



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA MECÁNICA
ELÉCTRICA



**“APLICACIÓN DE LAS REDES NEURONALES RECURRENTE
PARA LA PREDICCIÓN DE LA DEMANDA DE ENERGÍA
ELÉCTRICA EN LA BARRA DE 10KV – JULIACA, DE LA
EMPRESA DE DISTRIBUCIÓN ELECTRO PUNO S.A.A”**

TESIS

PRESENTADA POR:

JHONATAN EDWIN LARICO CAPIA

**PARA OBTAR EL TÍTULO PROFESIONAL DE:
INGENIERO MECÁNICO ELECTRISISTA**

PUNO – PERÚ

2022



Reporte de similitud

NOMBRE DEL TRABAJO

APLICACIÓN DE LAS REDES NEURONALES RECURRENTES PARA LA PREDICCIÓN DE LA DEMANDA DE ENERGÍA ELÉCTRICA EN LA BARRA DE 10 KV - JULIACA, DE LA EMPRESA DE DISTRIBUCIÓN ELECTRO PUNO S.A.A.

AUTOR

JHONTAN EDWIN LARICO CAPIA

RECuento DE PALABRAS

12488 Words

RECuento DE CARACTERES

70013 Characters

RECuento DE PÁGINAS

82 Pages

TAMAÑO DEL ARCHIVO

2.3MB

FECHA DE ENTREGA

Nov 16, 2022 10:11 AM GMT-5

FECHA DEL INFORME

Nov 16, 2022 10:16 AM GMT-5

● 16% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos

- 14% Base de datos de Internet
- Base de datos de Crossref
- 11% Base de datos de trabajos entregados
- 5% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 8 palabras)



Firmado digitalmente por RAMOS
CUTIPA Jose Manuel FAU
20145495170 soft
Motivo: Soy el autor del documento
Fecha: 16.11.2022 15:02:16 -05:00

Resumen



DEDICATORIA

Dedico este trabajo a mis padres y a mi hermana por haberme dado su apoyo incondicional durante mis estudios, y por creer siempre en mis capacidades.

Jhonatan Edwin Larico Capia



AGRADECIMIENTOS

El más sincero agradecimiento a la Escuela Profesional de Ingeniería Mecánica Eléctrica por darme la oportunidad de formar parte de esta prestigiosa institución, como también a todos los docentes que me formaron durante los cinco años que curse mis estudios y a todos mis compañeros con los que pasamos grandes momentos.

Jhonatan Edwin Larico Capia



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ACRONIMOS

RESUMEN 12

ABSTRACT..... 13

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA..... 14

1.1.1. Diagnostico..... 14

1.1.2. Pronostico 15

1.1.3. Control del Pronostico..... 15

1.1.4. Formulación de problema..... 15

1.2. OBJETIVOS 15

1.2.1. Objetivo general 15

1.2.2. Objetivos específicos 16

1.3. HIPÓTESIS..... 16

1.3.1. Hipótesis general..... 16

1.3.2. Hipótesis específicas..... 16

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES 17



| | |
|---|-----------|
| 2.1.1. Antecedentes internacionales..... | 17 |
| 2.1.2. Antecedentes Nacionales..... | 19 |
| 2.2. HORIZONTES DE PRONÓSTICO DE LA DEMANDA DE ENERGÍA ELÉCTRICA | 21 |
| 2.2.1. Pronóstico a largo plazo..... | 21 |
| 2.2.2. Pronóstico a mediano plazo..... | 21 |
| 2.2.3. Pronóstico a corto plazo..... | 21 |
| 2.3. FACTORES QUE INFLUYEN EN LA DEMANDA ELÉCTRICA..... | 22 |
| 2.3.1. Factores económicos..... | 22 |
| 2.3.2. Factores temporales..... | 23 |
| 2.3.3. Factores climáticos..... | 23 |
| 2.4. MÉTODOS DE PREDICCIÓN | 23 |
| 2.4.1. Regresión lineal múltiple..... | 24 |
| 2.4.2. Alisamiento exponencial | 24 |
| 2.4.3. Series temporales estocásticas | 25 |
| 2.4.4. Modelo econométrico..... | 26 |
| 2.5. REDES NEURONALES..... | 26 |
| 2.5.1. Neurona biología y artificial..... | 26 |
| 2.5.2. Topología de una red neuronal | 28 |
| 2.5.3. Elementos de una red neuronal..... | 29 |
| 2.5.4. Aprendizaje y Entrenamiento..... | 31 |
| 2.6. REDES NEURONALES RECURRENTES..... | 34 |
| 2.6.1. Red neuronal recurrente LSTM (Long short-term memory) | 36 |
| 2.7. DEFINICIÓN DE TÉRMINOS | 38 |



CAPITULO III

MATERIALES Y MÉTODOS

| | |
|---|-----------|
| 3.1. MATERIALES..... | 40 |
| 3.2. CLASIFICACIÓN DE LA INVESTIGACIÓN..... | 40 |
| 3.3. VARIABLES DE LA INVESTIGACIÓN..... | 40 |
| 3.3.1. Definición de las variables..... | 40 |
| 3.3.2. Operacionalización de variables..... | 41 |
| 3.4. POBLACIÓN Y MUESTRA..... | 41 |
| 3.4.1. Población..... | 41 |
| 3.4.2. Muestra..... | 41 |
| 3.5. TÉCNICAS DE RECOLECCIÓN DE DATOS..... | 41 |
| 3.5.1. Recolección de datos..... | 41 |
| 3.6. METODOLOGÍA PARA EL DISEÑO DE LA RED NEURONAL RECURRENTE..... | 42 |
| 3.6.1. Preprocesamiento de datos..... | 42 |
| 3.6.2. Desarrollo de las Redes..... | 50 |
| 3.6.3. Entrenamiento..... | 55 |
| 3.6.4. Comparación de modelos..... | 56 |

CAPITULO IV

RESULTADOS Y DISCUSIÓN

| | |
|---|-----------|
| 4.1. DATOS DE ENTRADA A LA RED NEURONAL..... | 62 |
| 4.2. MODELO RNR LSTM SIMPLE VS LSTM APILADO VS LSTM BIDIRECCIONAL..... | 63 |
| 4.3. LSTM BIDIRECCIONAL CON 96 VALORES DE ENTRADA..... | 66 |
| 4.4. MODELO RNA – LSTM CON DATOS SIN PROCESAR Y PROCESADO... 66 | |



| | |
|---|-----------|
| 4.5. DISCUSIÓN DE RESULTADOS | 68 |
| V. CONCLUSIONES..... | 69 |
| VI. RECOMENDACIONES..... | 70 |
| VII. REFERENCIAS BIBLIOGRÁFICAS..... | 71 |
| ANEXOS..... | 73 |
| ANEXO A. DIAGRAMA UNIFILAR DE LA SET TAPARACHI..... | 73 |
| ANEXO B. RESULTADOS DE LAS PREDICCIONES MEDIANTE RNR LSTM DESDE ENERO HASTA AGOSTO DEL 2021..... | 74 |
| ANEXO C: PROGRAMACIÓN DE LOS MODELOS RNN LSTM..... | 77 |
| ANEXO D: RESULTADO DEL MAPE DE TODOS LOS MODELOS | 78 |

