta agosto del 2012. El perfil de demanda comprende información de potencia activa (MW) cada 30 minutos para un periodo diario de 24 horas. Por lo cual diariamente se tiene 48 datos medidos de potencia activa cada 30 minutos. Esta información está disponible en el portal web del COES.

Los datos históricos de los años 2010 y 2011 servirán para realizar el entrenamiento de la red neuronal con el software desarrollado en Matlab, y los datos del 2012 permitirán realizar la comparación de los resultados obtenidos luego de realizar el entrenamiento de la red neuronal.

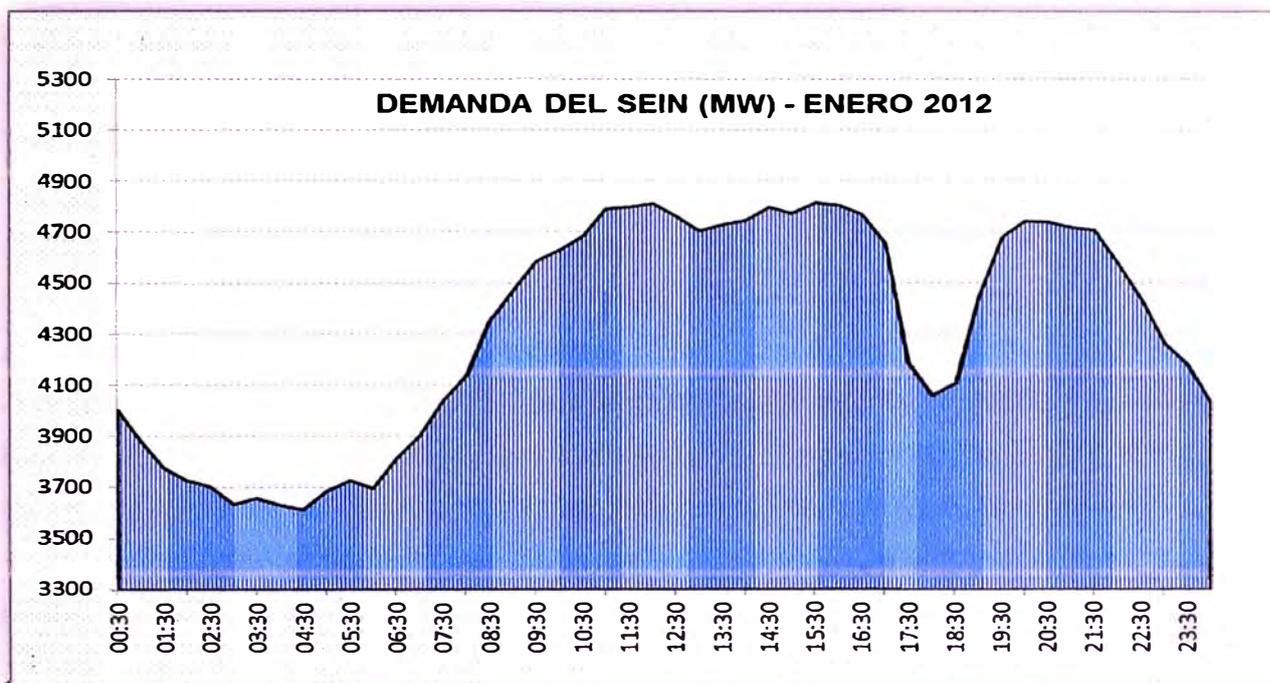


Fig. N° 4.1 Perfil de demanda eléctrica del SEIN en periodo de avenida
(Fuente: Comité de operación económica del sistema interconectado nacional)

En la figura 4.1 se muestra el perfil de demanda eléctrica del SEIN en la estación del verano.

4.3 Elaboración de la base de datos

4.3.1 Depuración de datos históricos

De la información histórica obtenida para los años de análisis se debe realizar una depuración de datos considerando que existen mediciones erradas.

A continuación se detallan los pasos a seguir para la corrección de los valores de demanda eléctrica históricos, el cual permitirá elaborar la base de datos.

- a. Seleccionar y agrupar todos los días de la semana.
- b. Identificar qué días de la semana han sido feriados.
- c. Para cada día típico de la semana (que no es feriado) hallar la mediana de las matrices de las demandas históricas en grupos de cuatro valores contiguos para cada día típico.
- d. Hacer una comparación de la mediana con los cuatro datos históricos contiguos, de haber una variación mayor al +/-10 % se procede a corregir estos valores históricos. El nuevo valor corregido será la mediana de los otros cuatro valores históricos.

e. Para los periodos en los cuales se perdió una demanda eléctrica mayor al 10% de la barra ya sea por mantenimiento programado o por perturbaciones. Se tiene que identificar la fecha y el horario en el cual se perdió la demanda eléctrica, la cual será corregida y reemplazada con los datos de la mediana de los cuatros datos históricos contiguos del día típico que ocurrió el evento.

4.3.2 Ordenamiento de la base de datos

Luego de culminar la depuración de los datos históricos, se procede a elaborar un formato en Excel para el ingreso en datos en Matlab. El formato utilizado para el ingreso de datos históricos debe ser igual al formato utilizado para realizar la comparación luego del entrenamiento de la red neuronal. En la tabla 4.1 se muestra el formato de ingreso de datos [8].

Tabla N° 4.1 Formato de ingreso de datos
(Fuente: elaboración propia)

DEMANDA ELÉCTRICA DIARIA DEL SEIN							
FECHA	00:00	00:30	01:00	23:59
Día 1							
Día 2							
Día 2							
...							
...							
Día n							

4.4 Características de la red neuronal artificial usado

En la presente tesis se ha desarrollado una RNA con las siguientes características:

4.4.1 Topología de la red neuronal artificial

Las capas de entrada (compuesta por neuronas de entradas) están formadas por 99 neuronas de entrada, los cuales se detallan a continuación:

- a. 48 neuronas que corresponden a los valores de la demanda eléctrica cada media hora, de la semana anterior correspondiente al día típico a pronosticar.

- b. 48 neuronas que corresponden a los valores de demanda eléctrica del día anterior al día a proyectar.
- c. 1 neurona que corresponde al día típico de la semana a proyectar.
- d. 1 neurona que permite identificar si el día a proyectar es feriado o no.
- e. 1 neurona de la estación de año.

La capa de salida (compuesta por neuronas de salida), está formada por 48 neuronas que corresponden a los valores de demanda eléctrica que ha sido proyectado en intervalos de cada media hora para un día.

Las capas ocultas, la red neuronal artificial seleccionada consta de tres (3) capas ocultas las cuales se interconectan con las capas de entrada y salida.

4.4.2 Función de activación de la red neuronal artificial

Se ha utilizado la función de activación o transferencia denominado "Purelin", la cual corresponde a la función de transferencia lineal, cuya característica se detalla en la figura 4.2

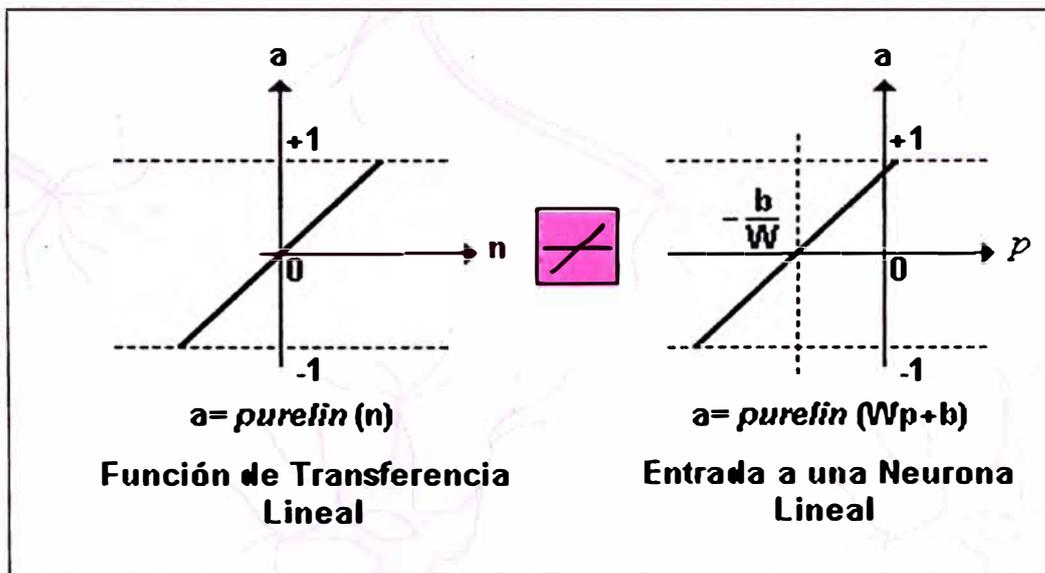


Fig. N° 4.2 Función de transferencia lineal "purelin"

(Fuente: Entrenamiento de redes neuronales basado en algoritmos evolutivos)

4.4.3 Método de entrenamiento de la red neuronal artificial

El algoritmo de backpropagation es el método usado para el entrenamiento de la red neuronal seleccionado. Es un método de aprendizaje supervisado de gradiente descendente. Cuando se usa el software matlab en el proceso de programación de la red neuronal, se usa la función "traingdx" para este tipo de entrenamiento.

Con la finalidad de obtener los mejores resultados en la proyección de la demanda, se realizaron entrenamientos de la red neuronal, considerando la estación del año y sin considerar las estaciones del año.

Los valores de salida obtenidos, vendrían a ser el pronóstico de la demanda diaria (a corto plazo). Ósea se obtienen 48 valores que corresponden a la proyección de la demanda cada media hora para un día específico.

4.4.4 Arquitectura de la red neuronal artificial

A continuación se muestra la arquitectura del modelo de red neuronal propuesto, sin considerar las estaciones del año.

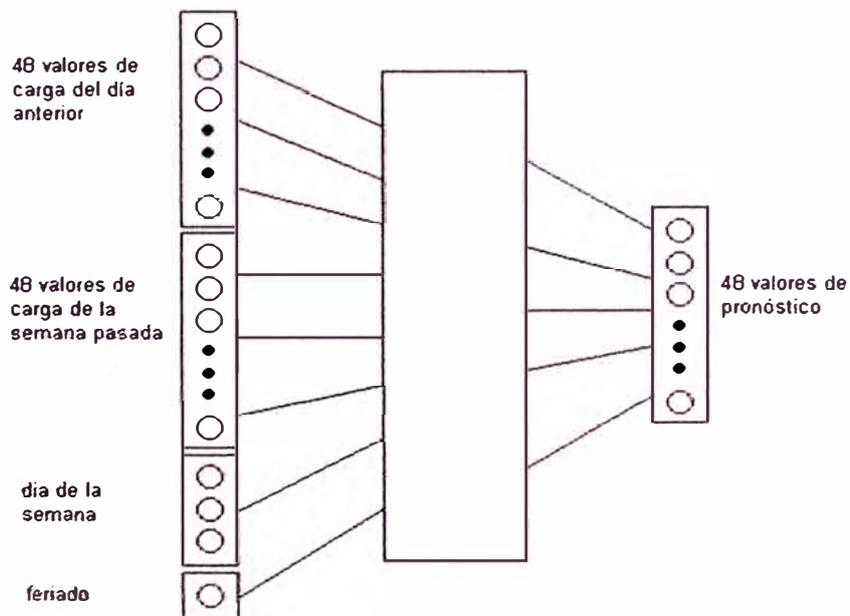


Fig. N° 4.3 Arquitectura de la RNA sin considerar las estaciones del año
(Fuente: elaboración propia)

A continuación se muestra la arquitectura del modelo de red neuronal propuesto, considerando las estaciones del año.

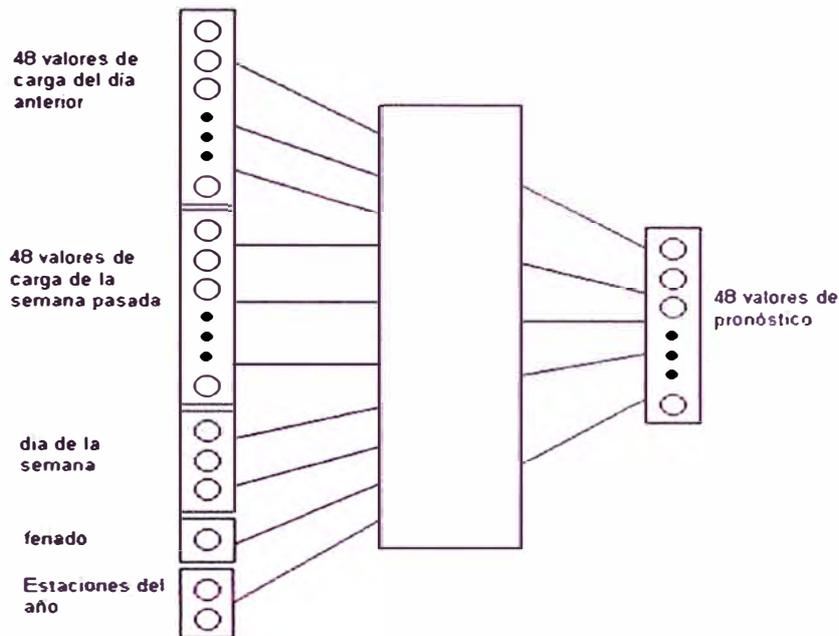


Fig. N° 4.4 Arquitectura de la RNA considerando las estaciones del año
(Fuente: elaboración propia)

4.5 Indicador del pronóstico de la demanda eléctrica

Los indicadores del pronóstico de la demanda, se usan para determinar qué tan eficaz es un pronóstico con respecto a los valores reales; es decir, se busca obtener una medida que muestre que tan lejos se encuentran los valores pronosticados de los obtenidos en la realidad.

Con la finalidad de validar los resultados obtenidos a través de este método de cálculo y realizar una comparación con la metodología actual que se viene realizando para la proyección de la demanda diaria del SEIN, usaremos como indicador de precisión de pronóstico el MAPE (error medio porcentual absoluto) para cada día de estimación.

Entonces el MAPE diario estará dado por:

$$MAPE = \frac{\sum_{i=1}^{48} \left[\left| \frac{x_i - y_i}{y_i} \right| \times 100\% \right]}{48} \quad (4.1)$$

Donde:

X_i ; es el valor de la demanda real (ejecutado)

Y_i ; es el valor de la demanda pronosticado (proyectado)

$i = 1, 2, \dots, 48$; son los intervalos del día cada media hora de 00:00 a 24:00 h

CAPITULO V

SIMULACIONES Y RESULTADOS

En este capítulo se detalla el entrenamiento de la Red Neuronal Artificial y las simulaciones realizadas, utilizando diferentes arquitecturas, hasta obtener la topología propuesta con la cual se obtiene mejores resultados de la proyección de la demanda eléctrica. Aquí también se presentan los métodos de validación de los resultados y se analizan los resultados desde el punto de vista de desempeño de las RNA.

5.1 Diseño y entrenamiento de la red neuronal artificial

El diseño de la red neuronal, consiste principalmente en determinar el número de capas ocultas y la cantidad de neuronas en cada una de ellas. Para lo cual se realizó el método de prueba – error, es decir; se buscó el mejor número de capas ocultas de la RNA de forma tal que produzca una mejor predicción de la demanda eléctrica o nos arroje el menor error de cálculo. Sobre esta óptica se desarrolla el trabajo para encontrar los mejores resultados previstos para cada día de la semana.

Durante el caso de estudio, se realizaron varios entrenamientos a las RNA utilizando el tipo de entrenamiento denominado “traingdx” como algoritmo de aprendizaje. “Traingdx” significa: Gradient descent with momentum and adaptive learning rate backpropagat.

La información histórica de los años 2010 y 2011, son usados como datos de ingreso, para la Red Neuronal Artificial, los cuales permiten efectuar el proceso de **entrenamiento y proyección de la demanda eléctrica diaria**. Estos datos de ingreso han sido corregidos de acuerdo a la metodología propuesta en este trabajo.

La información histórica del 1 de enero al 31 de agosto del año 2012, es usada para hacer las comparaciones con las proyecciones obtenidas de la Red Neuronal Artificial. De esta forma se conoce la performance de la RNA luego de su entrenamiento. Ósea se usa esta data para comparar la proyección de la demanda realizada por la RNA con la demanda real.

En esta tesis se realizaron varios entrenamientos de la RNA, con una arquitectura de 2; 3; 4 y 5 capas internas. Los mejores resultados se obtuvieron con 2 y 3 capas internas. Siendo la RNA con 3 capas la de mejor performance.

También se realizaron simulaciones para diferentes funciones de transferencia. Los mejores resultados se obtuvieron con las funciones del tipo “tribas” y “purelin” en ese orden. Así podemos afirmar que estas dos funciones de transferencias pueden ser usadas para las proyecciones de demandas porque son las que mejor se acomodan para este caso.

En la figura 5.1 se muestra el proceso de entrenamiento de la red neuronal en el entorno de matlab. Se puede visualizar que la red neuronal está compuesta de tres (3) capas ocultas, también se visualiza el algoritmo de entrenamiento, su performance y el modelo usado. Así mismo también se visualiza el proceso iterativo, en el cual se puede apreciar el tiempo de entrenamiento de la red neuronal y cuál es su error.

Se puede culminar con el proceso de entrenamiento haciendo clic en el botón “stop training”, luego se procede a guardar la red neuronal entrenada.