Área: Control y automatización

Tema: Redes Neuronales Recurrentes

FECHA DE SUSTENTACIÓN: 25 de noviembre del 2022



ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Horizontes de pronostico | 22 |
| Figura 2: Modelo de series temporales..... | 25 |
| Figura 3: Representación de una neurona biológica..... | 27 |
| Figura 4: Estructura de una neurona artificial | 28 |
| Figura 5: Topología de una red neuronal | 28 |
| Figura 6: Elementos de una neurona artificial..... | 29 |
| Figura 7: Funciones de activación | 30 |
| Figura 8: Descenso del gradiente | 33 |
| Figura 9: Problemas de la taza de aprendizaje | 34 |
| Figura 10: Clasificación de las redes neuronales artificiales..... | 34 |
| Figura 11: Neurona recurrente simplificada a través del tiempo | 35 |
| Figura 12: Red neuronal recurrente a través del tiempo..... | 35 |
| Figura 13: Célula lstm | 37 |
| Figura 14: Metodología del trabajo de investigación..... | 42 |
| Figura 15: Serie de tiempo de la demanda de energía eléctrica | 43 |
| Figura 16: Perfil de carga diaria desde el 2018 hasta agosto del 2021..... | 44 |
| Figura 17: Valores no registrados..... | 45 |
| Figura 18: Diagrama de cajas | 46 |
| Figura 19: Datos atípicos de la serie de tiempo..... | 47 |
| Figura 20: Valor atípico, correspondiente al 26-04-2018 | 48 |
| Figura 21: Valor atípico, correspondiente al 19-07-2018 | 48 |
| Figura 22: Valor atípico, correspondiente al 25-01-2021 | 49 |
| Figura 23: Valor atípico, correspondiente al 12-08-2018 | 49 |
| Figura 24: Consecuencias del dropout | 52 |



| | |
|---|----|
| Figura 25: Arquitectura lstm simple..... | 53 |
| Figura 26: Arquitectura lstm apilado | 54 |
| Figura 27: Arquitectura lstm bidireccional..... | 54 |
| Figura 28: Entrenamiento de la rnn | 55 |
| Figura 29: Proceso de aprendizaje de la rnn, error vs épocas | 56 |
| Figura 30: Predicción de la demanda para el mes de enero del 2021 – lstm simple | 58 |
| Figura 31: Predicción de la demanda para el mes de enero del 2021 – lstm apilado | 58 |
| Figura 32: Predicción de la demanda para el mes de enero del 2021 – lstm bidireccional | 59 |
| Figura 33: Predicción de la demanda para el mes de enero del 2021 – lstm bidireccional con 96 valores de entrada..... | 60 |
| Figura 34: Predicción de la demanda de energía eléctrica para una semana..... | 63 |
| Figura 35: Predicción de la demanda de energía con lstm apilado | 64 |
| Figura 36: Predicción de la demanda de energía con lstm bidireccional | 64 |
| Figura 37: Predicción de la demanda de energía eléctrica para la primera semana de enero del 2021 con el modelo lstm bidireccional | 65 |
| Figura 38: Predicción de la demanda de energía eléctrica para los últimos 9 días del mes de agosto del 2021 con el modelo lstm bidireccional | 65 |
| Figura 39: Comparación de resultados con preprocesamiento y sin preprocesamiento de datos | 67 |



ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Definición de las variables..... | 40 |
| Tabla 2. Operacionalización de variables | 41 |
| Tabla 3. Cuartiles | 47 |
| Tabla 4. Resultados del mape | 57 |
| Tabla 5. Resultados mape lstm apilado..... | 59 |
| Tabla 6. Resultados mape lstm bidireccional | 60 |
| Tabla 7. Resultados mape lstm bidireccional | 61 |
| Tabla 8. Comparación de mape dependiendo los datos de entrada | 62 |
| Tabla 9. Comparación de mape entre modelos lstm | 64 |
| Tabla 10. Comparación de mape lstm bidireccional de 48 entradas vs 96 entradas..... | 66 |
| Tabla 11. Comparación del mape lstm bidireccional..... | 67 |



ÍNDICE DE ACRONIMOS

| ACRÓNIMO | SIGNIFICADO |
|-----------------|---------------------------------|
| RNR | Red neuronal recurrente |
| LSTM | Long Short-term memory |
| MAPE | Error percentual absolute medio |
| MSE | Error cuadrático medio |
| RNA | Red neuronal artificial |



RESUMEN

El presente trabajo de investigación tiene como propósito el de Implementar un modelo basado en redes neuronales recurrentes el cual pueda predecir la demanda de energía eléctrica con el mayor porcentaje de exactitud posible, utilizando los datos históricos suministrados por la Empresa de distribución Electro Puno, proponiendo un modelo basado en redes neuronales recurrentes para la predicción de la demanda de la energía eléctrica, se utiliza una base de datos de la demanda de energía eléctrica durante los años 2018, 2019, 2020 hasta agosto del 2021, el cual fue suministrado por la empresa Electro Puno S.A.A., Iniciándose con la inspección de los datos para lo cual se utiliza python y sus librerías, se realizan graficas del perfil de carga de la demanda para luego buscar patrones para construir el modelo y procediéndose a encontrar los datos atípicos y faltantes. Seguido se procede a la construcción de la red neuronal recurrente LSTM para lo cual se utiliza Tensorflow como herramienta principal y se evalúa las diferentes arquitecturas como la simple, apilada y bidireccional, obteniendo los mejores resultados con la arquitectura bidireccional y se concluye que el modelo es capaz de predecir la demanda de energía eléctrica en un horizonte de corto y mediano plazo, llegándose a determinar como resultado implemento un modelo de predicción basado en redes neuronales recurrentes LSTM que puede predecir la demanda de energía eléctrica con un 97.18% exactitud.

Palabras Clave: Modelo de predicción, Demanda eléctrica, preprocesamiento de datos, redes neuronales recurrentes LSTM, Python, Tensorflow



ABSTRACT

The present research work proposes a model based on recurrent neural networks for the prediction of the demand for electrical energy, a database of the demand for electrical energy is used during the years 2018, 2019, 2020 until August 2021, the which was supplied by the company Electro Puno SAA. The project begins with the inspection of the data for which python and its libraries are used, graphs of the demand load profile are made to later look for patterns that can help to build the model and then proceed to find the data atypical and missing. Once the data is processed, the LSTM recurrent neural network is built, for which TensorFlow is used as the main tool and the different architectures are evaluated, such as simple, stacked and bidirectional, obtaining the best results with the bidirectional architecture and it is concluded that the model is capable of predicting the demand for electrical energy in a short and medium term horizon.

Keywords: Prediction model, Electric demand, data preprocessing, LSTM recurrent neural networks, Python, TensorFlow



CAPÍTULO I

INTRODUCCIÓN

Debido al incremento de los sistemas eléctricos, esto debido al incremento de la población y también debido a la industria, es necesario usar nuevas tecnologías que están teniendo un gran auge estos días, como es la inteligencia artificial, ya que propone nuevos modelos para realizar predicciones esto es el caso de las redes neuronales recurrentes que utilizan la información recopilada durante varios años para establecer relaciones y patrones los cuales logra aprender y como resultado se obtiene un valor de una predicción.

1.1. PLANTEAMIENTO DEL PROBLEMA

Debido al continuo crecimiento urbano e industrial el pronóstico de la demanda de energía eléctrica debe realizarse con el menor porcentaje de error posible esto para evitar posibles pérdidas debido al sobre abastecimiento de energía o la falta de abastecimiento debido a una mala proyección, es por eso que este proyecto de investigación propone el uso de nuevas técnicas de predicción como son las que utiliza la inteligencia artificial.

1.1.1. Diagnostico

Una mala predicción representa para una concesionaria pérdidas económicas los mismos que vienen disminuyendo las utilidades de los resultados de Gestión de la Empresa ElectroPuno S.A.A., el mismo que se tendrá la realización y contratos sobre o sub dimensionados.



1.1.2. Pronostico

De tener una determinación de la demanda tanto de la Potencia como de energía imprecisas o con bajos niveles de exactitud, los resultados representarán según se presente los dos escenarios, si la determinación representa sobre dimensionamiento, se efectuará un contrato y desembolso económico en exceso, un segundo escenario es que la proyección esta sub dimensionada, la contratación y desembolso, generará pagos adicionales o mayores por excesos de consumo tanto de potencia y energía.

1.1.3. Control del Pronostico

La implementación de un modelo que propone el uso de nuevas técnicas de predicción como son las que utiliza la inteligencia artificial, reducirá los errores de sobre y/o sub dimensionamiento, que también reducirá las perdidas y/o pagos en exceso sobre los contratos tanto de potencia y energía ante las empresas generadoras.

Según lo planteado, el estudio deberá responder a las siguientes interrogantes.

1.1.4. Formulación de problema

¿Qué modelo de predicción basado en redes neuronales recurrentes minimizara el error de predicción de la demanda de la energía eléctrica?

1.2. OBJETIVOS

1.2.1. Objetivo general

Implementar un modelo basado en redes neuronales recurrentes el cual pueda predecir la demanda de energía eléctrica con el mayor porcentaje de exactitud posible,



utilizando los datos históricos suministrados por la Empresa de distribución Electro Puno

1.2.2. Objetivos específicos

- Desarrollar el preprocesamiento de datos del modelo de predicción basado en redes neuronales recurrentes.
- Implementar los diferentes tipos de arquitecturas LSTM en Python y evaluar los resultados para obtener el mejor modelo de predicción basado en redes neuronales recurrentes
- Evaluar los resultados obtenidos del modelo de predicción basado en redes neuronales recurrentes.

1.3. HIPÓTESIS

1.3.1. Hipótesis general

El modelo de predicción basado en redes neuronales recurrentes minimiza el error de predicción de la demanda de energía eléctrica.

1.3.2. Hipótesis específicas

- El preprocesamiento de datos minimiza el error del modelo de predicción basado en redes neuronales recurrentes.
- Con la arquitectura LSTM bidireccional se obtiene mejores resultados sobre el modelo de predicción basado en redes neuronales recurrentes.
- El error porcentual absoluto medio(MAPE) evalúa el error del modelo de predicción basado en redes neuronales recurrentes



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES

2.1.1. Antecedentes internacionales

Lozada, Acosta, Paredes, & Vique (2022), realizaron la investigación con el objetivo de pronosticar la demanda de electricidad utilizando un modelo de Red Neuronal Artificial (ANN). Para la elaboración de esta previsión se han utilizado los datos históricos de consumo facilitados por las empresas eléctricas (demanda máxima al mes) para el periodo 2019-2021. Este como resultado probó un modelo de red neuronal artificial que se desarrolló con una frecuencia creciente a lo largo del tiempo, se desarrollaron en varios pasos, que incluyen: entrenar la ANN usando el 85 % de los datos históricos, probar la ANN contra los criterios de evaluación del error porcentual absoluto medio (MAPE) en el 15 % de los valores que no son de entrenamiento. y la fase permite pronósticos con horizontes de tiempo más largos para demostrar la confiabilidad del modelo.

Lozano (2017) en su tesis “Aplicación de la red neuronal artificial Feedforward Back propagation para la predicción de la demanda de energía eléctrica en la empresa eléctrica Riobamba” Realiza un modelo basado en la red neuronal Artificial Feedforward Back propagation con el objetivo de reducir el error de predicción obtenido por la empresa de distribución Riobamba. El modelo cuenta con 70128 datos de los cuales 61344 los utiliza para entrenar el modelo y 8784 para realizar las pruebas de predicción, al realizar el preprocesamiento de datos obtiene 406 datos perdidos y 320 datos atípicos, al realizar las predicciones obtiene un error medio absoluto del 2.63% para el modelo Feedfoward Back propagation y realiza la comparación con el modelo estadístico como



es la regresión lineal obteniendo 4.56% de error y así concluye su trabajo de investigación confirmando el que modelo Feedforward Back propagation tiene menor error de predicción.

Morillo (2020), en su tesis “Modelo para el pronóstico de la demanda de energía eléctrica de EMELNORTE S.A – Ibarra aplicando redes neuronales artificiales” Realiza un modelo basado en RNA para la predicción de la demanda de energía utilizando los datos de la empresa EMELNORTE S.A, de los datos obtiene 48 mediciones por día durante cuatro años y los divide en datos de entrenamiento y prueba , como herramienta de programación utiliza el software MATLAB y la arquitectura de RNA que propone es de 2 neuronas de entrada, una capa oculta y 10 neuronas en la capa de salida, a su vez para el entrenamiento utiliza el método de regulación bayesiana.

Cabrera (2014), en su tesis “Diseño de una red neuronal artificial para la predicción de la demanda eléctrica” Realiza un modelo de predicción con la base de datos proporcionados por la empresa EERSSA, los datos son mediciones de la demanda máxima que registra el sistema SCADA a la salida de cada circuito en las subestaciones, consisten en observaciones de la demanda de consumo medida en intervalos de tiempo de 15 minutos, durante las 24 horas del día todos los días del año, trabaja con la información de los últimos 5 años. Como entradas hacia la red neuronal utiliza las mediciones registradas por el sistema a la salida del circuito Yaguarcuna en un intervalo de tiempo de un año, en la base de datos que equivale a 12 meses, y el consumo de cada mes seleccionado equivaldría a 24 datos. La cantidad de neuronas de entrada son 24 datos, una capa oculta con 24 neuronas y 1 neurona en la salida. Como resultados obtiene que el modelo puede pronosticar la demanda de energía eléctrica con una exactitud del 96%.



García (2009), en su tesis “Predicción de la demanda Eléctrica empleando redes neuronales artificiales” Realiza una red neuronal para la predicción de la demanda de energía aplicado a una barra de 33KV, utilizando datos de demanda de 24 horas del año 2008, los valores de entrada hacia la red neuronal son el mes, día, hora y demanda del día anterior al de la predicción de la energía, para determinar la cantidad de capas oculta realiza pruebas de ensayo y error obteniendo así 12 neuronas en la primera capa oculta y 8 en la segunda capa oculta, para la capa de salida utiliza una neurona. Para la etapa de entrenamiento utiliza 1991 datos, utiliza la técnica de aleatoriedad de datos para que la red no sufra de sobreajuste.

Ortiz (2011), en su tesis “Pronóstico de la demanda de energía eléctrica a corto plazo utilizando una red neuronal óptima” El proyecto realiza el pronóstico de la demanda a corto plazo utilizando RNA con el algoritmo de backpropagation cuya arquitectura se compone de 24 neuronas en la capa de entrada, 1 capa oculta con cuatro neuronas y 24 neuronas en la capa de salida, para obtener mejores resultados utiliza el preprocesamiento de datos para encontrar valores faltantes, atípicos y realizar el análisis estadístico de los datos lo cual es de vital importancia para clasificar los patrones de carga eléctrica a nivel diario. Para obtener una mayor exactitud del modelo de predicción utiliza el método de pronóstico del Ingenuo o caminata aleatoria lo cual permite reducir el error del pronóstico y como resultado se obtiene que la RNA con el algoritmo de backpropagation pronostica la demanda de energía eléctrica con una exactitud del 94%.