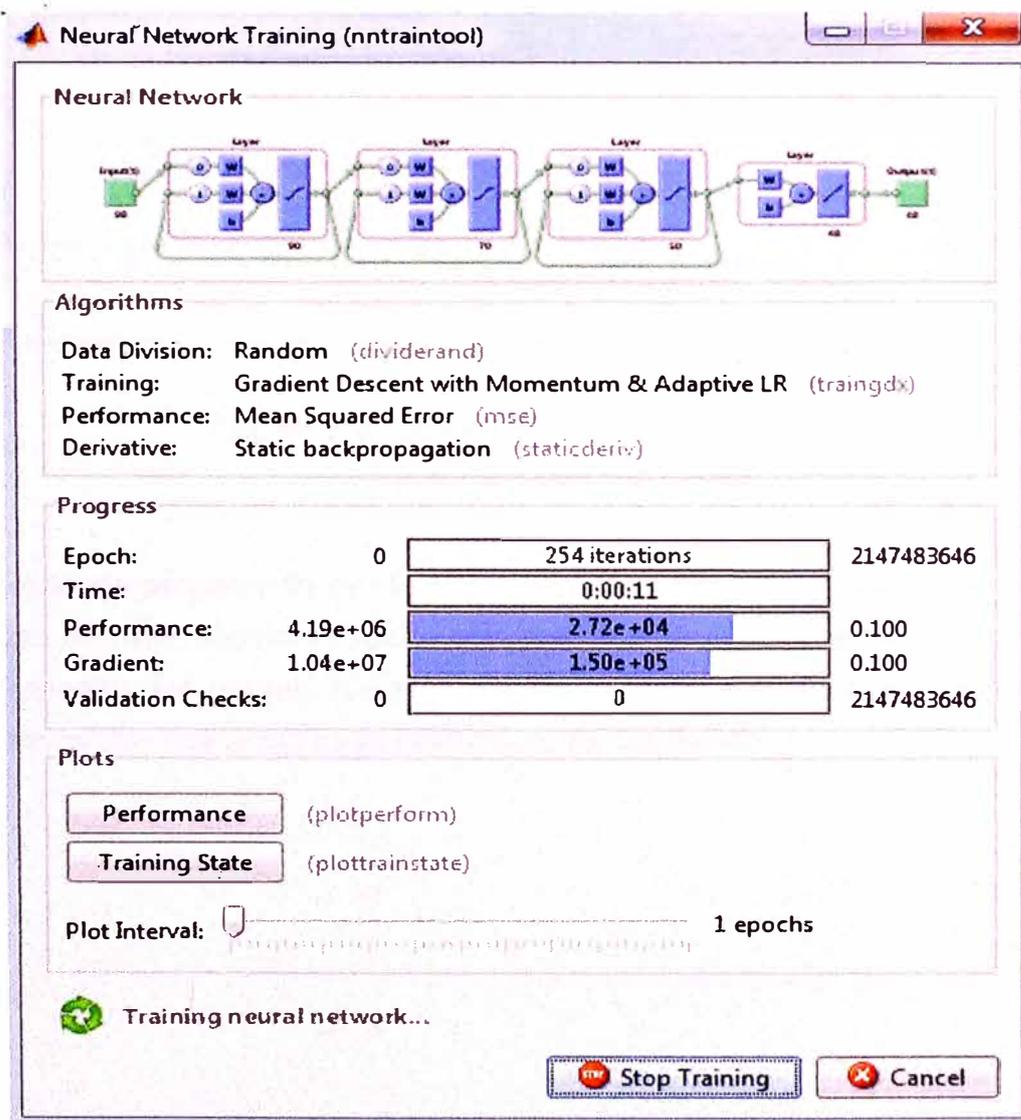


Fig. N° 5.1 Proceso de entrenamiento de la RNA
(Fuente: elaboración propia)

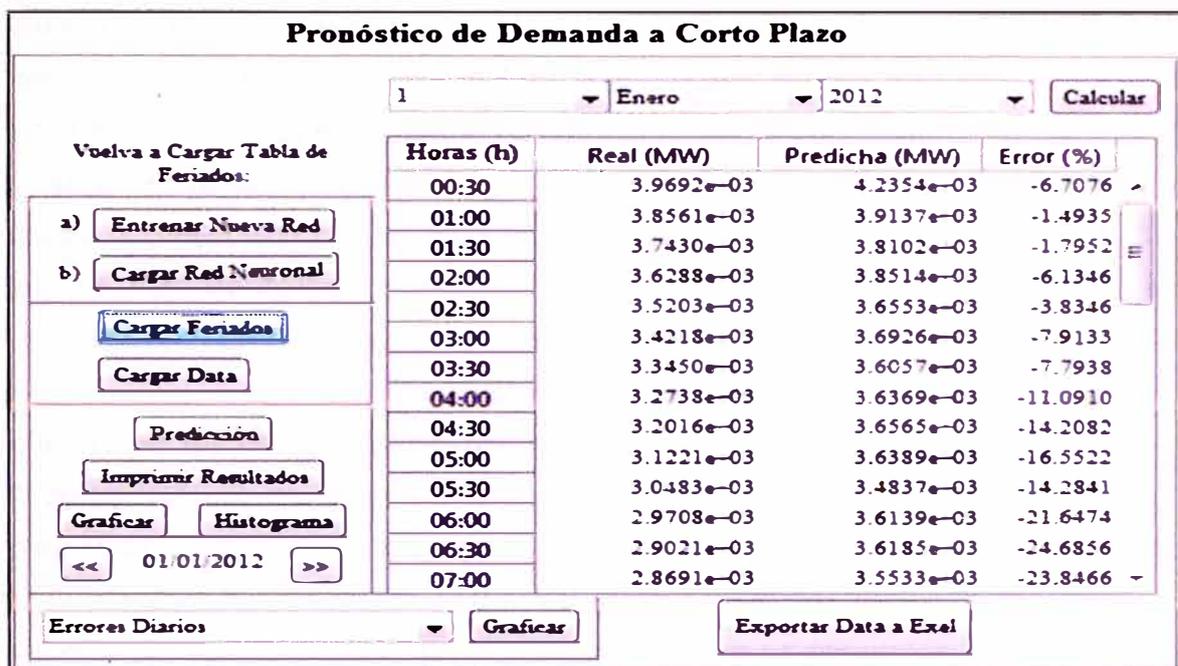


Fig. N° 5.3 Módulo de proyección y análisis de performance de la RNA
(Fuente: elaboración propia)

- **Módulo de proyección diaria de la demanda:** Una vez que ya se ha encontrado la mejor RNA para la proyección de la demanda, se procede a realizar la proyección de la demanda diaria a través de este módulo.

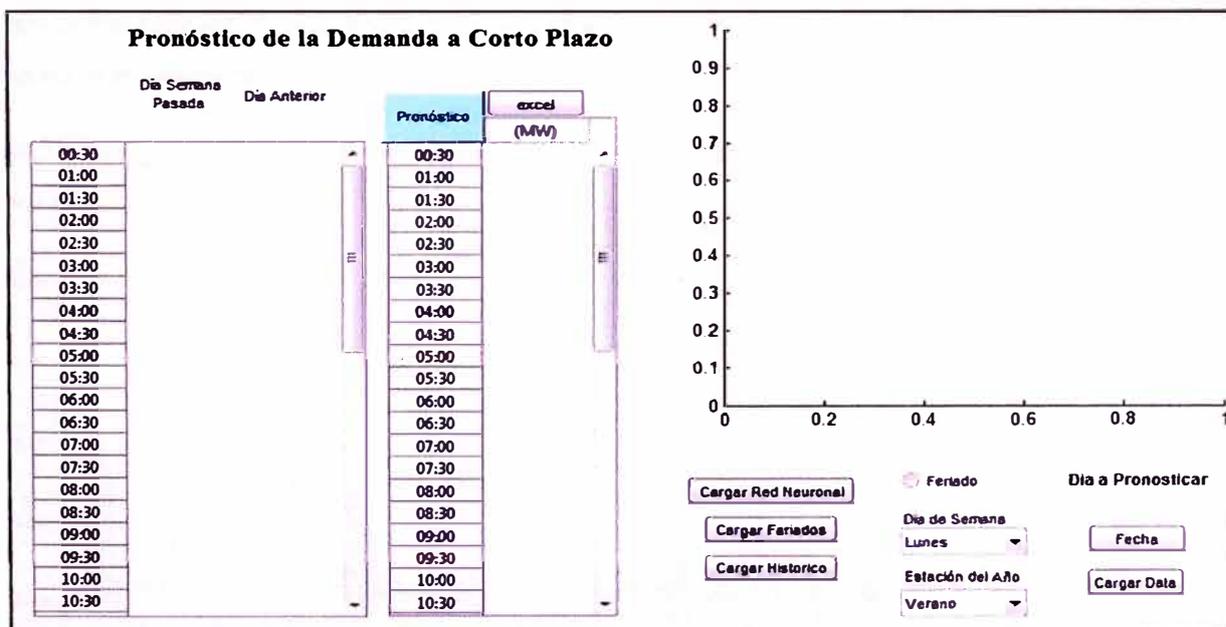


Fig. N° 5.4 Módulo de proyección diaria de la demanda
(Fuente: elaboración propia)

5.3 Validación de los resultados de la proyección

La proyección de la demanda eléctrica obtenida por la Red Neuronal Artificial, es validada por el indicador de error medio porcentual absoluto para cada día de proyección (MAPE – Mean Absolute Percent Error). El MAPE diario es obtenido comparando los resultados de la proyección con los valores reales de la demanda del SEIN del año 2012. El objetivo de este trabajo es obtener valores de MAPE menores del 2.0%. De esta manera consideraremos si la proyección eléctrica diaria es aceptable.

5.4 Resultados de las proyecciones diarias del SEIN

Se hicieron las proyecciones de la demanda eléctrica diaria del SEIN, variando la cantidad de capas ocultas de la RNA, es decir; con 2 y 3 capas ocultas con la finalidad de encontrar la arquitectura que mejores resultados arroje en la proyección de la demanda.

Las proyecciones diarias del SEIN usando RNA, que se presentan en este Capítulo corresponden a los meses de abril, Mayo, junio, julio y agosto del año 2012, los cuales se muestran a continuación.

5.4.1 Proyección diaria usando la RNA con 2 capas ocultas

En la figura N° 5.5 se muestra de manera gráfica y resumida el indicador MAPE de las proyecciones diarias del SEIN usando la RNA con 2 capas ocultas. Esta proyección corresponde a los meses de abril, mayo, junio, julio y agosto del 2012. Como podemos observar los resultados de estas proyecciones arrojan desviaciones con valores de MAPE mayores a nuestro objetivo (2.0%).

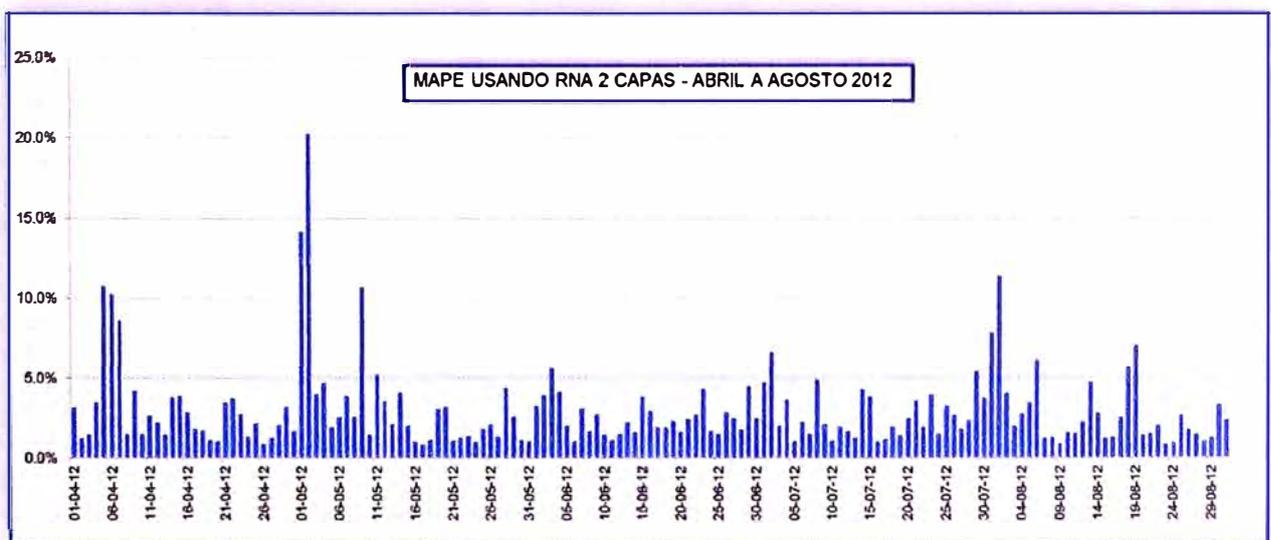


Fig. N° 5.5 MAPE de las proyecciones del SEIN usando RNA con 2 capas ocultas
(Fuente: elaboración propia)

Los mayores errores de proyección por mes se detallan a continuación:

- Mes de Abril: 05.04.12 (MAPE = 10.74%); 06.04.12 (MAPE = 10.26%) y 07.04.12 (MAPE = 8.61%)
- Mes de Mayo: 01.05.12 (MAPE = 14.15%); 02.05.12 (MAPE = 20.25%) y 09.05.12 (MAPE = 10.63%);
- Mes de Julio: 02.07.12 (MAPE = 6.60%) y 31.07.12 (MAPE = 7.78%)
- Mes de agosto: 01.08.12 (MAPE = 11.36%); 06.08.12 (MAPE = 6.09%) y 19.08.12 (MAPE = 6.99%);

En la tabla N° 5.1 se muestra la relación de los MAPE de las proyecciones diarias del SEIN usando RNA con 2 capas ocultas, correspondiente a los meses de abril, mayo, junio, julio y agosto del 2012.

Tabla N° 5.1 MAPE de las proyecciones del SEIN usando RNA con 2 capas ocultas

(Fuente: elaboración propia)

DIA	MAPE	DIA	MAPE	DIA	MAPE	DIA	MAPE	DIA	MAPE
01-04-12	3.11%	01-05-12	14.15%	01-06-12	3.25%	01-07-12	4.72%	01-08-12	11.36%
02-04-12	1.19%	02-05-12	20.25%	02-06-12	3.90%	02-07-12	6.60%	02-08-12	4.03%
03-04-12	1.44%	03-05-12	4.00%	03-06-12	5.63%	03-07-12	2.03%	03-08-12	1.99%
04-04-12	3.47%	04-05-12	4.63%	04-06-12	4.12%	04-07-12	3.61%	04-08-12	2.78%
05-04-12	10.74%	05-05-12	1.89%	05-06-12	2.04%	05-07-12	1.04%	05-08-12	3.40%
06-04-12	10.26%	06-05-12	2.56%	06-06-12	1.04%	06-07-12	2.24%	06-08-12	6.09%
07-04-12	8.61%	07-05-12	3.86%	07-06-12	3.11%	07-07-12	1.48%	07-08-12	1.23%
08-04-12	1.51%	08-05-12	2.54%	08-06-12	1.67%	08-07-12	4.86%	08-08-12	1.30%
09-04-12	4.21%	09-05-12	10.63%	09-06-12	2.71%	09-07-12	2.11%	09-08-12	0.88%
10-04-12	1.48%	10-05-12	1.46%	10-06-12	1.45%	10-07-12	1.10%	10-08-12	1.58%
11-04-12	2.65%	11-05-12	5.23%	11-06-12	1.11%	11-07-12	1.93%	11-08-12	1.54%
12-04-12	2.17%	12-05-12	3.54%	12-06-12	1.49%	12-07-12	1.67%	12-08-12	2.24%
13-04-12	1.44%	13-05-12	2.11%	13-06-12	2.24%	13-07-12	1.26%	13-08-12	4.71%
14-04-12	3.78%	14-05-12	4.07%	14-06-12	1.63%	14-07-12	4.28%	14-08-12	2.79%
15-04-12	3.87%	15-05-12	2.00%	15-06-12	3.81%	15-07-12	3.83%	15-08-12	1.21%
16-04-12	2.83%	16-05-12	1.02%	16-06-12	2.94%	16-07-12	1.01%	16-08-12	1.27%
17-04-12	1.81%	17-05-12	0.82%	17-06-12	1.90%	17-07-12	1.16%	17-08-12	2.53%
18-04-12	1.68%	18-05-12	1.11%	18-06-12	1.91%	18-07-12	1.94%	18-08-12	5.65%
19-04-12	1.14%	19-05-12	3.04%	19-06-12	2.29%	19-07-12	1.40%	19-08-12	6.99%
20-04-12	1.04%	20-05-12	3.16%	20-06-12	1.60%	20-07-12	2.46%	20-08-12	1.41%
21-04-12	3.44%	21-05-12	1.10%	21-06-12	2.37%	21-07-12	3.57%	21-08-12	1.51%
22-04-12	3.74%	22-05-12	1.24%	22-06-12	2.68%	22-07-12	1.95%	22-08-12	2.03%
23-04-12	2.71%	23-05-12	1.36%	23-06-12	4.31%	23-07-12	3.96%	23-08-12	0.82%
24-04-12	1.35%	24-05-12	1.01%	24-06-12	1.67%	24-07-12	1.49%	24-08-12	0.90%
25-04-12	2.15%	25-05-12	1.81%	25-06-12	1.48%	25-07-12	3.23%	25-08-12	2.64%
26-04-12	0.86%	26-05-12	2.11%	26-06-12	2.82%	26-07-12	2.62%	26-08-12	1.79%
27-04-12	1.24%	27-05-12	1.32%	27-06-12	2.47%	27-07-12	1.82%	27-08-12	1.45%
28-04-12	2.08%	28-05-12	4.36%	28-06-12	1.77%	28-07-12	2.35%	28-08-12	1.05%
29-04-12	3.15%	29-05-12	2.61%	29-06-12	4.46%	29-07-12	5.40%	29-08-12	1.24%
30-04-12	1.65%	30-05-12	1.11%	30-06-12	2.48%	30-07-12	3.75%	30-08-12	3.31%
		31-05-12	1.05%			31-07-12	7.78%	31-08-12	2.37%
MAPE ABRIL	3.03%	MAPE MAYO	3.59%	MAPE JUNIO	2.54%	MAPE JULIO	2.86%	MAPE AGOSTO	2.71%

El MAPE promedio de las proyecciones de la RNA con 2 capas ocultas, correspondiente a los meses de abril, mayo, junio, julio, y agosto fue 2.95%. Estas altas desviaciones de proyección se deben principalmente a la inadecuada topología de la RNA con solo 2 capas ocultas, y a los días feriados y días contiguos a los feriados que incrementan el error de proyección.

Así mismo también se obtuvieron proyecciones cuyos MAPE fueron menores al 1%, a través de esta topología de la RNA, las cuales detallamos a continuación:

- **26.04.12 (MAPE = 0.86%)**
- **17.05.12 (MAPE = 0.82%)**
- **09.08.12 (MAPE = 0.88%)**
- **23.08.12 (MAPE = 0.82%)**
- **24.08.12 (MAPE = 0.90%)**

A continuación se muestran gráficamente los resultados obtenidos de los días 23 y 24 de agosto del 2012, cuyos valores de MAPE son 0.82% y 0.90% respectivamente.