2.1.2. Antecedentes Nacionales

Al realizar la respectiva revisión de antecedentes en el ámbito peruano solo se encontraron dos trabajos referentes a la predicción de la demanda de energía eléctrica



usando redes neuronales, pero no se encontró ninguno acerca de redes neuronales recurrentes LSTM

Fernández (2014) en su tesis “Pronostico de la demanda eléctrica a corto plazo utilizando redes neuronales artificiales aplicado al sistema interconectado nacional”, Realiza la predicción de la demanda de energía eléctrica con los datos de COES, en donde nos indica que las predicciones de la demanda de energía eléctrica del Comité de Operación Económica del Sistema Interconectado Nacional difieren en gran medida con los valores reales de la demanda es por eso que con su proyecto plantea el uso de RNA para realizar el pronóstico de la demanda eléctrica a corto plazo con el objetivo de minimizar el error de predicción. La RNA tiene 98 valores de entrada, 1 capa oculta, para las neuronas en la capa oculta, realizó pruebas de ensayo y error a fin de encontrar el número adecuado de neuronas y para la capa de salida tiene 48 neuronas que representan los valores de la demanda de energía para un día, para realizar la programación utilizo MATLAB y como resultado final obtiene un MAPE del 2.99% y una desviación estándar de 3.29%.

Cuadrado y Ramos (2010), en su tesis “Implementación de una red neuronal para mejorar el pronóstico de la demanda de energía en la empresa BlueStar Energy Servecis Inc.” El proyecto plantea el uso de una RNA para predecir la demanda de energía en la empresa Blue Energy Services con el menor error posible a fin de reducir las pérdidas económicas debido al exceso de abastecimiento de energía. El proyecto empieza identificando los factores que puedan afectar el consumo de energía tales como los factores climáticos, atmosféricos, meteorológicos, comportamiento del consumo, el modelo que plantean es usar una RNA con 24 valores de entrada, 1 capa oculta con 4 neuronas y 24 valores de salida, el entrenamiento se realiza con el algoritmo de



backpropagation y como resultados obtienen que a través del uso de RNA se logra una mejor aproximación a la demanda real.

2.2. HORIZONTES DE PRONÓSTICO DE LA DEMANDA DE ENERGÍA ELÉCTRICA

Ariza (2013), nos indica que, el pronóstico de la demanda de energía eléctrica se realiza dependiendo al periodo y complejidad que se quiera proyecta y nos indica que los horizontes de pronóstico pueden ser corto, mediano y largo plazo.

2.2.1. Pronóstico a largo plazo

Se basa en realizar la proyección de la energía eléctrica en el transcurso de un año a más, a su vez realizar este tipo de pronóstico es de vital importancia ya que determina la planificación del sistema eléctrico permitiendo así determinar futuros proyectos de generación y transmisión de un sistema eléctrico de potencia y también las inversiones que se realizan en el sector energético de un determinado país. (Fernandez, 2014)

2.2.2. Pronóstico a mediano plazo

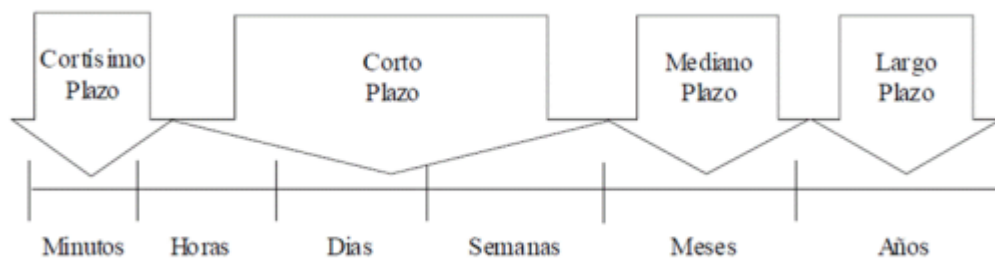
Se basa en realizar la proyección de la energía eléctrica en el transcurso de un mes hasta un año, los pronósticos de demanda de energía a mediano plazo pueden variar desde semanas a meses hasta cinco años. Es fundamental para la planificación de compras de combustible, ejecución de planes de mantenimiento de grupos electrógenos principales, coordinación y ejecución de contratos de compra/venta de energía. (Orellana, 2012)

2.2.3. Pronóstico a corto plazo

Se basa en realizar la proyección de la energía en el transcurso de un día hasta un mes, a su vez este tipo de proyección es útil para operadores de un sistema eléctrico de

potencia ya que deben realizar varias funciones como lo son el despacho económico, programación de transferencia de energía y control en tiempo real. Este tipo de proyección es de vital importancia para el operador de un sistema de potencia ya que realizar una proyección conservadora o elevada puede incurrir en pérdidas económicas (Fernandez, 2014)

Figura 1: *Horizontes de pronóstico*



Fuente: (Fernandez, 2014, pág. 6)

2.3. FACTORES QUE INFLUYEN EN LA DEMANDA ELÉCTRICA

Para realizar la predicción de la demanda de energía eléctrica se debe de tener en cuenta ciertos factores que pueden afectar sobre los resultados obtenidos, estos factores son las variables meteorológicas, socioeconómicas y demográficas es por eso que estas variables deben ser elegidas cuidadosamente y dependiendo el horizonte del pronóstico. (Fernandez, 2014)

2.3.1. Factores económicos

Los factores económicos influyen en el comportamiento de la demanda de energía debido a la actividad y crecimiento por parte de la industria, mientras que la industria crezca la actividad económica aumenta y a su vez se requieren nuevos proyectos



energéticos, por lo contrario, un decrecimiento en la económica significaría el decremento de la demanda de energía. (Ortiz, 2011)

2.3.2. Factores temporales

Según (Fernandez, 2014) los factores temporales son aquellos que ocurren ocasionalmente o temporariamente como las estaciones del año, días festivos, feriados, etc. Estos factores también afectan la demanda de energía eléctrica

2.3.3. Factores climáticos

Según (Fernandez, 2014) indica que las condiciones meteorológicas influyen sobre el comportamiento de la demanda eléctrica debido a que a bajas temperaturas se suele utilizar equipos de calefacción y por lo contrario debido a altas temperaturas se usa equipos de aire acondicionado, a su vez nos indica que en varios sistemas eléctricos de potencia la temperatura es tratada como una variable para realizar los respectivos pronósticos de la demanda de energía eléctrica,

2.4. MÉTODOS DE PREDICCIÓN

Realizar un adecuado modelo para la predicción de la energía eléctrica es de vital importancia ya que puede definir los futuros planes de inversión energética como la construcción de centrales eléctricas, líneas de transmisión y subestaciones eléctricas todo esto para poder asegurar un adecuado funcionamiento de los sistemas eléctricos de potencia y brindar un servicio adecuado y eficiente hacia los consumidores. Para realizar la predicción de la demanda de la energía eléctrica existen métodos estadísticos dentro de los cuales los más usados son la regresión lineal, regresión múltiple, alisamiento exponencial, series temporales, modelos econométricos. Por otra parte, también surgen



otros métodos de predicción los cuales son las redes neuronales artificiales los cuales forman parte de la inteligencia artificial.

2.4.1. Regresión lineal múltiple

La regresión lineal múltiple es una técnica estadística que se basa en predecir la variable dependiente, suministrándole una o más variables independientes, estas variables pueden ser la temperatura, registros históricos, etc. (Orellana, 2012)

La regresión lineal está definida por la siguiente ecuación:

$$Y(t) = a_0 + a_1x_1(t) + a_2x_2(t) + \dots \dots a_nx_n(t) + \epsilon(t)$$

En donde:

$$Y(t) = \text{Variable dependiente}$$

$$X_i(t) = \text{Variables independiente}$$

$$\epsilon(t) = \text{Variables aleatorias}$$

2.4.2. Alisamiento exponencial

El alisamiento exponencial utiliza funciones temporales $f(t)$ lo cual le permite realizar un modelo sobre la carga que es la variable dependiente en un determinado instante de tiempo. (Fernandez, 2014).

El alisamiento exponencial está definido por la siguiente ecuación:

$$Y(t) = \beta^t(t) * f(t) + \epsilon(t)$$

En donde:

$$\beta(t) \quad : \text{Vector de coeficientes} \qquad \epsilon(t) \quad : \text{Indicador aleatorio}$$

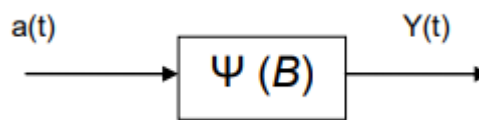
$$T \quad : \text{Transpuesta del vector}$$

2.4.3. Series temporales estocásticas

Las series temporales estocásticas son observaciones de series temporales, donde un conjunto de observaciones de tiempo se consideran una muestra de toda la serie temporal, este modelo es el más usado para realizar predicciones de series temporales, dentro de los modelos más usados se tiene el modelo autorregresivo integrado de medias móviles (ARIMA) y el modelo de media móvil integrada autorregresivo estacional (SARIMA). (Orellana, 2012)

La siguiente figura representa el modelo simple de una serie temporal la cual se representa como un filtro lineal.

Figura 2: Modelo de series temporales



Fuente: (Fernandez, 2014, pág. 23)

Donde:

$\Psi(B)$: Función de transferencia

B : Operador de retardo

La siguiente ecuación describe el modelo de una serie de tiempo

$$\emptyset(B) * \Phi(B^s) * v^{-d} * v_s^{-D} * Y(t) = \theta(B) * \Theta B^s * a(t)$$

Donde:

$\emptyset(B) * \Phi(B^s)$ Polinomios de grado p y q

$\theta(B) * \Theta(B^s)$ Polinomios de grado q y Q



2.4.4. Modelo econométrico

El modelo econométrico establece una relación entre diferentes variables con el fin de realizar predicciones, a estos modelos se los clasifica por regresión biecualcional e uniecualcional, además estos modelos consideran al producto bruto interno, población, tarifa eléctrica como variables explicativas y a su vez se ayuda de técnicas estadísticas como la prueba de Dickey-Fuller. (Orellana, 2012)

Uno de los métodos más usados por los modelos econométricos son los mínimos cuadrados el cual permite representar relaciones entre una variable económica endógena y una o varias variables exógenas. (Fernandez, 2014)

La siguiente ecuación representa los mínimos cuadrados

$$Y = a_1 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots \dots \beta_n X_n$$

2.5. REDES NEURONALES

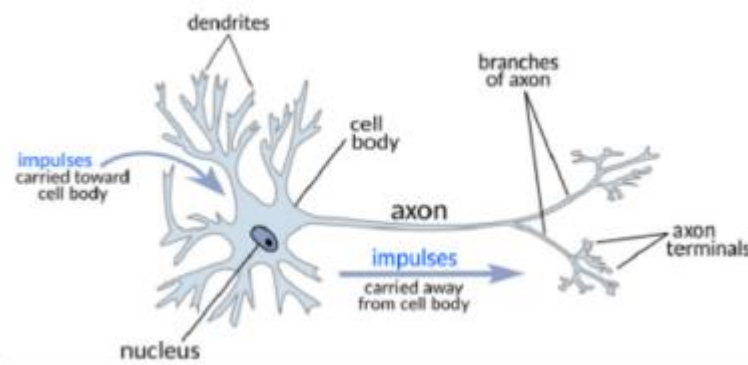
Las redes neuronales surgen con la intención de crear maquinas inteligentes que sean capaces de aprender patrones y relaciones entre variables de un conjunto de datos suministrado, las redes neuronales se inspiraron del funcionamiento del cerebro humano el cual establece conexiones entre las neuronas lo que le permite generar nuevo aprendizaje. Dentro de las aplicaciones más resaltantes de las redes neuronales se tiene los análisis de regresión, predicción de series temporales, clasificación de patrones, procesamiento de datos, visión por computadora. (Jiménez, 2013)

2.5.1. Neurona biología y artificial

Las redes neuronales se crean como inspiración de las neuronas cerebrales, tratando de imitar su comportamiento el cual se base en la interconexión de neuronas y

la transmisión de señales eléctricas que procesan la información, una red neuronal cerebral está conformada por cuatro partes que son: las dendritas los cuales transmiten las señales eléctricas, el soma que se encarga de la información, el axón que se encarga de transmitir la información hacia otra neurona y la sinapsis que es contacto electroquímico entre neuronas. (Morillo Tarapues, 2020)

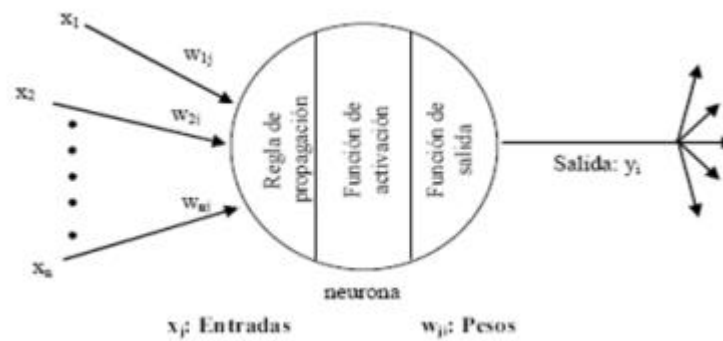
Figura 3: *Representación de una neurona biológica*



Fuente: (Manchado, 2018, pág. 5)

Las redes neuronas artificiales son una sub área de la inteligencia artificial, a su vez las redes neuronales forman parte del aprendizaje profundo el cual se basa en la múltiple interconexión entre las redes y diferentes topologías de estas. El funcionamiento de una neurona se comprende de entradas hacia las neuronas las cuales a su vez se interconectan con otras neuronas las cuales se encargan del aprendizaje, a esta interconexión se le conoce como los pesos que nos indican la intensidad de la conexión entre neuronas, finalmente el resultado son las salidas donde se tienen los resultados del aprendizaje de la red neuronal. (Manchado, 2018)