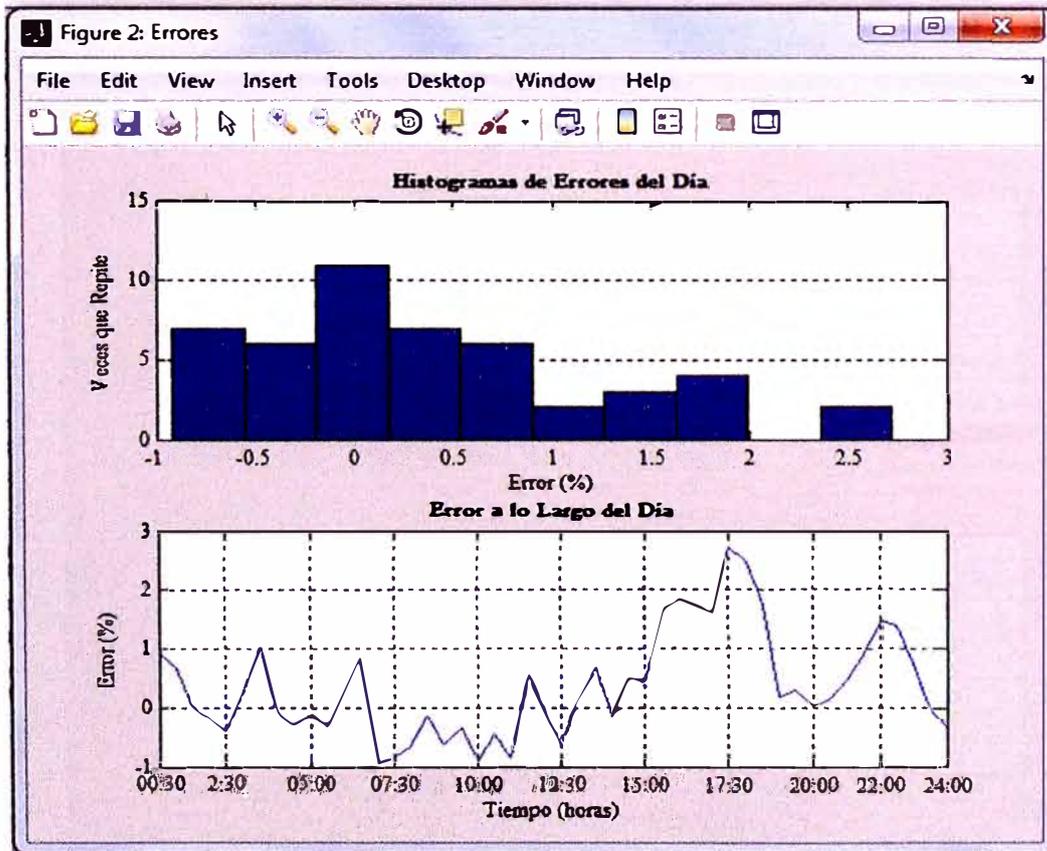
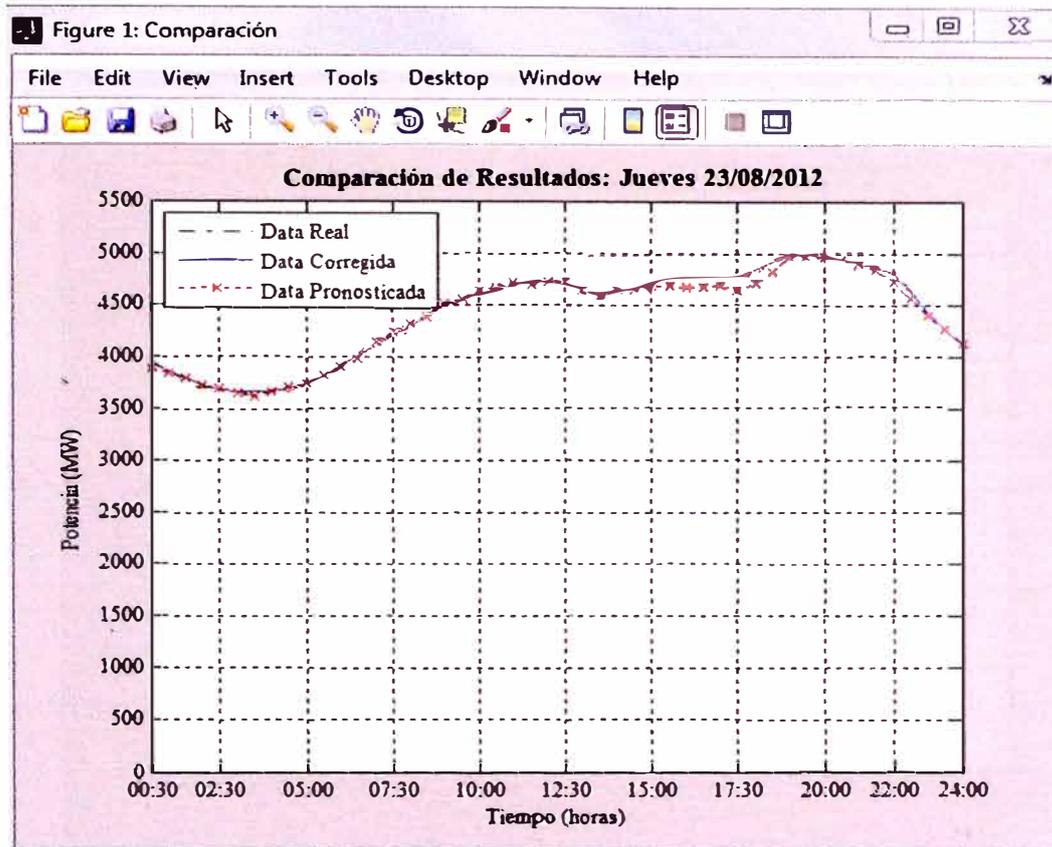


Fig. N° 5.6 Proyección del 23 agosto 2012 con histograma de errores
(Fuente: elaboración propia)

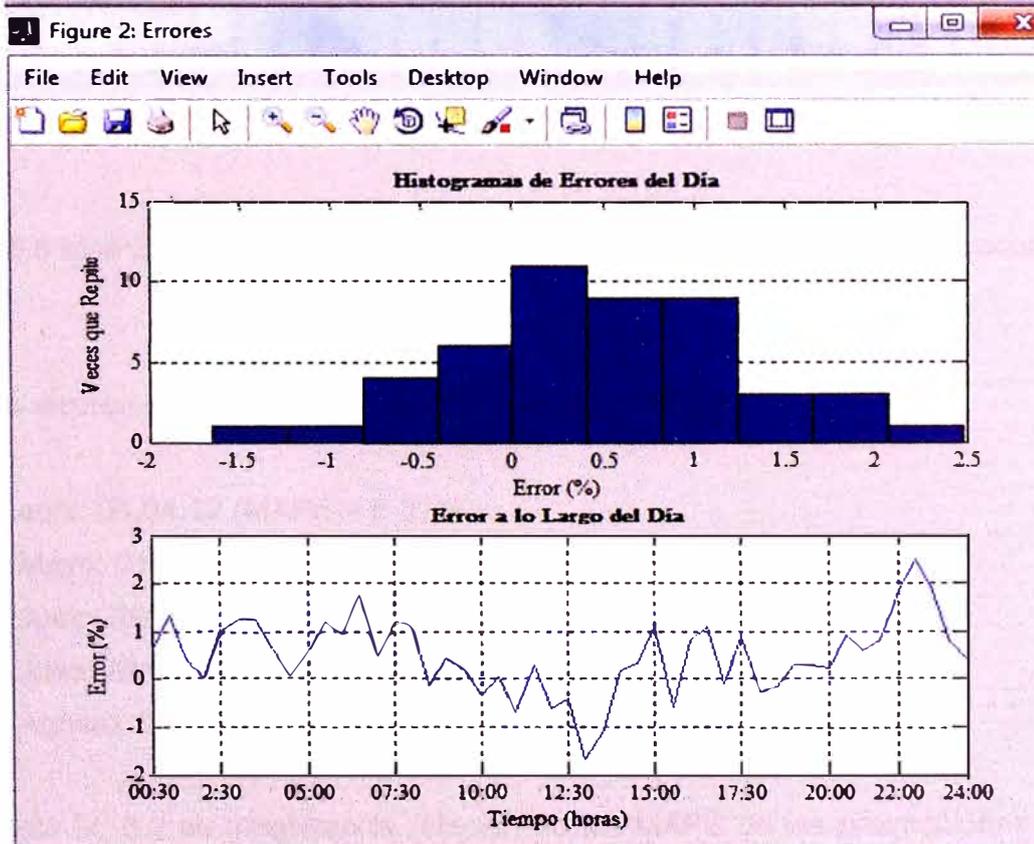
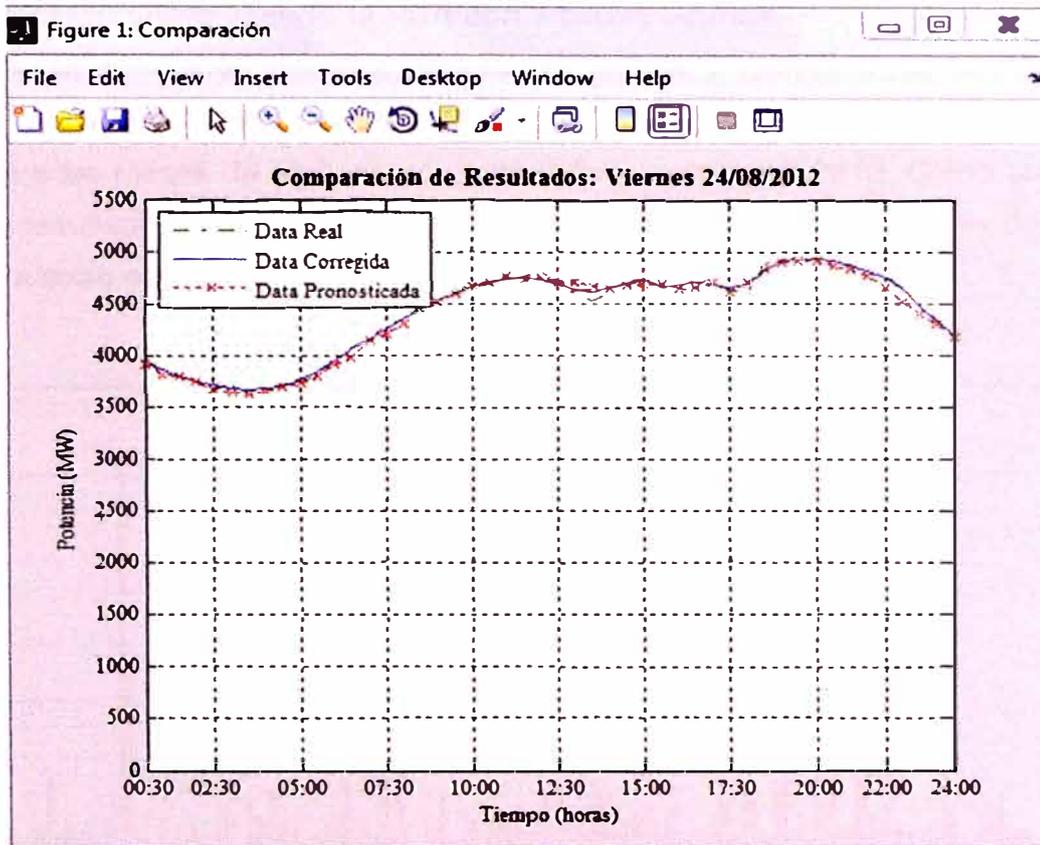


Fig. N° 5.7 Proyección del 24 agosto 2012 con histograma de errores
(Fuente: elaboración propia)

5.4.2 Proyección diaria usando la RNA con 3 capas ocultas

En la figura N° 5.8 se muestra de manera gráfica y resumida los MAPE de la proyección diaria del SEIN usando RNA con 3 capas ocultas. Esta proyección corresponde a los meses de abril, mayo, junio, julio y agosto del 2012. Como podemos observar los resultados de estas proyecciones arrojan desviaciones con valores de MAPE cercanos a nuestro objetivo (2.0%).

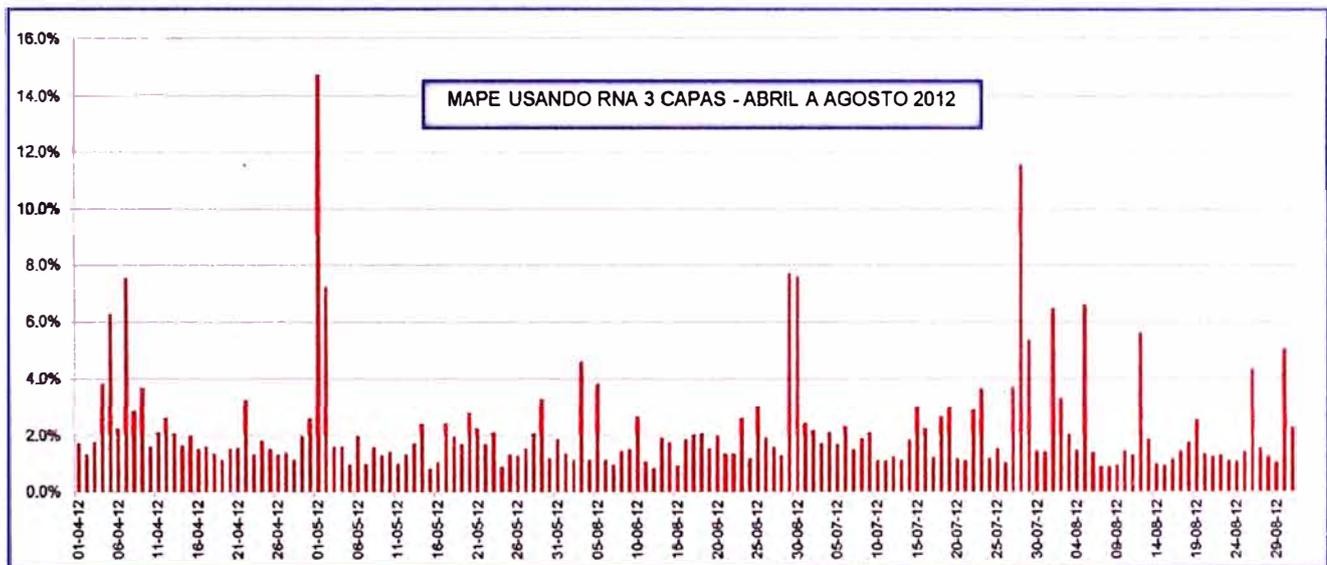


Fig. N° 5.8 MAPE de las proyecciones del SEIN usando RNA con 3 capas ocultas
(Fuente: elaboración propia)

Los mayores errores de las proyecciones por mes se detallan a continuación:

- Mes de abril: 05.04.12 (MAPE = 6.27%)
- Mes de Mayo: 01.05.12 (MAPE = 14.74%) y 02.05.12 (MAPE = 7.25%)
- Mes de Junio: 29.06.12 (MAPE = 7.73%) y 30.06.12 (MAPE = 7.60%)
- **Mes de Julio: 28.07.12 (MAPE = 11.58%)**
- Mes de Agosto: 01.08.12 (MAPE = 6.48%) y 05.08.12 (MAPE = 6.60%)

En la tabla N° 5.2 se muestran la relación de los MAPE de las proyecciones diarias del SEIN usando RNA con 3 capas ocultas

Tabla N° 5.2 MAPE de las proyecciones del SEIN usando RNA con 3 capas ocultas
(Fuente: elaboración propia)

DIA	MAPE	DIA	MAPE	DIA	MAPE	DIA	MAPE	DIA	MAPE
01-04-12	1.70%	01-05-12	14.74%	01-06-12	1.36%	01-07-12	2.45%	01-08-12	6.48%
02-04-12	1.32%	02-05-12	7.25%	02-06-12	1.14%	02-07-12	2.18%	02-08-12	3.32%
03-04-12	1.72%	03-05-12	1.60%	03-06-12	4.61%	03-07-12	1.72%	03-08-12	2.02%
04-04-12	3.83%	04-05-12	1.62%	04-06-12	1.14%	04-07-12	2.10%	04-08-12	1.46%
05-04-12	6.27%	05-05-12	0.97%	05-06-12	3.83%	05-07-12	1.69%	05-08-12	6.60%
06-04-12	2.25%	06-05-12	1.97%	06-06-12	1.13%	06-07-12	2.32%	06-08-12	1.38%
07-04-12	7.54%	07-05-12	1.00%	07-06-12	0.97%	07-07-12	1.48%	07-08-12	0.90%
08-04-12	2.88%	08-05-12	1.58%	08-06-12	1.44%	08-07-12	1.88%	08-08-12	0.89%
09-04-12	3.66%	09-05-12	1.30%	09-06-12	1.51%	09-07-12	2.10%	09-08-12	0.95%
10-04-12	1.61%	10-05-12	1.42%	10-06-12	2.68%	10-07-12	1.12%	10-08-12	1.45%
11-04-12	2.12%	11-05-12	0.99%	11-06-12	1.08%	11-07-12	1.10%	11-08-12	1.30%
12-04-12	2.62%	12-05-12	1.32%	12-06-12	0.84%	12-07-12	1.23%	12-08-12	5.62%
13-04-12	2.08%	13-05-12	1.72%	13-06-12	1.90%	13-07-12	1.11%	13-08-12	1.86%
14-04-12	1.64%	14-05-12	2.41%	14-06-12	1.74%	14-07-12	1.83%	14-08-12	0.98%
15-04-12	1.99%	15-05-12	0.81%	15-06-12	0.94%	15-07-12	3.01%	15-08-12	0.94%
16-04-12	1.50%	16-05-12	1.05%	16-06-12	1.86%	16-07-12	2.24%	16-08-12	1.16%
17-04-12	1.62%	17-05-12	2.46%	17-06-12	2.03%	17-07-12	1.20%	17-08-12	1.45%
18-04-12	1.36%	18-05-12	1.93%	18-06-12	2.09%	18-07-12	2.67%	18-08-12	1.73%
19-04-12	1.14%	19-05-12	1.69%	19-06-12	1.54%	19-07-12	2.98%	19-08-12	2.54%
20-04-12	1.51%	20-05-12	2.79%	20-06-12	1.98%	20-07-12	1.19%	20-08-12	1.35%
21-04-12	1.55%	21-05-12	2.26%	21-06-12	1.36%	21-07-12	1.13%	21-08-12	1.24%
22-04-12	3.26%	22-05-12	1.68%	22-06-12	1.37%	22-07-12	2.93%	22-08-12	1.28%
23-04-12	1.31%	23-05-12	2.11%	23-06-12	2.64%	23-07-12	3.66%	23-08-12	1.10%
24-04-12	1.80%	24-05-12	0.88%	24-06-12	1.19%	24-07-12	1.18%	24-08-12	1.07%
25-04-12	1.51%	25-05-12	1.32%	25-06-12	3.03%	25-07-12	1.51%	25-08-12	1.41%
26-04-12	1.31%	26-05-12	1.26%	26-06-12	1.91%	26-07-12	1.02%	26-08-12	4.33%
27-04-12	1.38%	27-05-12	1.54%	27-06-12	1.60%	27-07-12	3.71%	27-08-12	1.53%
28-04-12	1.14%	28-05-12	2.09%	28-06-12	1.28%	28-07-12	11.58%	28-08-12	1.24%
29-04-12	1.96%	29-05-12	3.27%	29-06-12	7.73%	29-07-12	5.36%	29-08-12	1.05%
30-04-12	2.60%	30-05-12	1.18%	30-06-12	7.60%	30-07-12	1.44%	30-08-12	5.05%
		31-05-12	1.87%			31-07-12	1.41%	31-08-12	2.29%
MAPE ABRIL	2.27%	MAPE MAYO	2.26%	MAPE JUNIO	2.18%	MAPE JULIO	2.34%	MAPE AGOSTO	2.13%

El MAPE promedio de las proyecciones de la RNA con 3 capas ocultas, correspondiente a los meses de abril, mayo, junio, julio, y agosto fue 2.24%. Con esta arquitectura se obtienen resultados cercanos al MAPE requerido. Los días feriados y contiguos a los feriados incrementan el error de la proyección.

Con esta arquitectura de RNA se obtuvieron proyecciones cuyos MAPE fueron menores al 1%, los cuales se detallan a continuación:

- Mes de Mayo: 05.05.12 (MAPE = 0.97%); 11.05.12 (MAPE = 0.99%); 15.05.12 (MAPE = 0.81%) y 24.05.12 (MAPE = 0.88%)

- Mes de Junio: 07.06.12 (MAPE = 0.97%) 12.06.12 (MAPE = 0.84%) y 15.06.12 (MAPE = 0.94%)
- Mes de Agosto: 07.08.12 (MAPE = 0.90%); 08.08.12 (MAPE = 0.89%); 09.08.12 (MAPE = 0.95%); 14.08.12 (MAPE = 0.98%) y 15.08.12 (MAPE = 0.94%);

Estas proyecciones diarias son mejores, pues los valores proyectados son muy cercanos a los reales. Lo cual demuestra que la RNA con 3 capas ocultas es la mejor arquitectura para este tipo de proyecciones.

A continuación se muestran algunos resultados correspondiente a los días 15 de junio (MAPE = 0.94%) y 8 de agosto del 2012 (MAPE = 0.89%).

En la proyección del día 15 de junio, se presenta una pequeña desviación entre las 11:30 h y 13:30 h, el cual representa un error de la proyección de demanda del 2%, luego de las 13:30 h, el error de proyección se mantiene por debajo del 0.5%.

En la proyección del día 8 de agosto, se presenta una pequeña desviación entre las 18:30 h y 20:30 h, el cual representa un error de proyección de demanda del -2.5%, luego el error de proyección baja hasta 1.0%.

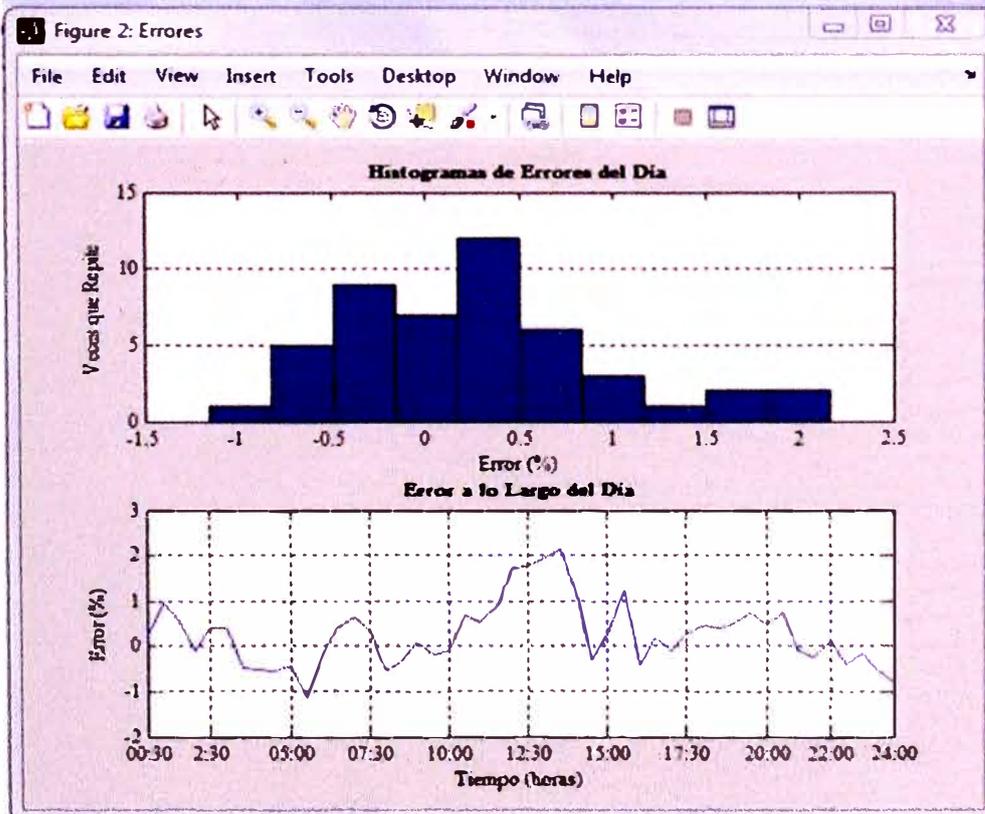
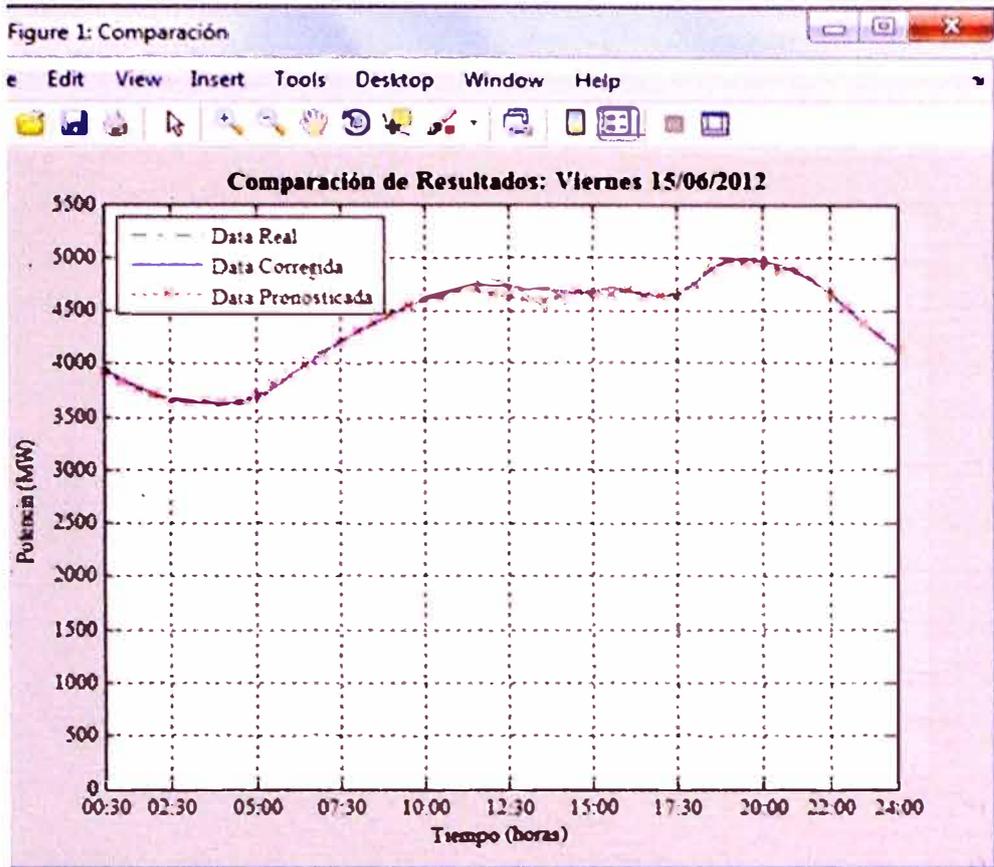


Fig. N° 5.9 Proyección del 15 junio 2012 con histograma de errores
(Fuente: elaboración propia)

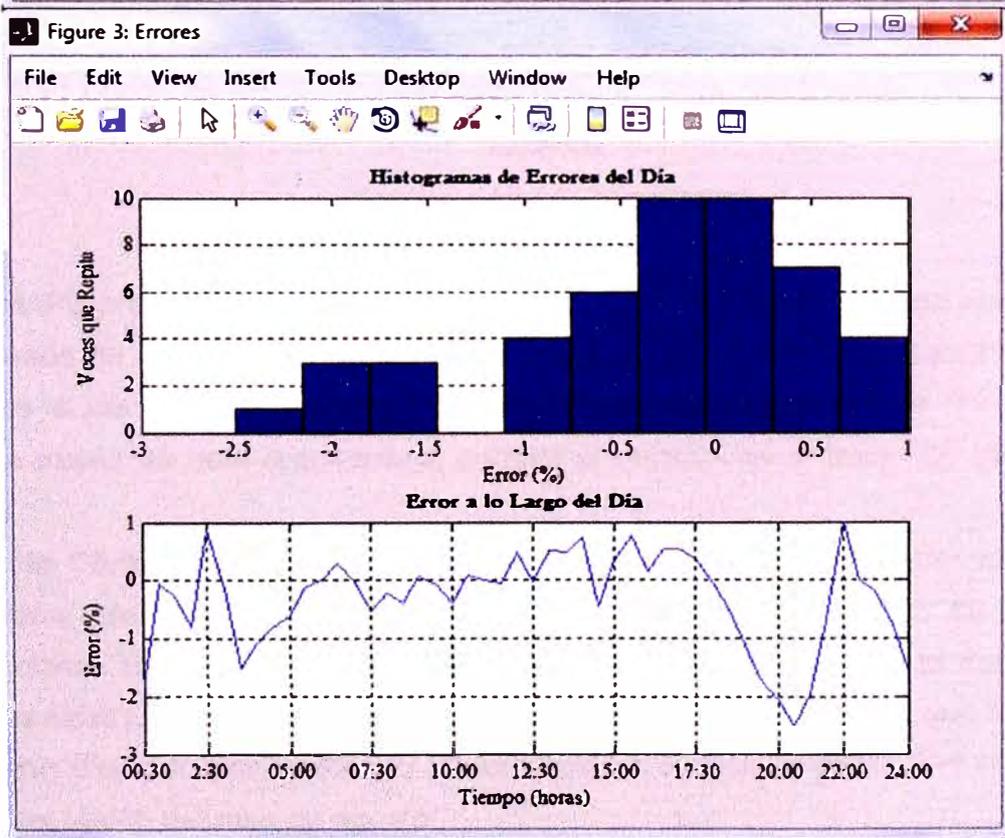
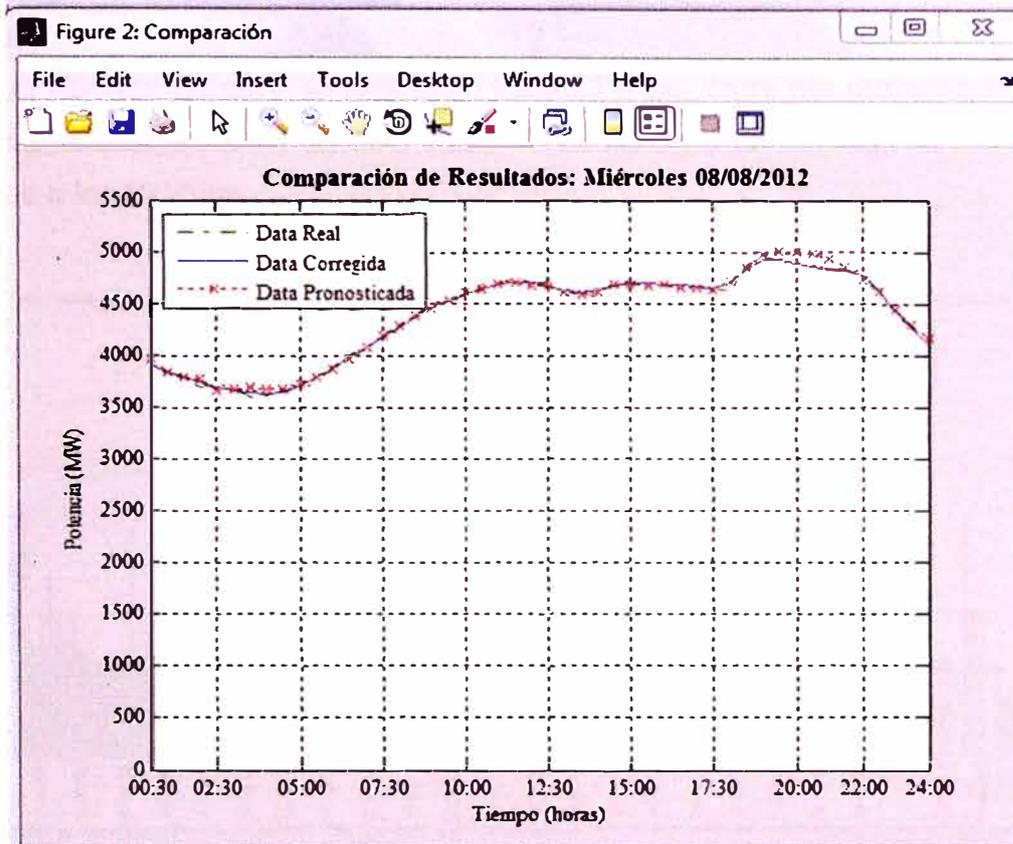


Fig. N° 5.10 Proyección del 08 agosto 2012 con histograma de errores
(Fuente: elaboración propia)

5.4.3 Proyección diaria usando RNA con 3 capas ocultas sin considerar días feriados

En el siguiente gráfico se muestran los MAPE por cada día proyectado usando RNA con 3 capas ocultas, pero en este caso no se consideran los días feriados, ni los días contiguos a los feriados.

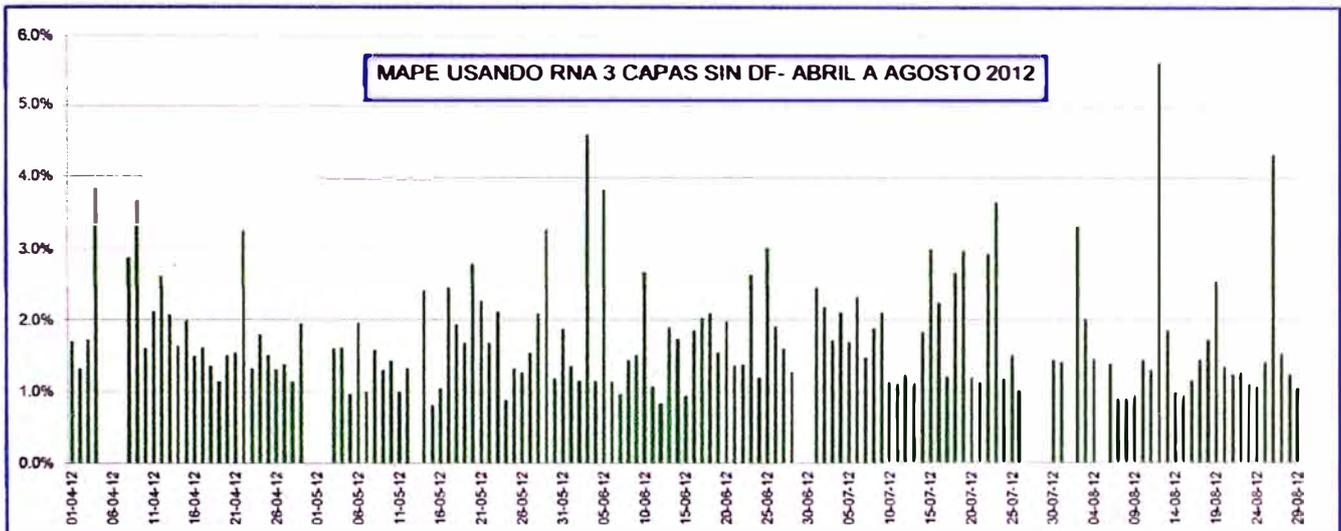


Fig. N° 5.11 MAPE de proyección del SEIN usando RNA con 3 capas sin considerar días feriados

(Fuente: elaboración propia)

El MAPE promedio de la proyección de la RNA con 3 capas ocultas, correspondiente a los meses de abril, mayo, junio, julio, y agosto sin considerar los días feriados y días contiguos a los feriados fue 1.75%. Podemos afirmar entonces que las proyecciones diarias a través de esta arquitectura, cumple el objetivo de la tesis, sin considerar días feriados.

El día 05.08.12 se realizó la interconexión del sistema peruano con el sistema ecuatoriano, desde las 04:38 h a 18:00 h, se transfirió aprox. 80 MW de las subestaciones Emeloro 1 y 2, del sistema Ecuatoriano al SEIN. Esta transferencia de demanda eléctrica de 80 MW adicionales al SEIN alteró la proyección que realizó la RNA para dicho día, por esta razón no tomaremos en cuenta el MAPE del 05.08.12 en el cálculo del MAPE del mes de agosto.

Así mismo para el cálculo del MAPE del mes de agosto no se incluyó la proyección de los días 01 y 12 de agosto, debido a la reducción de la demanda eléctrica no

programada de los usuarios libres mineras Antamina (no consumió 85 MW) y Yanacocha (no consumió 65 MW) respectivamente, por mantenimiento correctivo.

- Mes de Abril: MAPE = 1.90%
- Mes de Mayo: MAPE = 1.66%
- Mes de Junio: MAPE = 1.79%
- Mes de Julio: MAPE = 1.85%
- Mes de Agosto: MAPE = 1.53%
- MAPE Total (abril, mayo, junio, julio, agosto) = 1.75%

Tabla N° 5.3 MAPE de las proyecciones del SEIN usando RNA con 3 capas ocultas sin considerar días feriados
(Fuente: elaboración propia)

DIA	MAPE	DIA	MAPE	DIA	MAPE	DIA	MAPE	DIA	MAPE
01/04/12	1.70%	01/05/12	14.74%	01/06/12	1.36%	01/07/12	2.45%	01/08/12	6.48%
02/04/12	1.32%	02/05/12	7.25%	02/06/12	1.14%	02/07/12	2.18%	02/08/12	3.32%
03/04/12	1.72%	03/05/12	1.60%	03/06/12	4.61%	03/07/12	1.72%	03/08/12	2.02%
04/04/12	3.83%	04/05/12	1.62%	04/06/12	1.14%	04/07/12	2.10%	04/08/12	1.46%
05/04/12	6.27%	05/05/12	0.97%	05/06/12	3.83%	05/07/12	1.69%	05/08/12	6.60%
06/04/12	2.25%	06/05/12	1.97%	06/06/12	1.13%	06/07/12	2.32%	06/08/12	1.38%
07/04/12	7.54%	07/05/12	1.00%	07/06/12	0.97%	07/07/12	1.48%	07/08/12	0.90%
08/04/12	2.88%	08/05/12	1.58%	08/06/12	1.44%	08/07/12	1.88%	08/08/12	0.89%
09/04/12	3.66%	09/05/12	1.30%	09/06/12	1.51%	09/07/12	2.10%	09/08/12	0.95%
10/04/12	1.61%	10/05/12	1.42%	10/06/12	2.68%	10/07/12	1.12%	10/08/12	1.45%
11/04/12	2.12%	11/05/12	0.99%	11/06/12	1.08%	11/07/12	1.10%	11/08/12	1.30%
12/04/12	2.62%	12/05/12	1.32%	12/06/12	0.84%	12/07/12	1.23%	12/08/12	5.62%
13/04/12	2.08%	13/05/12	1.72%	13/06/12	1.90%	13/07/12	1.11%	13/08/12	1.86%
14/04/12	1.64%	14/05/12	2.41%	14/06/12	1.74%	14/07/12	1.83%	14/08/12	0.98%
15/04/12	1.99%	15/05/12	0.81%	15/06/12	0.94%	15/07/12	3.01%	15/08/12	0.94%
16/04/12	1.50%	16/05/12	1.05%	16/06/12	1.86%	16/07/12	2.24%	16/08/12	1.16%
17/04/12	1.62%	17/05/12	2.46%	17/06/12	2.03%	17/07/12	1.20%	17/08/12	1.45%
18/04/12	1.36%	18/05/12	1.93%	18/06/12	2.09%	18/07/12	2.67%	18/08/12	1.73%
19/04/12	1.14%	19/05/12	1.69%	19/06/12	1.54%	19/07/12	2.98%	19/08/12	2.54%
20/04/12	1.51%	20/05/12	2.79%	20/06/12	1.98%	20/07/12	1.19%	20/08/12	1.35%
21/04/12	1.55%	21/05/12	2.26%	21/06/12	1.36%	21/07/12	1.13%	21/08/12	1.24%
22/04/12	3.26%	22/05/12	1.68%	22/06/12	1.37%	22/07/12	2.93%	22/08/12	1.28%
23/04/12	1.31%	23/05/12	2.11%	23/06/12	2.64%	23/07/12	3.66%	23/08/12	1.10%
24/04/12	1.80%	24/05/12	0.88%	24/06/12	1.19%	24/07/12	1.18%	24/08/12	1.07%
25/04/12	1.51%	25/05/12	1.32%	25/06/12	3.03%	25/07/12	1.51%	25/08/12	1.41%
26/04/12	1.31%	26/05/12	1.26%	26/06/12	1.91%	26/07/12	1.02%	26/08/12	4.33%
27/04/12	1.38%	27/05/12	1.54%	27/06/12	1.60%	27/07/12	3.71%	27/08/12	1.53%
28/04/12	1.14%	28/05/12	2.09%	28/06/12	1.28%	28/07/12	11.58%	28/08/12	1.24%
29/04/12	1.96%	29/05/12	3.27%	29/06/12	7.73%	29/07/12	5.36%	29/08/12	1.05%
30/04/12	2.60%	30/05/12	1.18%	30/06/12	7.60%	30/07/12	1.44%	30/08/12	5.05%
		31/05/12	1.87%			31/07/12	1.41%	31/08/12	2.29%
MAPE ABRIL	1.90%	MAPE MAYO	1.66%	MAPE JUNIO	1.79%	MAPE JULIO	1.85%	MAPE AGOS1	1.53%

5.5 Comparación de modelos de proyecciones del COES y la red neuronal artificial

Una de las razones por las cuales se planteó realizar la proyección de la demanda eléctrica del SEIN, considerándola como una demanda eléctrica nodal, a través del nuevo método de Red Neuronal Artificial, es el hecho que podemos hacer la validación de los resultados con datos reales y compararlo con la proyección resultante que realiza el COES a través de su metodología la cual fue detallada en esta tesis.