Figura 4: Estructura de una neurona artificial

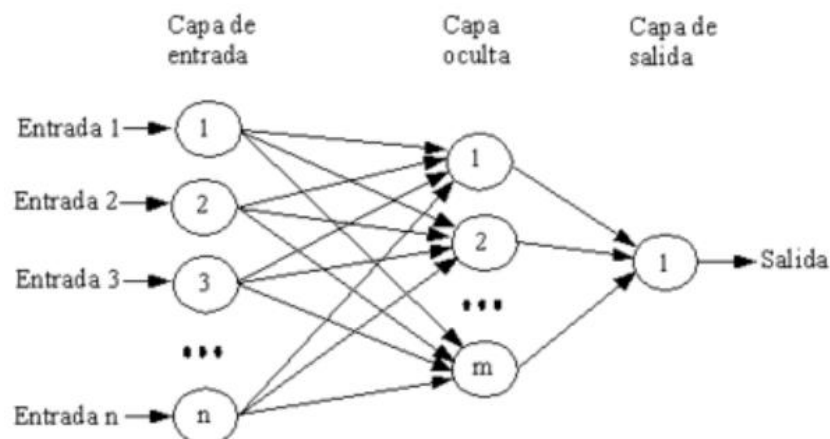


Fuente: (Manchado, 2018)

2.5.2. Topología de una red neuronal

Las redes neuronales pueden construirse e interconectarse de varias formas a esto se le conoce como la topología de una red neuronal, donde los componentes principales son la capa de entrada, capas intermedias y capa de salida, las neuronas que puede tener cada capa dependen del tipo de problema resolver, la capa de entrada se encarga de recibir la información o variables, las capas intermedias se utilizan para establecer relaciones entre las variables y en la capa de salida se obtiene el resultado. (Serrano, Soria, & Martín, 2009)

Figura 5: Topología de una red neuronal

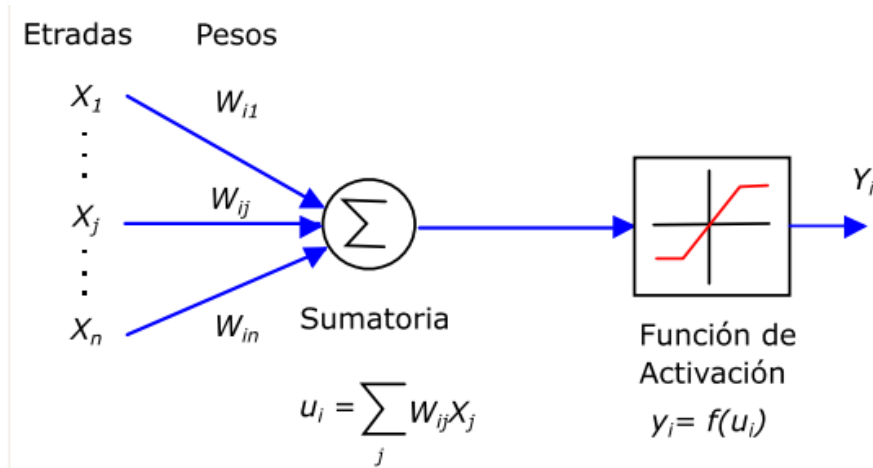


Fuente: (Serrano, Soria, & Martín, 2009)

2.5.3. Elementos de una red neuronal

Una red neuronal está compuesta por la interconexión de varias neuronas, la figura 6 muestra una sola neurona con sus respectivos elementos. (Fernando, 2020)

Figura 6: Elementos de una neurona artificial



Fuente: (Fernando, 2020, pág. 23)

Entradas

Es la información que recibe la primera capa de entrada, las cuales también se les conoce como los datos de entrenamiento.

Pesos

Los pesos establecen la relación o interconexión entre las neuronas, donde un valor positivo establece una mayor conexión, un valor negativo disminuye la conexión y un valor cero establece la nula conexión entre neuronas. (Fernandez, 2014)

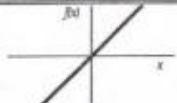
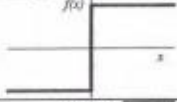
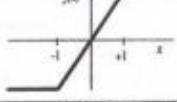
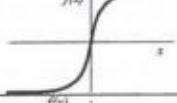


Función de activación

El objetivo de una función de activación es que ante un determinado valor de entrada hacia la neurona, determina un valor de salida, existen varias funciones de activación y se usan dependiendo del problema a solucionar por ejemplo se puede usar una función identidad si queremos realizar una predicción ya que esta nos devuelve valores

continuos, por el contrario si queremos realizar un problema de clasificación podemos una usar una función sigmoidea ya que como resultado nos da una probabilidad. (Jiménez, 2013)

Funciones de activación:

Figura 7: Funciones de activación

| | Función | Rango | Gráfica |
|----------------|---|-----------------------------|--|
| Identidad | $y = x$ | $[-\infty, +\infty]$ |  |
| Escalón | $y = \text{sign}(x)$ $y = H(x)$ | $\{-1, +1\}$ $\{0, +1\}$ |  |
| Línea a tramos | $y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$ | $[-1, +1]$ |  |
| Sigmoidea | $y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$ | $[0, +1]$ $[-1, +1]$ |  |
| Gaussiana | $y = Ae^{-bx^2}$ | $[0, +1]$ |  |
| Sinusoidal | $y = A \text{sen}(ax + \varphi)$ | $[-1, +1]$ |  |

Fuente: (Manchado, 2018, pág. 9)

Salidas

Las salidas son el resultado del proceso de aprendizaje de la red neuronal, dependiendo de los valores a la salida de las funciones de activación (Fernandez, 2014)

Función de coste

Son un indicativo de si es que la red neuronal se entrenó bien o mal, establece el error entre la variable que se tiene como objetivo contra la salida de la red neuronal, en la etapa de entrenamiento se espera que esta función se minimice lo cual será un buen indicativo de que entre la variable objetivo y la salida de la red neuronal existe poco error, las funciones de coste más utilizadas son: error absoluto medio (MAE), error de porcentaje



absoluto medio (MAPE), error cuadrático medio(MSE), binary cross entropy, categorical crossentropy. (Manchado, 2018)

Optimizador

Se encargará de actualizar los pesos de la red neuronal con el fin de minimizar el error de la función de coste. (Cabrera, 2014)

Métrica

Es un indicador para saber cómo de bueno es el modelo, a su vez se utiliza para comparar modelos y obtener los mejores resultados, el indicador más importante es la precisión (accuracy) (Manchado, 2018)

2.5.4. Aprendizaje y Entrenamiento

Para realizar el proceso de aprendizaje de la red neuronal se utilizan técnicas para minimizar el error, optimizar el modelo mediante una “función de recompensa” que pueda cambiar el valor de los pesos que son los que se encargan de definir la conexión entre las neuronas y esto da como resultado las salidas óptimas de la red. Es por eso que es necesario utilizar funciones que puedan medir el error del modelo en un instante dado. (Jiménez, 2013)

Según (Manchado, 2018) las etapas del proceso de aprendizaje de la RNN son:

1. **Inicialización de la red:** Se genera pesos aleatorios y con estos se calcula el primer error.
2. **Hacia adelante:** Se ingresa los datos hacia la red neuronal, que avanza hasta la salida realizando los cálculos con los pesos asignados y pasando por las funciones de activación.
3. **Cálculo del error:** Se calcula el error entre la predicción y el valor real
4. **Retro propagación del error:** Se programa el error por toda la red desde la salida de la red hasta la entrada, actualizándose así los pesos.



5. **Ajuste:** Se ajusta los pesos en cada iteración que realiza el algoritmo del descenso del gradiente

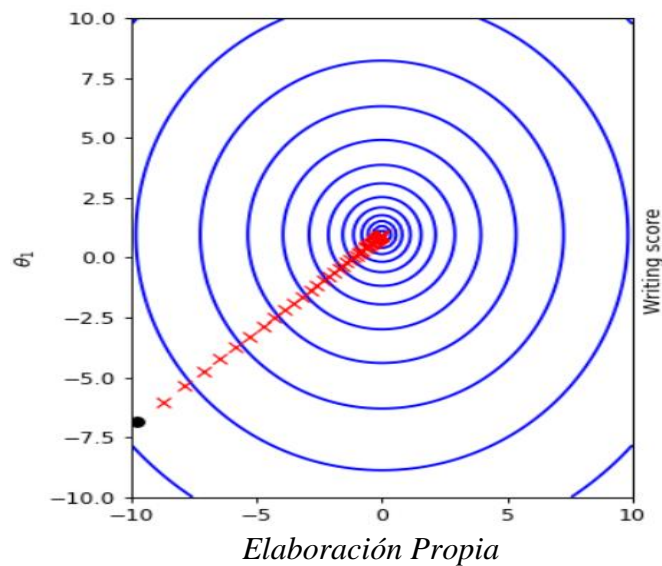
Sobre el entrenamiento es el proceso de alimentar una parte de los datos totales para que la red pueda identificar y aprender patrones en la serie temporal. Este proceso es para ajustar los valores de peso y sesgo utilizados para realizar la proyección después del proceso de entrenamiento.

Backpropagation

Es el algoritmo más usado el cual se encarga de obtener el error en la capa de salida y luego este error lo propaga por toda la red neuronal hasta la capa de entrada, con la finalidad de encontrar la contribución del error de cada neurona. Este algoritmo define una función $F(x)$ la cual es diferenciable en un entorno de a , lo cual le permite disminuir rápidamente buscando el mínimo de una función para lo cual utiliza el negativo del gradiente y para asegurar llegar al mínimo de la función se utiliza la tasa de aprendizaje “ μ ” que no es más que los pasos con los que se desciende al mínimo de la función. La siguiente ecuación describe el algoritmo del descenso del gradiente que no es más que restar el termino $\mu \nabla F(a_n)$ a a_n ya que se busca moverse en la dirección contraria al gradiente, buscando el mínimo de la función. (Montaño, 2002)

$$a_{n+1} \leq a_n - \mu \nabla F(a_n)$$

Figura 8: *Descenso del gradiente*

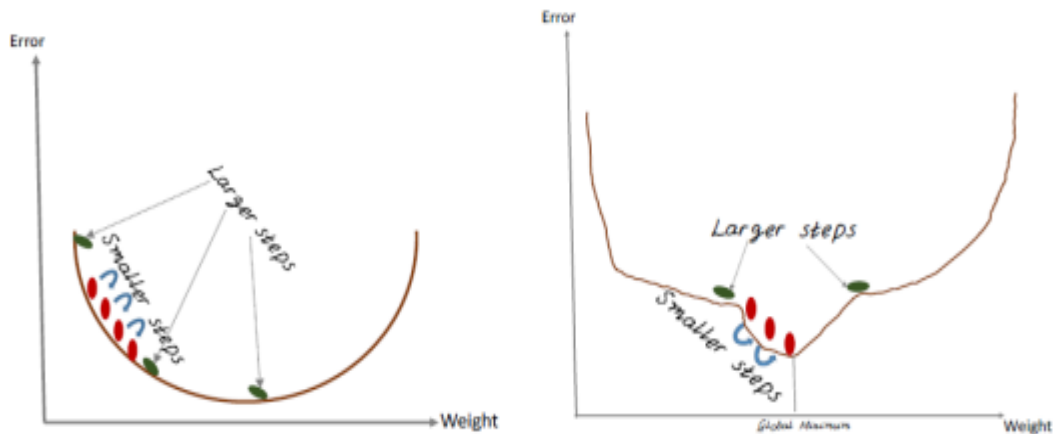


Manchado (2018) indica que el objetivo del algoritmo del descenso del gradiente es descender hasta encontrar un mínimo global, a su vez nos dice que para realizar este algoritmo se debe de seguir los siguientes pasos.

- Establecer un valor para la tasa de aprendizaje μ
- Realizar el cálculo del gradiente que permita encontrar la dirección del mínimo global y evitar mínimos locales.
- Realizar varias iteraciones que permitan moverse hacia la dirección del mínimo global hasta que el vector de gradiente sea lo más cercano a cero

La siguiente imagen representa el algoritmo del descenso del gradiente el cual como se mencionó anteriormente busca encontrar el mínimo global que minimice la función de coste, la figura nos dice que para un determinado valor de la tasa de aprendizaje o pasos del descenso del gradiente se recorre cierta cantidad sobre la función, ya que si se tiene una tasa de aprendizaje elevada el algoritmo no podrá converger por lo que es recomendable utilizar un valor pequeño para la tasa de aprendizaje y el algoritmo puede converger a un valor mínimo. (Manchado, 2018)

Figura 9: Problemas de la tasa de aprendizaje



Fuente: (Manchado, 2018)

2.5.5. Clasificación de las redes neuronales

Se pueden clasificar de acuerdo al mecanismo de aprendizaje como muestra la figura 10.

Figura 10: Clasificación de las redes neuronales artificiales

| | | | | |
|----------------|----------------------------------|--------------------------------------|-------------------------------------|-----------------------|
| SUPERVISADO | UNIDIRECCIONAL | Perceptrón | Rosenblatt | 1958 |
| | | Adaline | Widrow y Hoff | 1960 |
| | | Perceptrón Multicapa Backpropagation | Rumelhart, Hinton y Wilians | 1986 |
| | | Correlación en cascada | Fahiman y Lebeire | 1990 |
| | | Máquina de Bolzman | Ackel, Hinton y Sejnowski | 1985 |
| | | Learning Vector Quantization | Kohonen | 1988 |
| | General Regreción Neural Network | Specht | 1991 | |
| | RECURRENTES | Recurrent bacpropagation | Pineda | 1989 |
| | | Brian Sate in a Box | Andersan, Silverstein, Titz y Jones | 1977 |
| Jordan | | Jordan | 1986 | |
| NO SUPERVISADO | UNIDIRECCIONALES | Neocognitrón | Fukushima, Miyake e Ito, Fukushima | 1983; 1988 |
| | | Principal Component Analysis | Oja; Hertz | 1982; 1991 |
| | | Mapa autoorganizado | Kohonen | 1982; 1995 |
| | RECURRENTES | Red de Hopfield | Holfield | 1982 |
| | | Resonancia Adaptativa(ART) | Carpenter, Grossberg y Rosen | 1987; 1990 1991 |
| | | Bidirectional Associative Memory | Kosko | 1992 |
| | | Radial Bassic Fuction | Bromheard y Love; Moody y Darken | 1988; 1989 |
| HIBRIDO | Contrapropagación | Hetcht - Nielsen | 1990 | |

Fuente: (Serrano, Soria, & Martín, 2009)

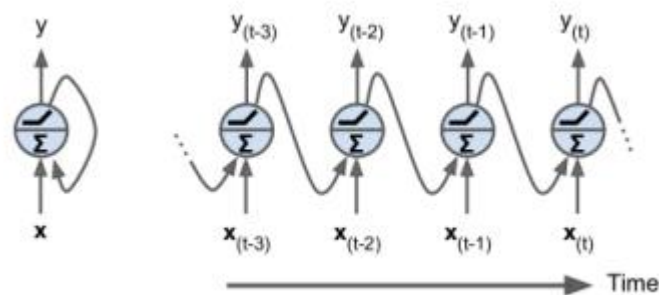
2.6. REDES NEURONALES RECURRENTES

Las redes neuronales recurrentes surgen con la finalidad de resolver problemas en las que se tiene una secuencia de datos como lo puede ser el procesamiento de imágenes

o las series de tiempo, ese tipo de redes permiten guardar información sobre datos anteriores de manera eficiente. (Manchado, 2018)