Para comparar ambos métodos, tomaremos las proyecciones realizadas de abril a agosto del 2012. Tener en cuenta que el método de Red Neuronal Artificial que usaremos es de 3 capas ocultas y sin considerar días feriados.

En la tabla N° 5.4 se muestran los MAPE de las proyecciones diarias de los meses de abril, mayo y junio del 2012 a través de ambos métodos. Como se puede observar ambos métodos arrojan proyecciones de MAPE similares para los meses de abril y junio. El MAPE del mes de mayo usando RNA es 1.66% y por la metodología del COES es 1.92%, hay una diferencia con mayor precisión de proyección usando el método de RNA.

En la figura N° 5.12 se muestra el resumen comparativo de los MAPE de las proyecciones de la demanda a través de la RNA y el método del COES correspondientes a las proyecciones del mes de abril del 2012.

En la figura N° 5.13 se muestra el resumen comparativo de los MAPE de las proyecciones de la demanda a través de la RNA y el método del COES correspondientes a las proyecciones del mes de mayo del 2012.

En la figura N° 5.14 se muestra el resumen comparativo de los MAPE de las proyecciones de la demanda a través de la RNA y el método del COES correspondientes a las proyecciones del mes de junio del 2012.

Tabla N° 5.4 Cuadro Comparativo de MAPE (abril-mayo-junio) usando RNA y metodología del COES
(Fuente: elaboración propia)

DIA	COES	RNA	DIA	COES	RNA	DIA	COES	RNA
01/04/12	1.91%	1.70%	01/05/12	1.76%	14.74%	01/06/12	1.82%	1.36%
02/04/12	1.85%	1.32%	02/05/12	8.12%	7.25%	02/06/12	0.88%	1.14%
03/04/12	3.49%	1.72%	03/05/12	3.14%	1.60%	03/06/12	1.17%	4.61%
04/04/12	1.37%	3.83%	04/05/12	0.90%	1.62%	04/06/12	1.51%	1.14%
05/04/12	6.03%	6.27%	05/05/12	1.55%	0.97%	05/06/12	3.81%	3.83%
06/04/12	4.59%	2.25%	06/05/12	1.89%	1.97%	06/06/12	1.49%	1.13%
07/04/12	4.30%	7.54%	07/05/12	0.93%	1.00%	07/06/12	1.26%	0.97%
08/04/12	2.84%	2.88%	08/05/12	1.19%	1.58%	08/06/12	1.12%	1.44%
09/04/12	1.49%	3.66%	09/05/12	1.37%	1.30%	09/06/12	1.81%	1.51%
10/04/12	0.83%	1.61%	10/05/12	1.35%	1.42%	10/06/12	0.89%	2.68%
11/04/12	1.58%	2.12%	11/05/12	1.41%	0.99%	11/06/12	1.35%	1.08%
12/04/12	1.14%	2.62%	12/05/12	1.83%	1.32%	12/06/12	1.47%	0.84%
13/04/12	0.87%	2.08%	13/05/12	5.14%	1.72%	13/06/12	1.93%	1.90%
14/04/12	0.95%	1.64%	14/05/12	3.56%	2.41%	14/06/12	1.31%	1.74%
15/04/12	0.92%	1.99%	15/05/12	0.98%	0.81%	15/06/12	1.20%	0.94%
16/04/12	1.46%	1.50%	16/05/12	1.40%	1.05%	16/06/12	0.87%	1.86%
17/04/12	1.68%	1.62%	17/05/12	1.52%	2.46%	17/06/12	2.15%	2.03%
18/04/12	1.75%	1.36%	18/05/12	1.86%	1.93%	18/06/12	2.40%	2.09%
19/04/12	1.47%	1.14%	19/05/12	1.05%	1.69%	19/06/12	0.84%	1.54%
20/04/12	3.05%	1.51%	20/05/12	1.18%	2.79%	20/06/12	0.99%	1.98%
21/04/12	1.80%	1.55%	21/05/12	1.36%	2.26%	21/06/12	0.96%	1.36%
22/04/12	1.18%	3.26%	22/05/12	2.07%	1.68%	22/06/12	1.14%	1.37%
23/04/12	1.62%	1.31%	23/05/12	1.16%	2.11%	23/06/12	2.00%	2.64%
24/04/12	1.34%	1.80%	24/05/12	1.10%	0.88%	24/06/12	2.32%	1.19%
25/04/12	0.84%	1.51%	25/05/12	0.77%	1.32%	25/06/12	1.89%	3.03%
26/04/12	1.49%	1.31%	26/05/12	0.64%	1.26%	26/06/12	2.60%	1.91%
27/04/12	1.43%	1.38%	27/05/12	1.05%	1.54%	27/06/12	2.47%	1.60%
28/04/12	1.21%	1.14%	28/05/12	2.87%	2.09%	28/06/12	1.33%	1.28%
29/04/12	0.75%	1.96%	29/05/12	3.79%	3.27%	29/06/12	2.19%	7.73%
30/04/12	1.99%	2.60%	30/05/12	1.34%	1.18%	30/06/12	4.90%	7.60%
			31/05/12	1.10%	1.87%			
MAPE ABRIL	1.91%	1.90%	MAPE MAYO	1.92%	1.66%	MAPE JUNIO	1.74%	1.79%

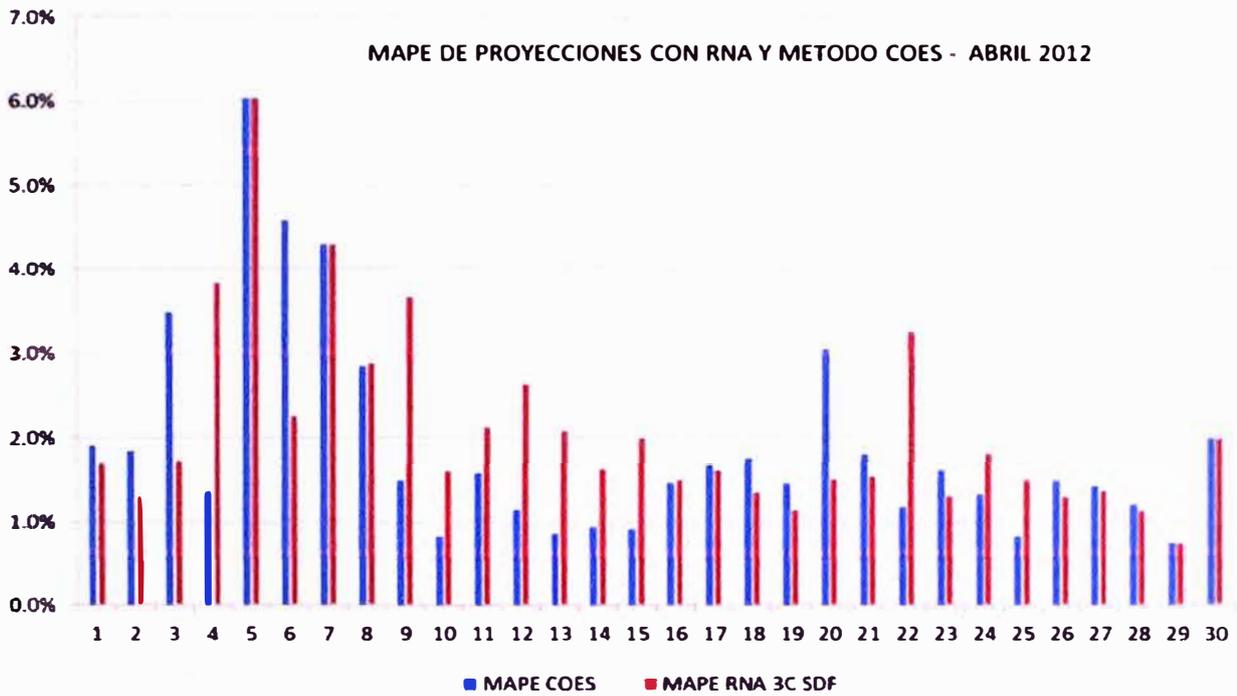


Fig. N° 5.12 Comparativo de MAPE de abril usando RNA y método COES
(Fuente: elaboración propia)

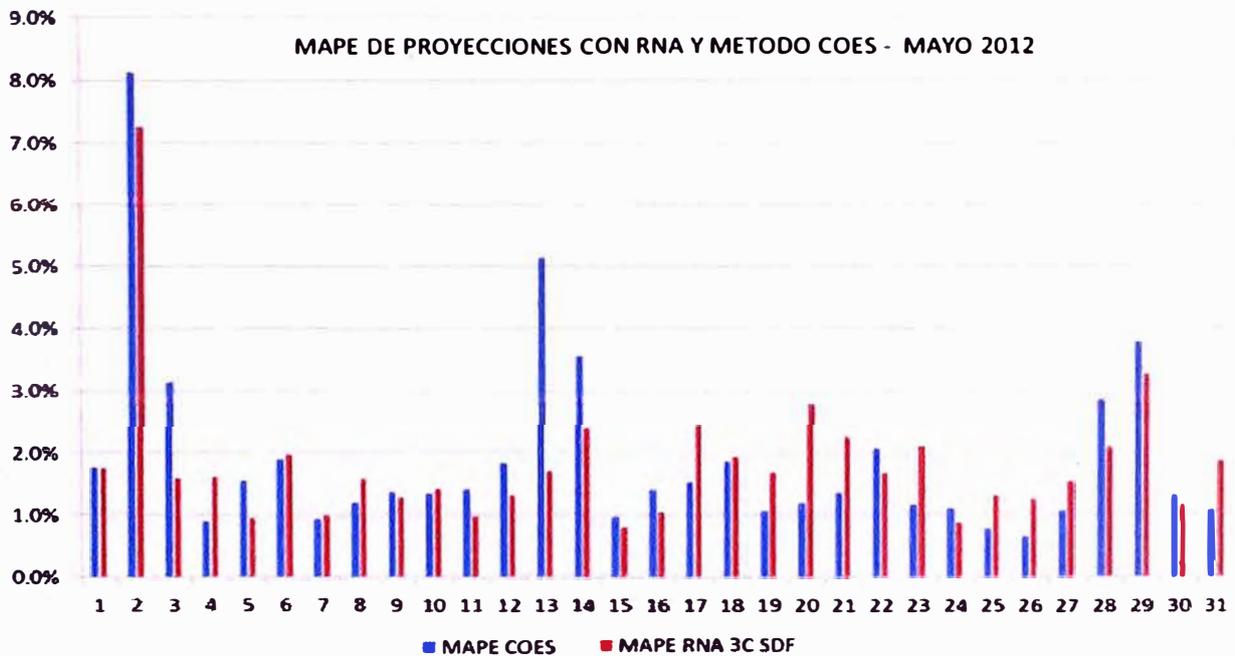


Fig. N° 5.13 Comparativo de MAPE de mayo usando RNA y método COES
(Fuente: elaboración propia)

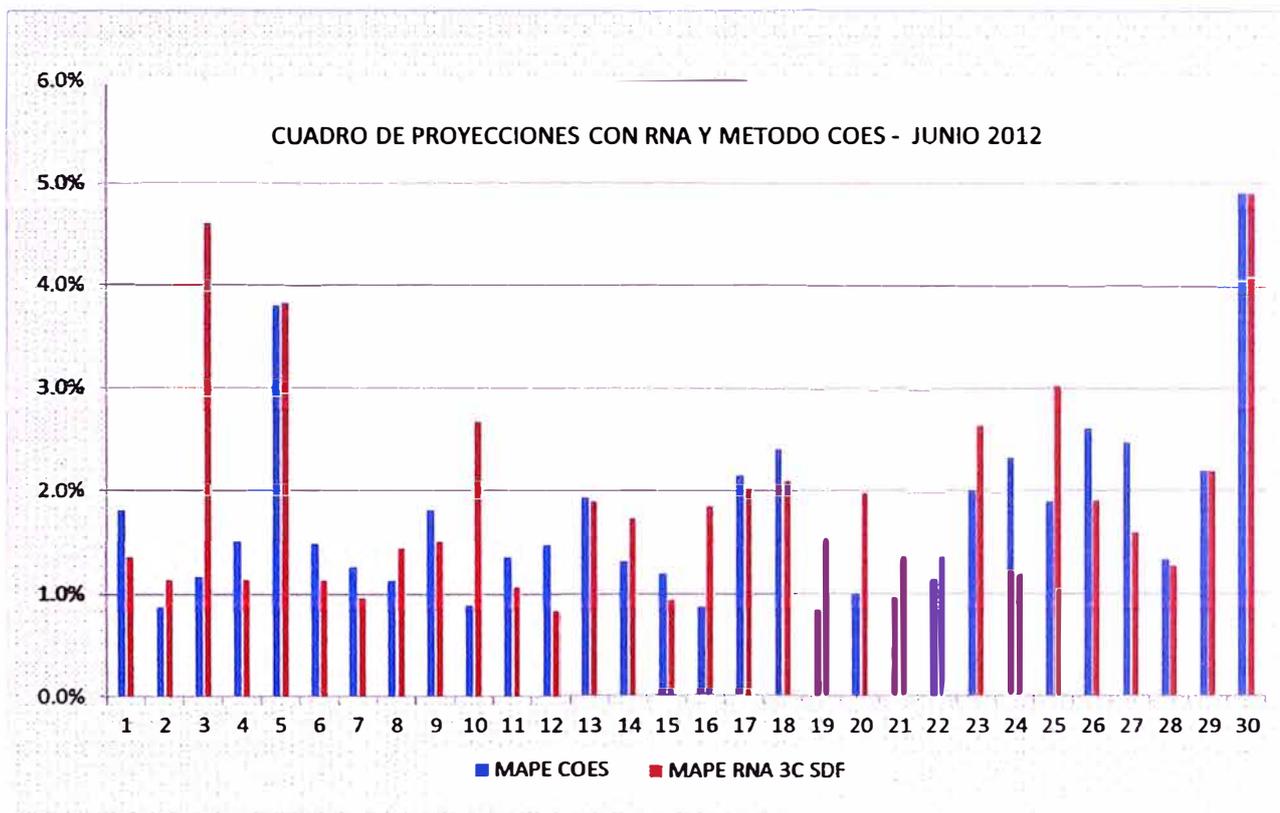


Fig. N° 5.14 Comparativo de MAPE de junio usando RNA y método COES
(Fuente: elaboración propia)

En la tabla N° 5.5 se muestran los MAPE de las proyecciones diarias de los meses julio y agosto del 2012 a través de ambos métodos. Como se puede observar ambos métodos arrojan proyecciones de MAPE similares.

En la figura N° 5.15 se muestra el resumen comparativo de los MAPE de las proyecciones a través de la RNA y el método del COES correspondientes a las proyecciones del mes de julio del 2012.

En la figura N° 5.16 se muestra el resumen comparativo de los MAPE de las proyecciones a través de la RNA y el método del COES correspondientes a las proyecciones del mes de agosto del 2012.

Tabla N° 5.5 Cuadro Comparativo de MAPE (julio - agosto) usando RNA y metodología del COES

(Fuente: elaboración propia)

DIA	COES	RNA	DIA	COES	RNA
01/07/12	2.76%	2.45%	01/08/12	1.73%	6.48%
02/07/12	1.21%	2.18%	02/08/12	2.18%	3.32%
03/07/12	2.57%	1.72%	03/08/12	1.11%	2.02%
04/07/12	2.26%	2.10%	04/08/12	1.39%	1.46%
05/07/12	1.96%	1.69%	05/08/12	0.85%	6.60%
06/07/12	1.25%	2.32%	06/08/12	1.26%	1.38%
07/07/12	0.98%	1.48%	07/08/12	1.75%	0.90%
08/07/12	1.75%	1.88%	08/08/12	1.29%	0.89%
09/07/12	1.59%	2.10%	09/08/12	1.23%	0.95%
10/07/12	1.75%	1.12%	10/08/12	1.69%	1.45%
11/07/12	1.41%	1.10%	11/08/12	1.47%	1.30%
12/07/12	0.67%	1.23%	12/08/12	2.18%	5.62%
13/07/12	1.32%	1.11%	13/08/12	1.24%	1.86%
14/07/12	1.08%	1.83%	14/08/12	1.28%	0.98%
15/07/12	3.51%	3.01%	15/08/12	1.41%	0.94%
16/07/12	1.55%	2.24%	16/08/12	1.73%	1.16%
17/07/12	1.57%	1.20%	17/08/12	1.37%	1.45%
18/07/12	0.79%	2.67%	18/08/12	1.37%	1.73%
19/07/12	1.33%	2.98%	19/08/12	1.46%	2.54%
20/07/12	1.50%	1.19%	20/08/12	0.96%	1.35%
21/07/12	1.06%	1.13%	21/08/12	1.04%	1.24%
22/07/12	1.68%	2.93%	22/08/12	1.77%	1.28%
23/07/12	0.96%	3.66%	23/08/12	0.90%	1.10%
24/07/12	2.20%	1.18%	24/08/12	1.10%	1.07%
25/07/12	1.30%	1.51%	25/08/12	1.22%	1.41%
26/07/12	1.09%	1.02%	26/08/12	1.18%	4.33%
27/07/12	2.84%	3.71%	27/08/12	1.29%	1.53%
28/07/12	3.63%	11.58%	28/08/12	1.20%	1.24%
29/07/12	1.84%	5.36%	29/08/12	0.91%	1.05%
30/07/12	1.76%	1.44%	30/08/12	2.27%	5.05%
31/07/12	2.72%	1.41%	31/08/12	1.77%	2.29%
MAPE JULIO	1.74%	1.85%	MAPE AGOSTO	1.41%	1.53%

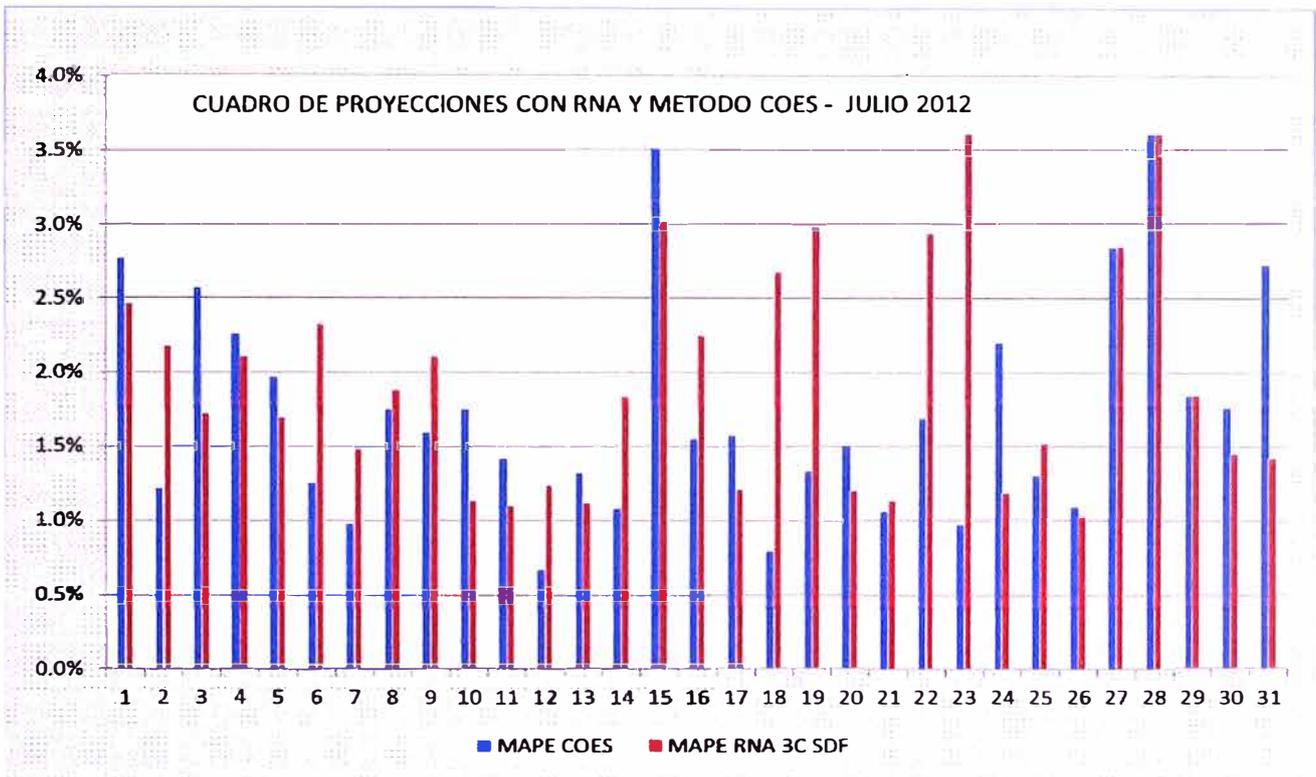


Fig. N° 5.15 Comparativo de MAPE de julio usando RNA y método COES
(Fuente: elaboración propia)

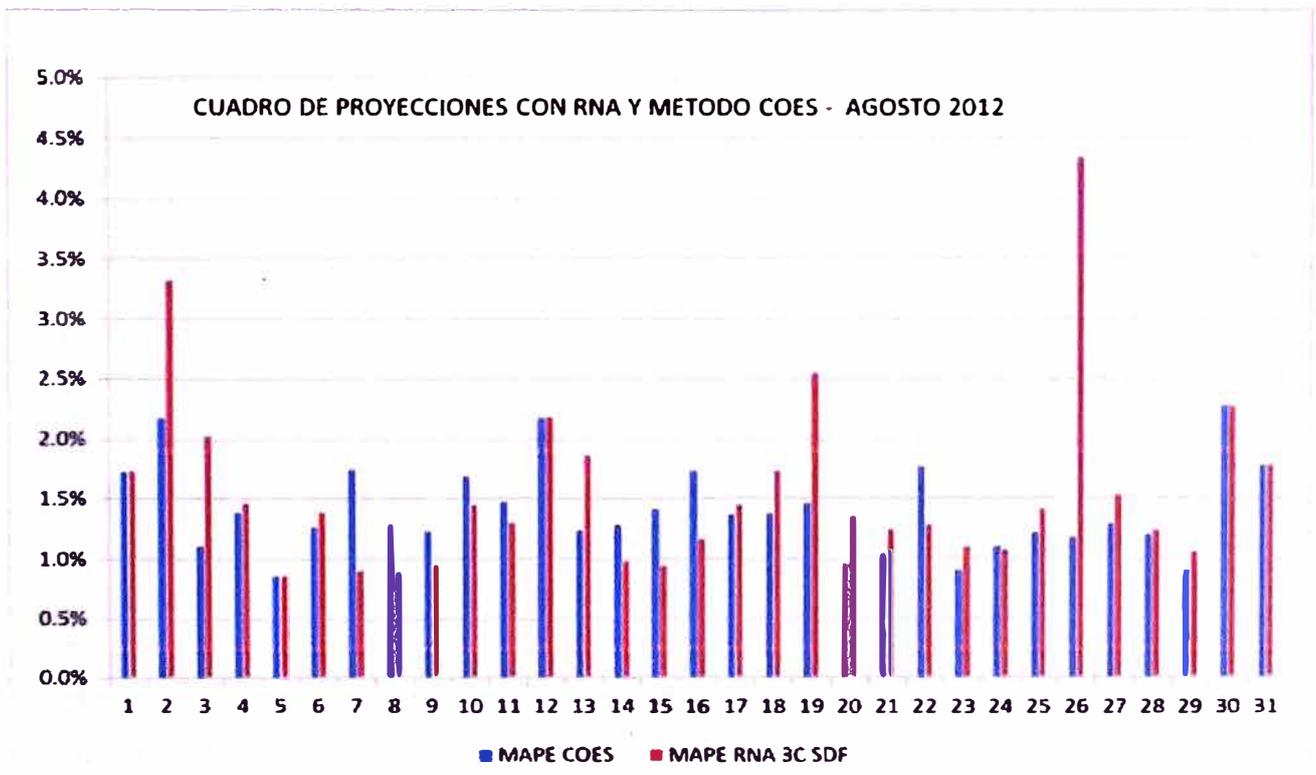


Fig. N° 5.16 Comparativo de MAPE de agosto usando RNA y método COES
(Fuente: elaboración propia)

De los resultados obtenidos con ambos métodos de proyección, se obtienen similares MAPE. Sin embargo existen factores de proyección diaria que de manera común siempre va existir y se aplica actualmente en la proyección diaria del COES, los cuales están relacionados a varios aspectos, tales como: la variación de la demanda eléctrica de los usuarios libres (como lo ocurrido los días 01 y 12 de agosto), las transferencias atípicas de demanda por interconexión del SEIN con el sistema eléctrico ecuatoriano (como lo ocurrido el 05 de agosto), las reducciones de demanda de usuarios regulados por días festivos declarados por el estado peruano, los movimientos sociales como paros o elecciones municipales y regionales, entre otros. Estos factores alteran la proyección de la Red Neuronal Artificial, incrementando el error de aproximación a través del indicador MAPE.

5.6 Reducción del costo de operación

5.6.1 Premisas para cuantificar los nuevos costos de operación

A continuación se detalla las premisas que se tomaron en cuenta para realizar la cuantificación de la reducción del costo de operación:

- Cuantificar el beneficio obtenido para los días en que se obtuvieron mejores aproximaciones usando el método de la RNA en comparación con el método **tradicional del COES**.
- Se considera los días del mes de mayo del 2012 con mejor estimación de la demanda, las cuales fueron obtenidos usando el método de proyección de la RNA en comparación con el método tradicional del COES.
- Se **cuantifica el beneficio para aquellos días**, en los cuales el centro de control del COES realizó reprogramas diarios por mayor o menor demanda eléctrica del SEIN, con la finalidad corregir las desviaciones entre lo programado y lo ejecutado.
- Para proceder con la cuantificación de la reducción del costo de operación, se ha **tomado como referencia el costo diario de la operación del SEIN**. El cual es el costo del despacho de las centrales de generación, que también abarca a los costos de arranque y parada de las centrales térmicas.
- El día seleccionado para cuantificar la reducción del costo de operación, debe ser **aquel cuyo costo no haya sido variado por perturbaciones relevantes o indisponibilidades de centrales de generación que hayan provocado que se altere el costo operativo diario del SEIN**.

A continuación se detalla la relación de los días del mes de mayo del 2012 seleccionados para que sean cuantificados:

Tabla N° 5.6 Relación de días para cuantificar el costo de operación
(Fuente: Comité de operación económica del sistema interconectado nacional)

FECHA	VARIACION DE LA DEMANDA	MAPE DIARIO	
		PROYECCION COES	PROYECCION RNA
03/05/12	Menor Demanda	3.14%	1.60%
05/05/12	Mayor Demanda	1.55%	0.97%
12/05/12	Mayor Demanda	1.83%	1.32%
13/05/12	Mayor Demanda	5.14%	1.72%
29/05/12	Menor Demanda	3.79%	3.27%
30/05/12	Mayor demanda	1.34%	1.18%

5.6.2 Resultados obtenidos de la reducción del costo de operación

El ahorro obtenido por realizar una mejor proyección de la demanda diaria a través del método de Redes Neuronales Artificiales, se obtiene de las diferencias de los resultados de los costos de operación de las centrales de generación usando la demanda del SEIN, con las proyecciones que realizó el COES y con las proyecciones de las Redes Neuronales Artificiales.

Los costos de operación se obtuvieron de las simulaciones del despacho de las centrales de generación usando el software NCP, con las proyecciones que usó el COES y con las proyecciones de las Redes Neuronales Artificiales. Estas simulaciones se realizaron en las mismas condiciones solo alterando la demanda diaria para cada escenario.

A continuación se muestran los resultados obtenidos, con el detalle del ahorro o beneficio de realizar una mejor aproximación diaria.

Tabla N° 5.7 Cuantificación del ahorro obtenido por mejor proyección de la RNA
(Fuente: elaboración propia)

FECHA	VARIACION DE LA DEMANDA	COSTO DE OPERACIÓN DIARIO DEL SEIN (MILES US\$)			
		PROYECCIÓN COES	PROYECCION RNA	EJECUTADO	AHORRO
03/05/12	Menor Demanda	652.820	575.530	580.182	67.986
05/05/12	Mayor Demanda	535.440	565.970	618.264	30.530
12/05/12	Mayor Demanda	601.090	669.030	659.534	48.949
13/05/12	Mayor Demanda	456.760	578.750	553.129	70.747
29/05/12	Menor Demanda	861.050	885.500	1079.190	24.450
30/05/12	Mayor demanda	834.960	1026.700	1257.305	191.740

CONCLUSIONES

1. Los resultados de las proyecciones de las demandas diarias del SEIN usando Redes Neuronales Artificiales cumplen con el objetivo de la presente tesis, porque se ha encontrado errores medios porcentuales absolutos (MAPE) menores del 2%. Esta metodología de proyección no aplica a las proyecciones de los días feriados. Para algunos casos se ha obtenido resultados con MAPE menores del 1%.
2. Una de las razones por las cuales se planteó realizar la proyección de la demanda eléctrica del SEIN, considerándola como demanda eléctrica nodal, es el hecho que podemos hacer la validación de los resultados con datos reales y compararlo con la proyección diaria que realiza el COES a través de su metodología actual. Los datos reales y las proyecciones diarias de la demanda del SEIN se obtuvieron del portal web del COES.
3. Al comparar los resultados de las proyecciones entre la RNA y el método del COES, para los meses de abril a agosto del 2012, se obtuvieron errores medios porcentuales absolutos similares de 1.75% y 1.74% respectivamente, lo cual significa que la metodología propuesta en esta tesis podría ser usada para las proyecciones que usa el COES. Para el caso del mes de mayo las proyecciones de la RNA obtuvo mejores resultados con un MAPE de 1.66% respecto a la proyección del COES con un MAPE de 1.92%.
4. La proyección diaria de la demanda en una barra o en una determinada área operativa de un sistema eléctrico de potencia, siempre estará sujeto a variables externas que influyen en la proyección de la demanda. Estas variables externas, normalmente son tomadas en cuenta por el programador que realiza la proyección diaria. En las proyecciones de las Redes Neuronales Artificiales que se realizaron en la presente tesis no se toman en cuenta estas variables externas, debido a que son proyecciones de días ya ejecutados. Si el programador usara la Red Neuronal Artificial para su proyección diaria debería de tomar en cuentas estas variables

externas, lo cual permitiría reducir el error medio porcentual absoluto obtenido en las simulaciones presentadas en este trabajo.

5. En relación a las proyecciones de los días feriados, esta es compleja de predecir, debido a que no se tiene un patrón de tendencia de la demanda eléctrica e información histórica suficiente para cada día feriado. Así mismo la proyección de los días contiguos al día feriado también se vuelve compleja, comúnmente la demanda antes y después del día festivo disminuye en relación a un día normal. Actualmente el COES no tiene una metodología definida para la proyección de los días feriados, sus proyecciones lo realiza en base a factores de proporción de días feriados pasados.
6. La proyección de la demanda a través de la Red Neuronal Artificial permite crear una base de datos para una determina barra en estudio, el cual debe seguir actualizándose con información histórica del día a día. Es necesario contar con herramientas que permitan analizar la información histórica para validar las hipótesis sobre las condiciones del mercado peruano y construir pronósticos más acertados.
7. Para realizar las proyecciones de demanda eléctrica de un día, cuya particularidad es que en la semana anterior y/o el día anterior a su proyección fue feriado, se debe realizar un ajuste a dichas demandas históricas, con la finalidad de no inyectar errores en su proyección.
8. Los resultados obtenidos muestran que los mayores errores de proyección se presentan en aquellos meses que se presentaron más días feriados o feriados largos declarados por el estado. En resumen se podría afirmar que las demandas históricas de los días feriados producen errores de cálculo en las proyecciones.
9. El pronóstico del perfil de demanda es una tarea difícil. Primero, porque los perfiles de demanda son complejas y presentan diferentes niveles de estacionalidad. Segundo, porque hay muchas variables exógenas importantes que deben ser consideradas, especialmente las relacionadas con el clima, así se demuestra que el perfil de demanda de un día, para diferentes estaciones del año es diferente.
10. Una desviación grande en el pronóstico de demanda puede implicar grandes costos de operación, por lo cual se debe tener especial cuidado con el uso de los modelos. Es así que se ha procedido a cuantificar los errores de proyección usando el método

actual del COES. Los resultados de las simulaciones por errores en la proyección de la demanda, han arrojado pérdidas de hasta 190 mil dólares americanos.

11. **Las características principales del perfil de demanda de corto plazo son la tendencia, estacionalidad, ciclos y la irregularidad, además dicha serie tiene una fuerte influencia de variables económicas y climáticas.**

12. **De la literatura revisada podemos afirmar que el uso de modelos no lineales para el estudio del pronóstico de la demanda aún no ha sido completamente estudiado, ya que no se ha agotado la lista de los modelos posibles para emplearse. No existen razones empíricas, metodológicas o teóricas para preferir un modelo específico entre varias alternativas.**

RECOMENDACIONES

1. Debido al dinamismo del sistema eléctrico de potencia, se debería realizar más simulaciones a la red neuronal artificial de esta tesis, con información histórica más actualizada. Para lo cual se deberá seguir el mismo procedimiento de prueba-error con la finalidad de encontrar la red neuronal con cierta cantidad de capas ocultas, que arroje los mejores resultados en la proyección.
2. Con la finalidad de mejorar las proyecciones de la demanda eléctrica usando redes neuronales artificiales, se recomienda incrementar la cantidad de neuronas de entrada con 48 datos de ingreso, los cuales corresponden al día típico de la semana antepasada. Esta variación en la topología de la red neuronal original, ocasionará que el proceso de entrenamiento sea más duradero.
3. Se puede variar la arquitectura de la red neuronal reagrupando los datos de ingreso en los siete días típicos de la semana, con finalidad de tener un mejor perfil de día típico a proyectar.
4. Se comprobó que la demanda eléctrica a corto plazo, depende de la temperatura. Sin embargo para las simulaciones de la red neuronal, esta no se consideró como una variable de ingreso, debido a que no se dispone de información histórica de la temperatura y la geografía de todo el territorio peruano es diferente y variable. En caso se requiera hacer la proyección de la demanda de un área geográfica que tenga una temperatura definida y se cuente con información histórica, se recomienda que para la proyección de la demanda se incluya a la temperatura como dato de ingreso.

BIBLIOGRAFÍA

- [1] **Da Silva, A. A., Mouli, L. S. (2000).** Confidence Intervals for Neural Network Based Short-Term Load Forecasting. IEEE TRANSACTIONS ON POWER SYSTEMS, Vol. 15, N° 4. PP 6.

- [2] **Da Silva, W. L. (1996).** Previsao de Carga no Período de Demanda de Ponta Utilizando Redes Neurais Artificiais. Universidad Estatal de Campinas. Brasil. PP 151.

- [3] **María, V., Velasquez, J. D., Franco, C. J. (2011).** Avances recientes en la predicción de la demanda de electricidad usando modelos no lineales. Universidad Nacional de Colombia. Colombia. PP 8.

- [4] **De Andrade, F. L. (1997).** Aplicacao de Redes Neurais Artificiais na Previsao de Curto Prazo da Carga Nodal de Sistemas de Energia Elétrica. Universidad Estatal de Campinas. Brasil PP 191.

- [5] **Izaurieta, F., Saavedra, C. (2001).** Redes Neuronales Artificiales. Universidad de Concepción de Chile. Chile. PP 15.

- [6] **Steinherz, H., Pedreira, C. E., Castro, R. (2001).** Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE TRANSACTIONS ON POWER SYSTEMS. PP 12.

- [7] **Bertona, L. F. (2005).** Entrenamiento de redes neuronales basado en algoritmos Evolutivos. Universidad de Buenos Aires. Tesis de grado. Argentina. PP 253.

- [8] **Molina, Y. (2010).** Implementación de un método para previsión de carga en corto plazo. Instituto de Investigación de la Universidad Nacional de Ingeniería. Proyecto de investigación. Perú. PP 25.

- [9] **Charytoniuk, W., Chen, M.** (2000). Very Short-Term Load Forecasting Using Artificial Neural Networks. IEEE TRANSACTIONS ON POWER SYSTEMS, Vol. 15, N° 4. PP 6.
- [10] **Gomes, R., Bernardon, D., Garcia, V.** (2000). Metodología para previsao de carga em curtissimo prazo considerando variaveis climáticas. Universidad Federal de Pampa. Brasil PP 4.
- [11] **Basogain, X.** (2001.) Redes nueronales artificiales y sus aplicaciones. Escuela superior de Ingeniería de Bilbao. España. PP 79.

ANEXOS

FUNCIONES Y LÓGICAS DE PROGRAMACION EN MATLAB

A. FUNCION CREATE_FIT_NET

```
function [net,outputs] = create_fit_net(inputs,targets)
%inputs=single(inputs);
%targets=single(targets);
%CREATE_FIT_NET Creates and trains a fitting neural network.
% NET = CREATE_FIT_NET(INPUTS,TARGETS) takes these arguments:
% INPUTS - RxQ matrix of Q R-element input samples
% TARGETS - SxQ matrix of Q S-element associated target samples
% arranged as columns, and returns these results:
% NET - The trained neural network
% For example, to solve the Simple Fit dataset problem with this function:
% load simplefit_dataset
% net = create_fit_net(simplefitInputs,simplefitTargets);
% simplefitOutputs = sim(net,simplefitInputs);
% To reproduce the results you obtained in NFTOOL:
% net = create_fit_net(entrada',anox');

% Create Network
numHiddenNeurons = [90 80 50];% 15]; % Adjust as desired
%net =newelm(inputs,targets,numHiddenNeurons);
net = newelm(inputs,targets,numHiddenNeurons);
    %fitnet
    %newpr: mas o menos
    %newcf
    %newfit
% Network Eficiente
%net = initlay(net);
%net = initwb(net,1);
net.divideParam.trainRatio = 100/100;% Adjust as desired
```

```

net.divideParam.valRatio = 0/100; % Adjust as desired
net.divideParam.testRatio = 0/100; % Adjust as desired

% Network Parámetros de Entrenamiento
net.adaptFcn='tribas';
%net = adapt(net,inputs);
% Network Parámetros de división de datos
%net.divideFcn='divideblock';
net.trainFcn = 'trainscg'; %'trainlm';
%'trainrp'; rapido p=10
%'traingda'; mas o menos
%'traincgp'; rapido p=9 pero inefic
%'traincgf'; rapido p=9 pero inefic
%'traincgb'; rapido p=9 pero inefic
%'trainb'; rapido p=9
%'trainscg'; rapido p=11pero inefic
net.trainParam.max_fail=2147483646; %máximo número de validaciones
net.trainParam.epochs=2147483646; %epochs (2147483647 número máximo de
iteraciones)
net.trainParam.goal = 1.0e-1; %2.7e-4; min Performance (precisión) para newelm
poner máximo 5.0e-4
net.trainParam.min_grad= 1.0e-1; %2.0e-6; min performance gradient (gradiente)
%net.trainParam.mem_reduc = 10; %Factor to use for memory/speed tradeoff
% Train and Apply Network
[net,tr] = train(net,inputs,targets);
outputs= sim(net,inputs);

% Plot
plotperf(tr)
%plotfit(net,inputs,targets)
%plotregression(targets,outputs)

```

B. FUNCION ENTRENAMIENTO

```
function varargout = Entrenamiento(varargin)
% ENTRENAMIENTO M-file for Entrenamiento.fig
%   ENTRENAMIENTO, by itself, creates a new ENTRENAMIENTO or raises the
existing
%   singleton*.
%   H = ENTRENAMIENTO returns the handle to a new ENTRENAMIENTO or the
handle to
%   the existing singleton*.
%   ENTRENAMIENTO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in ENTRENAMIENTO.M with the given input
arguments.
%   ENTRENAMIENTO('Property','Value',...) creates a new ENTRENAMIENTO or raises
the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Entrenamiento_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Entrenamiento_OpeningFcn via varargin.
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help Entrenamiento
% Last Modified by GUIDE v2.5 18-Apr-2011 23:42:48
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Entrenamiento_OpeningFcn, ...
    'gui_OutputFcn', @Entrenamiento_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```

else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% — Executes just before Entrenamiento is made visible.
function Entrenamiento_OpeningFcn(hObject, ~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Entrenamiento (see VARARGIN)
% Choose default command line output for Entrenamiento
handles.output = hObject;
    t=(0.5:0.5:24)';
    handles.t=t;
    caso=1;
    handles.caso=caso;
    entrenada=0;
    handles.caso1.entrenada=entrenada;
    handles.caso2.entrenada=entrenada;
    anox=0;
    handles.anox=anox;
    ano=0;
    handles.ano=ano;
    feriado="";
    handles.feriado=feriado;
    dia="";
    handles.dia=dia;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes Entrenamiento wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% — entrenada from this function are returned to the command line.
function varargout = Entrenamiento_OutputFcn(~, ~, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject    handle to figure

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% — Executes on button press in button_feriados.
function button_feriados_Callback(hObject, ~, handles)
% hObject handle to button_feriados (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
    utpl='Cargando Tabla de Feriados...';
    set(handles.text,'String',utpl);
% cargando tabla de feriados
    feriado=importferiado;
    handles.feriado=feriado;
    if isequal(feriado,"")
        utpl='Vuelva a Cargar Tabla de Feriados:>';
        set(handles.text,'String',utpl);
    else
        utpl='Cargar Datos de Carga para Entrenamiento:>';
        set(handles.text,'String',utpl);
    end

guidata(hObject, handles);
% — Executes on button press in button_cargar.
function button_cargar_Callback(hObject, ~, handles)
% hObject handle to button_cargar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
t=handles.t;

    utpl='cargando datos...';
    set(handles.text,'String',utpl);

%Código para Cargar Datos
[ano , dia, title]=importfile;

```

```

if isequal(ano,0)
    utpl='Vuelva a Cargar Datos de Carga para Entrenamiento:>';
    set(handles.text,'String',utpl);
    anox=0;
    handles.anox=anox;

else
    anox=createFit(t,ano);
    createfigure(t, ano, title)
    createfigure(t, anox, [title,' corregido'])

    handles.dia=dia;
    handles.ano=ano;
    handles.anox=anox;

    utpl='Listo Para Entrenar';
    set(handles.text,'String',utpl);
end

guidata(hObject, handles);

% -- Executes on button press in button_entrenar.
function button_entrenar_Callback(hObject, ~, handles)
% hObject    handle to button_entrenar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
caso =handles.caso ;
feriado =handles.feriado;
dia =handles.dia;
anox =handles.anox;

if isequal(feriado,")
    errordlg('Vuelva a Cargar la Data de Feriados','Error');
    utpl='Cargue la Data de Feriados:>';
    set(handles.text,'String',utpl);
elseif isequal(anox,0)

```

```

errorDlg('Vuelva a Cargar la Data de Años','Error');
utpl='Cargue la Data de Años:';
set(handles.text,'String',utpl);
elseif caso==1
    utpl='Entrenando la Red por Método 1...';
    set(handles.text,'String',utpl);

    entrada=orden(anox,dia,feriado,1);
    [net,entrenada] = create_fit_net(entrada,anox);
    [err errore_diario errore]=create_hist(anox,entrenada);
    ploterrhist(err,'bins',30)

    handles.caso1.net=net;
    handles.caso1.entrenada=entrenada;
    handles.caso1.err=err;
    handles.caso1.errore=errore;
    handles.caso1.errore_diario=errore_diario;

    utpl='Ya Puede Calcular Resultados';
    set(handles.text,'String',utpl);
elseif caso==2
    utpl='Entrenando la Red por Método 2...';
    set(handles.text,'String',utpl);

    [grupo1 , ano1, dia1] = ordenG1(anox, dia, feriado,1);
    [grupo2 , ano2, dia2] = ordenG2(anox, dia, feriado,1);
    [grupo3 , ano3, dia3] = ordenG3(anox, dia, feriado,1);
    [grupo4 , ano4, dia4] = ordenG4(anox, dia, feriado,1);

    szG1=size(ano1); nG1=szG1(2);
    szG2=size(ano2); nG2=szG2(2);
    szG3=size(ano3); nG3=szG3(2);
    szG4=size(ano4); nG4=szG4(2);

    [netG1,entrenadaG1] = create_fit_net(grupo1,ano1);
    [netG2,entrenadaG2] = create_fit_net(grupo2,ano2);

```

```

[netG3,entrenadaG3] = create_fit_net(grupo3,ano3);
[netG4,entrenadaG4] = create_fit_net(grupo4,ano4);

entrenada=anox;
entrenadaG=[entrenadaG1, entrenadaG2, entrenadaG3, entrenadaG4];
diaent=[dia1;dia2;dia3;dia4];
primdia=datenum(dia(1),'dd/mm/yyyy');
numdiaent=datenum(diaent,'dd/mm/yyyy');
index=numdiaent-primdia+1;
sz=size(index);
ind=sz(1);
for i=1:ind
    entrenada(:,index(i))=entrenadaG(:,i);
end

[err errore_diario errore]=create_hist(anox,entrenada);

handles.caso2.entrenada=entrenada;
handles.caso2.err=err;
handles.caso2.errore=errore;
handles.caso2.errore_diario=errore_diario;

handles.caso2.net.netG1=netG1;
handles.caso2.net.netG2=netG2;
handles.caso2.net.netG3=netG3;
handles.caso2.net.netG4=netG4;

handles.caso2.net.nG1=nG1;
handles.caso2.net.nG2=nG2;
handles.caso2.net.nG3=nG3;
handles.caso2.net.nG4=nG4;

utpl='Ya Puede Calcular Resultados';
set(handles.text,'String',utpl);
end

```

```

guidata(hObject, handles);
% --- Executes on selection change in menu.
function menu_Callback(hObject, ~, handles)
% hObject    handle to menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns menu contents as cell array
%        contents{get(hObject,'Value')} returns selected item from menu
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function menu_CreateFcn(hObject, ~, handles)
% hObject    handle to menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
guidata(hObject, handles);

% --- Executes on button press in button_graficar.
function button_graficar_Callback(hObject, ~, handles)
% hObject    handle to button_graficar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
t=handles.t;
caso=handles.caso;
entrenada1=handles.caso1.entrenada;
entrenada2=handles.caso2.entrenada;

popup_sel_index = get(handles.menu, 'Value');

```

```

if (caso==1 && isequal(entrenada1,0)) || (caso==2 && isequal(entrenada2,0))
    msgbox('Data no entrenada','Precaución');
elseif caso==1 || caso==2

% Para el Caso1
if caso==1
    err=handles.caso1.err;
    errore_diario=handles.caso1.errore_diario;
    entrenada=entrenada1;
% Para el Caso2
elseif caso==2
    err=handles.caso2.err;
    errore_diario=handles.caso2.errore_diario;
    entrenada=entrenada2;
end

switch popup_sel_index
case 1 %histograma de errores horarios
    figure
    hist(err);
    title({'Histograma de Errores por Hora';...
        'de Datos Utilizados para Entrenamiento'},'FontWeight','demi',...
        'FontSize',11,'FontName','Times New Roman');
    xlabel('error (%)','FontWeight','light','FontSize',11,...
        'FontName','Times New Roman');
    ylabel('veces que se repite','FontWeight','light','FontSize',12,...
        'FontName','Times New Roman');
case 2 %histograma de errores diarios
    figure
    hist(errore_diario);
    title({'Histograma de Errores por Día';...
        'de Datos Utilizados para Entrenamiento'},'FontWeight','demi',...
        'FontSize',11,'FontName','Times New Roman');
    xlabel('error (%)','FontWeight','light','FontSize',11,...
        'FontName','Times New Roman');

```

```

        ylabel('veces que se repite','FontWeight','light','FontSize',12,...
            'FontName','Times New Roman');
case 3 %gráfico de entrenamiento
    createfigure(t, entrenada, 'Gráficas de Predicción de Datos Entrenados')
case 4 %gráfico de error horario
    sz=size(err);
    h=sz(1);
    hr=0.5:0.5:h/2;
    figure
    plot(hr,abs(err));
    axis tight
    title('(%) Error Horario','FontWeight','demi',...
        'FontSize',11,'FontName','Times New Roman');
    xlabel('Horas (h)','FontWeight','light','FontSize',11,...
        'FontName','Times New Roman');
    ylabel('% Error','FontWeight','light','FontSize',12,...
        'FontName','Times New Roman');
case 5 % gráfico de error diario
    figure
    plot(abs(errore_diario));
    axis tight
    title('(%) Error Diario','FontWeight','demi',...
        'FontSize',11,'FontName','Times New Roman');
    xlabel('Días','FontWeight','light','FontSize',11,...
        'FontName','Times New Roman');
    ylabel('% Error','FontWeight','light','FontSize',12,...
        'FontName','Times New Roman');
end
end
guidata(hObject, handles);

% — Executes on button press in button_resultados.
function button_resultados_Callback(hObject, ~, handles)
% hObject    handle to button_resultados (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

caso=handles.caso;
dia=handles.dia;
entrenada1=handles.caso1.entrenada;
entrenada2=handles.caso2.entrenada;

if (caso==1 && isequal(entrenada1,0)) || (caso==2 && isequal(entrenada2,0))
    errorDlg('Falta Entrenar la Red','Error');
elseif caso==1
    err=handles.caso1.err;
    errore_diario=handles.caso1.errore_diario;
    errore=handles.caso1.errore;

    [tabla1 tabla2]=report(err, errore_diario, errore, dia);
    Reporte(tabla1,tabla2)
elseif caso==2
    err=handles.caso2.err;
    errore_diario=handles.caso2.errore_diario;
    errore=handles.caso2.errore;

    [tabla1 tabla2]=report(err, errore_diario, errore, dia);
    Reporte(tabla1,tabla2)
end

guidata(hObject, handles);

% --- Executes on button press in button_save.
function button_save_Callback(hObject, ~, handles)
% hObject    handle to button_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ano=handles.ano;
anox=handles.anox;
dia=handles.dia;
feriado=handles.feriado;

caso=handles.caso;

```

```

entrenada1=handles.caso1.entrenada;
entrenada2=handles.caso2.entrenada;

if (caso==1 && isequal(entrenada1,0)) || (caso==2 && isequal(entrenada2,0))
    errorDlg('Falta Entrenar la Red','Error');
elseif caso==1
    net=handles.caso1.net;
    entrenada=entrenada1;
    uisave({'net','ano','anox','dia','feriado','entrenada','caso'});
elseif caso==2
    net=handles.caso2.net;
    entrenada=entrenada2;
    uisave({'net','ano','anox','dia','feriado','entrenada','caso'});
end

```

```
guidata(hObject, handles);
```

```

% — Executes on button press in button_info1.
function button_info1_Callback(hObject, ~, handles)
% hObject    handle to button_info1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
texto=['Este tipo de entrenamiento considera el día anterior, ',...
    'el día de la semana pasada, la estación del año, el día ',...
    'de la semana y si es feriado o no'];
msgbox(texto,' Información ');
guidata(hObject, handles);

```

```

% — Executes on button press in button_info2.
function button_info2_Callback(hObject, ~, handles)
% hObject    handle to button_info2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
texto=['Este tipo de entrenamiento considera la data que utiliza',...
    ' el COES para su predicción de carga'];

```

```
msgbox(texto,' Información ');
guidata(hObject, handles);
```

% — **Executes when selected object is changed in panel.**

```
function panel_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in panel
% eventdata  structure with the following fields (see UIBUTTONGROUP)
% EventName: string 'SelectionChanged' (read only)
% OldValue: handle of the previously selected object or empty if none was selected
% NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)
```

```
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
```

```
    case 'metodo1'
```

```
        caso=1;
```

```
        handles.caso=caso;
```

```
        guidata(hObject,handles);
```

```
    case 'metodo2'
```

```
        caso=2;
```

```
        handles.caso=caso;
```

```
        guidata(hObject,handles);
```

```
    otherwise
```

```
        % Code for when there is no match.
```

```
End
```

C.- FUNCION: PRONOSTICO

```
function varargout = Pronostico(varargin)
```

```
% PRONOSTICO M-file for Pronostico.fig
```

```
% PRONOSTICO, by itself, creates a new PRONOSTICO or raises the existing  
% singleton*.
```

```
%
```

```
% H = PRONOSTICO returns the handle to a new PRONOSTICO or the handle to  
% the existing singleton*.
```

```
% PRONOSTICO('CALLBACK',hObject,eventData,handles,...) calls the local
```

```
% function named CALLBACK in PRONOSTICO.M with the given input arguments.
```

```

% PRONOSTICO('Property','Value',...) creates a new PRONOSTICO or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before Pronostico_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to Pronostico_OpeningFcn via varargin.
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text_Hora to modify the response to help Pronostico
% Last Modified by GUIDE v2.5 20-Apr-2011 05:01:01
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Pronostico_OpeningFcn, ...
    'gui_OutputFcn', @Pronostico_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% — Executes just before Pronostico is made visible.
function Pronostico_OpeningFcn(hObject,~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

`% varargin` command line arguments to Pronostico (see VARARGIN)

`% Choose default command line output for Pronostico`

`handles.output = hObject;`

`data=0;`

`handles.data=data;`

`net=0;`

`handles.net=net;`

`ano_entren_ruid=0;`

`handles.ano_entren_ruid=ano_entren_ruid;`

`ano_entren=0;`

`handles.ano_entren=ano_entren;`

`ano_pron_ruid=0;`

`handles.ano_pron_ruid=ano_pron_ruid;`

`ano_pron=0;`

`handles.ano_pron=ano_pron;`

`ano_tot_ruid=0;`

`handles.ano_tot_ruid=ano_tot_ruid;`

`ano_tot=0;`

`handles.ano_tot=ano_tot;`

`dia_entren="";`

`handles.dia_entren=dia_entren;`

`feriado_entren="";`

`handles.feriado_entren=feriado_entren;`

`dia_tot=0;`

```

handles.dia_tot=dia_tot;

dia_pron="";
handles.dia_pron=dia_pron;

feriado_pron="";
handles.feriado_pron=feriado_pron;

t=(0.5:0.5:24)';
handles.t=t;

entrenada=0;
handles.entrenada=entrenada;

pronosticada=0;
handles.pronosticada=pronosticada;

simu=0;
handles.simu=simu;

err=0;
handles.err=err;

errore_diario=0;
handles.errore_diario=errore_diario;

errore=0;
handles.errore=errore;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Pronostico wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% — Outputs from this function are returned to the command line.

```

```

function varargout = Pronostico_OutputFcn(hObject, ~, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
guidata(hObject, handles);

% — Executes on selection change in menu_dias.
function menu_dias_Callback(hObject, ~, handles)
% hObject handle to menu_dias (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
di0 = get(handles.menu_dias, 'Value');
me0 = get(handles.menu_meses, 'Value');
ano0 = get(handles.menu_anos, 'Value')-1+2007;

num=datenum(ano0,me0,di0);
tex=datestr(num,'dd/mm/yyyy');
set(handles.text_fecha,'String',tex);

% Hints: contents = cellstr(get(hObject,'String')) returns menu_dias contents as cell array
% contents{get(hObject,'Value')} returns selected item from menu_dias
guidata(hObject, handles);

% — Executes during object creation, after setting all properties.
function menu_dias_CreateFcn(hObject, ~, handles)
% hObject handle to menu_dias (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
guidata(hObject, handles);

% — Executes on selection change in menu_meses.
function menu_meses_Callback(hObject, ~, handles)
% hObject    handle to menu_meses (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
di0 = get(handles.menu_dias, 'Value');
me0 = get(handles.menu_meses, 'Value');
ano0 = get(handles.menu_anos, 'Value')-1+2007;

num=datenum(ano0,me0,di0);
tex=datestr(num,'dd/mm/yyyy');
set(handles.text_fecha,'String',tex);
% Hints: contents = cellstr(get(hObject,'String')) returns menu_meses contents as cell
array
%    contents{get(hObject,'Value')} returns selected item from menu_meses
guidata(hObject, handles);

% — Executes during object creation, after setting all properties.
function menu_meses_CreateFcn(hObject, ~, handles)
% hObject    handle to menu_meses (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
guidata(hObject, handles);

```

```

% — Executes on selection change in menu_anos.
function menu_anos_Callback(hObject, ~, handles)
% hObject    handle to menu_anos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
di0 = get(handles.menu_dias, 'Value');
me0 = get(handles.menu_meses, 'Value');
ano0 = get(handles.menu_anos, 'Value')-1+2007;

num=datenum(ano0,me0,di0);
tex=datestr(num,'dd/mm/yyyy');
set(handles.text_fecha,'String',tex);
% Hints: contents = cellstr(get(hObject,'String')) returns menu_anos contents as cell array
%    contents{get(hObject,'Value')} returns selected item from menu_anos
guidata(hObject, handles);

```

```

% — Executes during object creation, after setting all properties.
function menu_anos_CreateFcn(hObject, ~, handles)
% hObject    handle to menu_anos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
guidata(hObject, handles);

```

```

% — Executes on selection change in popupmenu.
function popupmenu_Callback(hObject, ~, handles)
% hObject    handle to popupmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu contents as cell array
%     contents{get(hObject,'Value')} returns selected item from popupmenu
guidata(hObject, handles);
```

```
% — Executes during object creation, after setting all properties.
```

```
function popupmenu_CreateFcn(hObject, ~, handles)
```

```
% hObject    handle to popupmenu (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
```

```
%     See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
guidata(hObject, handles);
```

```
% — Executes on button press in entrenar.
```

```
function entrenar_Callback(hObject, ~, handles)
```

```
% hObject    handle to entrenar (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
    utpl='Entrenando Nueva Red...';
```

```
    set(handles.text,'String',utpl);
```

```
    Entrenamiento;
```

```
    utpl='Cargar Red Entrenada';
```

```
    set(handles.text,'String',utpl);
```

```
guidata(hObject, handles);
```

```
% — Executes on button press in boton_net.
```

```

function button_net_Callback(hObject, ~, handles)
% hObject    handle to button_net (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

utpl='Cargando Red Neuronal...';
set(handles.text,'String',utpl);

fileToRead=uigetfile('*.mat','Seleccione Red Neuronal Entrenada');

if isequal(fileToRead,0)
    msgbox('Red Neuronal no Seleccionada',' Precaución ');
    utpl='vuelva a cargar la red neuronal';
    set(handles.text,'String',utpl);
else
    data=load(fileToRead);

    logic=isfield(data,{'net','ano','anox','dia','feriado','entrenada','caso'});

    if all(logic)
        caso=data.caso;
        handles.caso=caso;

        net=data.net;
        handles.net=net;

        ano_entren_ruid=data.ano;
        handles.ano_entren_ruid=ano_entren_ruid;

        ano_entren=data.anox;
        handles.ano_entren=ano_entren;

        dia_entren=data.dia;
        handles.dia_entren=dia_entren;

        feriado_entren=data.feriado;

```

```

handles.feriado_entren=feriado_entren;

entrenada=data.entrenada;
handles.entrenada=entrenada;

utpl='Cargue Tabla de Feriados';
set(handles.text,'String',utpl);

else
    errordlg('Seleccione otra Red Neuronal', ' Error');

    utpl='Cargue otra Red Neuronal';
    set(handles.text,'String',utpl);
end
end
guidata(hObject, handles);

% -- Executes on button press in button_feriados.
function button_feriados_Callback(hObject, ~, handles)
% hObject    handle to button_feriados (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% cargando nueva tabla de feriados
utpl='Cargando Tabla de Feriados...';
set(handles.text,'String',utpl);

feriado_pron=importferiado;
handles.feriado_pron=feriado_pron;

if isequal(feriado_pron,"")
    utpl='Vuelva a Cargar Tabla de Feriados:.';
    set(handles.text,'String',utpl);
else
    utpl='Cargue Data para Pronóstico:.';
    set(handles.text,'String',utpl);

```

end

guidata(hObject, handles);

% --- Executes on button press in button_data.

function button_data_Callback(hObject, ~, handles)

% hObject handle to button_data (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

t=handles.t;

utpl='Cargando Datos...';

set(handles.text,'String',utpl);

[ano_pron_ruid , dia_pron, title_pron]=importfile;

if isequal(ano_pron_ruid,0)

utpl='Vuelva a Cargar Datos de Carga para Entrenamiento:';

set(handles.text,'String',utpl);

else

ano_pron=createFit(t,ano_pron_ruid);

createfigure(t, ano_pron_ruid, title_pron)

createfigure(t, ano_pron, [title_pron,' corregido'])

handles.dia_pron=dia_pron;

handles.ano_pron_ruid=ano_pron_ruid;

handles.ano_pron=ano_pron;

utpl='Listo Para Predicción';

set(handles.text,'String',utpl);

end

guidata(hObject, handles);

% --- Executes on button press in button_prediccion.

function button_prediccion_Callback(hObject, ~, handles)

% hObject handle to button_prediccion (see GCBO)

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

net=handles.net;
feriado_pron=handles.feriado_pron;
ano_entren=handles.ano_entren;
ano_pron=handles.ano_pron;
dia_entren=handles.dia_entren;
dia_pron=handles.dia_pron;

if isequal(net,0)
    error('Falta Procesar Información','Error');
    utpl='Falta Cargar Net';
    set(handles.text,'String',utpl);
elseif isequal(feriado_pron,"")
    error('Falta Procesar Información','Error');
    utpl='Falta Cargar Feriados';
    set(handles.text,'String',utpl);
elseif isequal(ano_pron,0) || isequal(dia_pron,"")
    error('Falta Procesar Información','Error');
    utpl='Falta Cargar Data';
    set(handles.text,'String',utpl);
else
    utpl='Pronosticando...';
    set(handles.text,'String',utpl);
    caso=handles.caso;
    sz=size(ano_entren);
    n=sz(2);

    %direccionando data
    ano_entren_ruid=handles.ano_entren_ruid;
    ano_pron_ruid=handles.ano_pron_ruid;

    ano_tot=[ano_entren,ano_pron];
    handles.ano_tot=ano_tot;
    dia_tot=[dia_entren;dia_pron];

```

```

handles.dia_tot=dia_tot;
ano_tot_ruid=[ano_entren_ruid,ano_pron_ruid];
handles.ano_tot_ruid=ano_tot_ruid;

%utilizando la red neuronal para la predicción
if caso==1
    entrada_tot=orden(ano_tot,dia_tot,feriado_pron,1);
    entrada_pron=entrada_tot(:,n+1:end);
    pronosticada=sim(net,entrada_pron);
    handles.pronosticada=pronosticada;

elseif caso==2
    nG1=net.nG1; netG1=net.netG1;
    nG2=net.nG2; netG2=net.netG2;
    nG3=net.nG3; netG3=net.netG3;
    nG4=net.nG4; netG4=net.netG4;

    [grupo1 , ~, dia1] = ordenG1(ano_tot,dia_tot,feriado_pron,1);
    [grupo2 , ~, dia2] = ordenG2(ano_tot,dia_tot,feriado_pron,1);
    [grupo3 , ~, dia3] = ordenG3(ano_tot,dia_tot,feriado_pron,1);
    [grupo4 , ~, dia4] = ordenG4(ano_tot,dia_tot,feriado_pron,1);

    grupo1_pron=grupo1(:,nG1+1:end);
    grupo2_pron=grupo2(:,nG2+1:end);
    grupo3_pron=grupo3(:,nG3+1:end);
    grupo4_pron=grupo4(:,nG4+1:end);

    pronG1=sim(netG1,grupo1_pron);
    pronG2=sim(netG2,grupo2_pron);
    pronG3=sim(netG3,grupo3_pron);
    pronG4=sim(netG4,grupo4_pron);

    pronosticada=ano_pron;
    pronG=[pronG1,pronG2,pronG3,pronG4];
    diapron=[dia1(nG1+1:end);dia2(nG2+1:end);dia3(nG3+1:end);dia4(nG4+1:end)];
    primdia=datetime(dia_pron(1),'dd/mm/yyyy');

```

```

numdiapron=datenum(diapron,'dd/mm/yyyy');
index=numdiapron-primdia+1;
sz=size(index);
ind=sz(1);
for i=1:ind
    pronosticada(:,index(i))=pronG(:,i);
end
handles.pronosticada=pronosticada;
end

%direccionando nueva data
entrenada=handles.entrenada;
simu=[entrenada, pronosticada];
handles.simu=simu;

%calculando los errores de la predicci3n
[err errore_diario
errore]=create_hist_fer(ano_pron,pronosticada,dia_pron,feriado_pron);
ploterrhist(err,'bins',30)
handles.err=err;
handles.errore_diario=errore_diario;
handles.errore=errore;

utpl='Puede Calcular Resultados';
set(handles.text,'String',utpl);

txt=dia_pron(1);
set(handles.text_fecha,'String',txt);

[an0 me0 di0]=datevec(txt,'dd/mm/yyyy');

set(handles.menu_dias,'Value',di0);
set(handles.menu_meses,'Value',me0);
set(handles.menu_anos,'Value',an0-2007+1);

```

```

end
guidata(hObject, handles);

% — Executes on button press in button_resultados.
function button_resultados_Callback(hObject, ~, handles)
% hObject    handle to button_resultados (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

ano_pron=handles.ano_pron;
dia_pron=handles.dia_pron;
pronosticada=handles.pronosticada;

err=handles.err;
errore_diario=handles.errore_diario;
errore=handles.errore;

if isequal(ano_pron,0) || isequal(pronosticada,0) || isequal(dia_pron,'')
    errordlg('Falta Procesar Información','Error');
    utpl='Falta a Cargar Data';
    set(handles.text,'String',utpl);
elseif isequal(errore,0)
    errordlg('Falta Procesar Información','Error');
    utpl='Falta Pronosticar Data';
    set(handles.text,'String',utpl);
else
    [tabla1 tabla2]=report(err, errore_diario, errore, dia_pron);
    Reporte(tabla1,tabla2)
end

guidata(hObject, handles);

% — Executes on button press in button_graficarmenu.
function button_graficarmenu_Callback(hObject, ~, handles)
% hObject    handle to button_graficarmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)
err=handles.err;
errore_diario=handles.errore_diario;
pronosticada=handles.pronosticada;
t=handles.t;

popup_sel_index = get(handles.popupmenu, 'Value');

switch popup_sel_index
    case 1 %histograma de errores horarios
        if isequal(err,0)
            msgbox('Data no entrenada','Precaución');
        else
            figure
            hist(err);
            title({'Histograma de Errores por Hora';...
                'de Datos Utilizados para Entrenamiento'},'FontWeight','demi',...
                'FontSize',11,'FontName','Times New Roman');
            xlabel('error (%)','FontWeight','light','FontSize',11,...
                'FontName','Times New Roman');
            ylabel('veces que se repite','FontWeight','light','FontSize',12,...
                'FontName','Times New Roman');
        end
    case 2 %histograma de errores diarios
        if isequal(errore_diario,0)
            msgbox('Data no entrenada','Precaución');
        else
            figure
            hist(errore_diario);
            title({'Histograma de Errores por Día';...
                'de Datos Utilizados para Entrenamiento'},'FontWeight','demi',...
                'FontSize',11,'FontName','Times New Roman');
            xlabel('error (%)','FontWeight','light','FontSize',11,...
                'FontName','Times New Roman');
            ylabel('veces que se repite','FontWeight','light','FontSize',12,...
                'FontName','Times New Roman');
        end
end

```

```

end
case 3 %gráfico de entrenamiento
if isequal(pronosticada,0)
    msgbox('Data no entrenada','Precaución');
else
    createfigure(t, pronosticada, 'Gráficas de Predicción de Datos Entrenados')
end
case 4 %gráfico de error horario
if isequal(err,0)
    msgbox('Data no entrenada','Precaución');
else
    sz=size(err);
    h=sz(1);
    hr=0.5:0.5:h/2;
    figure
    plot(hr,abs(err));
    axis tight
    title('% Error Horario','FontWeight','demi',...
        'FontSize',11,'FontName','Times New Roman');
    xlabel('Horas (h)','FontWeight','light','FontSize',11,...
        'FontName','Times New Roman');
    ylabel('% Error','FontWeight','light','FontSize',12,...
        'FontName','Times New Roman');
end
case 5 % gráfico de error diario
if isequal(errore_diario,0)
    msgbox('Data no entrenada','Precaución');
else
    figure
    plot(abs(errore_diario));
    axis tight
    title('% Error Diario','FontWeight','demi',...
        'FontSize',11,'FontName','Times New Roman');
    xlabel('Días','FontWeight','light','FontSize',11,...
        'FontName','Times New Roman');
    ylabel('% Error','FontWeight','light','FontSize',12,...

```

```

        'FontName','Times New Roman');
    end
end
guidata(hObject, handles);

% --- Executes on button press in button_graficar.
function button_graficar_Callback(hObject, eventdata, handles)
% hObject    handle to button_graficar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
t=handles.t;
ano_tot_ruid=handles.ano_tot_ruid;
ano_tot=handles.ano_tot;
simu=handles.simu;
dia_tot=handles.dia_tot;

if isequal(ano_tot_ruid,0) || isequal(ano_tot,0) || isequal(simu,0) || isequal(dia_tot,"")
    %errorldg('Falta Procesar Información','Error');
else
    index_ano = get(handles.menu_anos, 'Value')-1+2007;
    index_mes = get(handles.menu_meses, 'Value');
    index_dia = get(handles.menu_dias, 'Value');

    fecha=[index_ano, index_mes, index_dia];
    index=datenum(fecha)-datenum(dia_tot(1),'dd/mm/yyyy')+1

    numdia=datenum(fecha);
    fecha=datestr(numdia,24);
    numsem=weekday(numdia);
    semana={'Domingo','Lunes','Martes','Miércoles','Jueves','Viernes','Sábado'};
    diasem=char(semana(numsem));
    titulo=[diasem,' ',fecha];

    sz=size(dia_tot);
    if index<=sz(1) && index>0
        dato_real=ano_tot_ruid(:,index);
    end
end

```

```

dato_limpio=ano_tot(:,index);
dato_pron=simu(:,index);
createfigure_comp(t,[dato_real,dato_limpio,dato_pron],titulo)

%txt=dia_tot(index);
%set(handles.text_fecha,'String',txt);
button_calcular_Callback(hObject, eventdata, handles)
else
    errorDlg('Falta Completar Data','Error');
end
end
guidata(hObject, handles);

% — Executes on button press in button_histograma.
function button_histograma_Callback(hObject, eventdata, handles)
% hObject    handle to button_histograma (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
t=handles.t;
ano_tot=handles.ano_tot;
simu=handles.simu;
dia_tot=handles.dia_tot;

if isequal(ano_tot,0) || isequal(simu,0) || isequal(dia_tot,"")
    %errorDlg('Falta Procesar Información','Error');
else
    index_ano = get(handles.menu_anos, 'Value')-1+2007;
    index_mes = get(handles.menu_meses, 'Value');
    index_dia = get(handles.menu_dias, 'Value');

    fecha=[index_ano, index_mes, index_dia];
    index=datenum(fecha)-datenum(dia_tot(1),'dd/mm/yyyy')+1

    sz=size(dia_tot);
    if index<=sz(1) && index>0
        dato_limpio=ano_tot(:,index);

```

```

dato_pron=simu(:,index);

errordia=100*(dato_limpio-dato_pron)./dato_limpio;
create_error(t, errordia)

%txt=dia_tot(index);
%set(handles.text_fecha,'String',txt);
button_calcular_Callback(hObject, eventdata, handles)
else
    errordlg('Falta Completar Data','Error');
end
end
guidata(hObject, handles);

% -- Executes on button press in button_atraz.
function button_atraz_Callback(hObject, eventdata, handles)
% hObject    handle to button_atraz (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

dia_tot=handles.dia_tot;
sz=size(dia_tot);
n=sz(1);

if isequal(dia_tot,0)
else
    di0 = get(handles.menu_dias, 'Value');
    me0 = get(handles.menu_meses, 'Value');
    ano0 = get(handles.menu_anos, 'Value')-1+2007;

    num=datenum(ano0,me0,di0);

    index=datenum(num-1)-datenum(dia_tot(1),'dd/mm/yyyy')+1

    if index>0 && index<=n
        [index_ano index_mes index_dia]=datevec(num-1);

```

```

set(handles.menu_dias,'Value',index_dia);
menu_dias_Callback(hObject, eventdata, handles)
set(handles.menu_meses,'Value',index_mes);
menu_meses_Callback(hObject, eventdata, handles)
set(handles.menu_anos,'Value',index_ano-2007+1);
menu_anos_Callback(hObject, eventdata, handles)

%button_graficar_Callback(hObject, eventdata, handles)
%button_histograma_Callback(hObject, eventdata, handles)
button_calcular_Callback(hObject, eventdata, handles)
end
end
guidata(hObject, handles);

% -- Executes on button press in button_adelante.
function button_adelante_Callback(hObject, eventdata, handles)
% hObject    handle to button_adelante (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

dia_tot=handles.dia_tot;
sz=size(dia_tot);
n=sz(1);

if isequal(dia_tot,0)
else
    di0 = get(handles.menu_dias, 'Value');
    me0 = get(handles.menu_meses, 'Value');
    ano0 = get(handles.menu_anos, 'Value')-1+2007;

    num=datenum(ano0,me0,di0);

    index=datenum(num+1)-datenum(dia_tot(1),'dd/mm/yyyy')+1

    if index>0 && index<=n

```

```

[index_ano index_mes index_dia]=datevec(num+1);

set(handles.menu_dias,'Value',index_dia);
menu_dias_Callback(hObject, eventdata, handles)
set(handles.menu_meses,'Value',index_mes);
menu_meses_Callback(hObject, eventdata, handles)
set(handles.menu_anos,'Value',index_ano-2007+1);
menu_anos_Callback(hObject, eventdata, handles)

%button_graficar_Callback(hObject, eventdata, handles)
%button_histograma_Callback(hObject, eventdata, handles)
button_calcular_Callback(hObject, eventdata, handles)
end
end
guidata(hObject, handles);

% — Executes on button press in button_calcular.
function button_calcular_Callback(hObject, ~, handles)
% hObject    handle to button_calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

ano_tot=handles.ano_tot;
simu=handles.simu;
dia_tot=handles.dia_tot;
sz=size(dia_tot);
n=sz(1);

if isequal(ano_tot,0) || isequal(simu,0) || isequal(dia_tot,")
else
    di0 = get(handles.menu_dias, 'Value');
    me0 = get(handles.menu_meses, 'Value');
    ano0 = get(handles.menu_anos, 'Value')-1+2007;

    num=datenum(ano0,me0,di0);

```

```

index=datenum(num)-datenum(dia_tot(1),'dd/mm/yyyy')+1;

if index>0 && index<=n
    dato_limpio=ano_tot(:,index);
    dato_pron=simu(:,index);
    errordia=100*(dato_limpio-dato_pron)./dato_limpio;

    data=[dato_limpio,dato_pron,errordia];

    set(handles.uitable,'Data',data);
else
    errordlg('Falta Completar Data','Error');
end
end
guidata(hObject, handles);

% — Executes on button press in button_exel.
function button_exel_Callback(hObject, ~, handles)
% hObject    handle to button_exel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ano_tot=handles.ano_tot;
simu=handles.simu;
dia_tot=handles.dia_tot;

if isequal(dia_tot,0) || isequal(simu,0)
    errordlg('Falta Procesar Información','Error');
else
    title=['día','00:30','01:00','01:30','02:00','02:30','03:00','03:30',...
          '04:00','04:30','05:00','05:30','06:00','06:30','07:00',...
          '07:30','08:00','08:30','09:00','09:30','10:00','10:30',...
          '11:00','11:30','12:00','12:30','13:00','13:30','14:00',...
          '14:30','15:00','15:30','16:00','16:30','17:00','17:30',...
          '18:00','18:30','19:00','19:30','20:00','20:30','21:00',...
          '21:30','22:00','22:30','23:00','23:30','23:59'];
    %data={dia_tot,num2cell(ano_tot'),num2cell(simu')};

```

```
data=[dia_tot,num2cell(simu')];

exel=[title;data];

[file ~] = uiputfile('*.xls','Guardar Data Como');
x=file;
if x==0
    msgbox('Data no Guardada','Observación');
else
    [stat ~]=xlswrite(file, exel);
    y=stat;
    if y==0
        errordlg('Primero cierre el exel y vuelva a cargar','Error');
    end
end
end
guidata(hObject, handles);
```