La siguiente figura ejemplifica una sola red neuronal recurrente donde se observa que en la entrada ingresa el valor actual y el valor anterior y como resultado se tiene el valor objetivo actual, a su vez se observa mejor la red desglosada a través del tiempo donde se puede ver como una neurona recibe una entrada y produce una salida que se lo envía como entrada a sí misma. (Fernando, 2020)

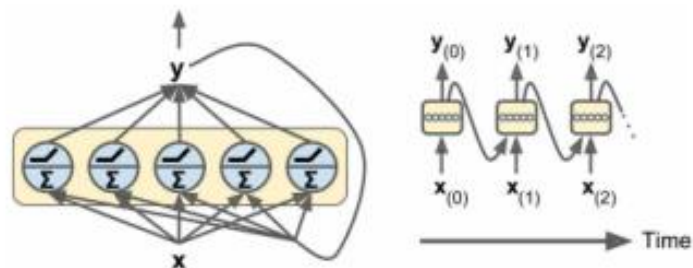
Figura 11: Neurona recurrente simplificada a través del tiempo



Fuente: (Manchado, 2018, pág. 15)

La siguiente figura nos muestra como la RNR a través del tiempo recibe la entrada $x_{(t)}$ y la salida anterior, $y_{(t-1)}$, esto lo podemos observar en la figura 11 y agrupando estas neuronas podemos obtener capas de RNN como muestra la figura 12.

Figura 12: Red neuronal recurrente a través del tiempo



Fuente: (Manchado, 2018)

“Cada neurona recurrente tiene dos conjuntos de pesos; uno para los inputs $x_{(t)}$ y otro para las salidas del paso temporal anterior $y_{(t-1)}$. Llamemos a estos pesos w_x y w_y . La salida $y_{(t)}$ de una neurona recurrente se puede calcular de la siguiente manera”:
(Manchado, 2018, pág. 16)

$$y_{(t)} = \varphi(x_{(t)}^T * w_x + y_{(t-1)}^T * w_y + b)$$

“donde b es el sesgo (bias) y $\varphi(\dots)$ es la función de activación. Podemos resumir la expresión anterior mediante el álgebra lineal de la siguiente manera.” (Manchado, 2018, pág. 16)

$$y_{(t)} = \varphi(X_{(t)} * W_x + Y_{(t-1)} * W_y + b) = \varphi([X_{(t)} * Y_{(t-1)}] * W + b)$$

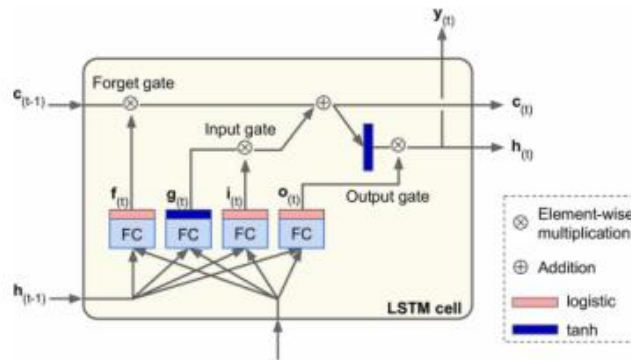
Donde:

- $Y_{(t)}$: Matriz que contiene las salidas
- $X_{(t)}$: Matriz que contiene las entradas
- $w_{(x)}$: Matriz de pesos de las entradas
- $w_{(y)}$: Matriz de pesos de las salidas
- b : Bías o sesgo

2.6.1. Red neuronal recurrente LSTM (Long short-term memory)

Las redes neuronales recurrentes LSTM pueden recordar datos relevantes en secuencias de tiempo complejas por lo que se les diferencia de las redes neuronales recurrentes simples debido a que pueden tener memoria a corto y largo plazo. (Fernando, 2020)

Figura 13: Célula LSTM



Fuente: (Manchado, 2018, pág. 18)

Según Manchado (2018), una célula LSTM es igual a una célula normal salvo que su estado ahora se divide en dos vectores, $h_{(t)}$ y $c_{(t)}$ (c viene de 'célula'). $h_{(t)}$ hace referencia al estado de corto plazo y $c_{(t)}$ al de largo plazo. “La principal idea de la célula es que la red pueda aprender qué almacenar en el estado de largo plazo, qué desechar y qué leer de ellos. Si nos fijamos en el término de largo plazo $c_{(t-1)}$ atraviesa la red de izquierda a derecha, primero pasa por una puerta de olvido (forget gate), deshaciéndose de algunos datos almacenados, luego añade nuevas informaciones que llegan de la puerta de entrada (input gate) y el resultado $c_{(t)}$ se envía directamente fuera de la célula y se deja como está y servirá de entrada para la siguiente época. Por lo tanto, en cada paso algunos recuerdos son eliminados mientras que otros nuevos se añaden. Además, una vez añadidos nuevos recuerdos, se hace una copia del estado a largo plazo y pasa por la función tangente hiperbólica para después ser filtrado por la puerta de salida (output gate). Este paso produce el estado de corto plazo $h_{(t)}$ (que es igual a la salida de la célula ($y_{(t)}$) para este paso temporal t).”. Las siguientes ecuaciones definen el proceso de cálculo de los estados a corto, largo plazo y la salida para un paso temporal. (Manchado, 2018, pág. 19)

$$i_{(t)} = \sigma(W_{xi}^T * x_{(t)} + W_{hi}^T * h_{(t-1)} + b_i)$$

$$f_{(t)} = \sigma(W_{xf}^T * x_{(t)} + W_{hf}^T * h_{(t-1)} + b_f)$$

$$o_{(t)} = \sigma(W_{xo}^T * x_{(t)} + W_{ho}^T * h_{(t-1)} + b_o)$$

$$g_{(t)} = \tanh(W_{xg}^T * x_{(t)} + W_{hg}^T * h_{(t-1)} + b_g)$$

$$c_{(t)} = f_t \otimes c_{t-1} + i_t \otimes g_t$$

$$y_{(t)} = h_t = o_t \otimes \tanh(c_t)$$

Donde:

- $W_{xi}, W_{xf}, W_{xo}, W_{xg}$: Matrices de conexión
- $W_{hi}, W_{hf}, W_{ho}, W_{hg}$: Matrices de conexión
- b_i, b_f, b_o y b_g : Bias

2.7. DEFINICIÓN DE TÉRMINOS

Energía: Se entiende por energía a la capacidad que poseen los cuerpos para efectuar un trabajo. Es lo que permite que suceda casi todo en el universo, la vida, el movimiento, la electricidad, el fuego, el ruido, el viento, calor, etc.

(Villanueva Ure, 2010)

Energía eléctrica: Es la forma de energía más conocida y difundida actualmente ya que puede convertirse en otras formas de energía como la calefacción, iluminación, energía mecánica. Esta energía proviene de la conversión de energía mecánica por medio de generadores. (Villanueva Ure, 2010)

Predicción eléctrica: La predicción de demanda de energía permite prever el futuro comportamiento de la demanda eléctrica, para realizar la predicción se utilizan modelos estadísticos, econométricos, redes neuronales.

Demanda de energía eléctrica: Se refiere a la cantidad de potencia eléctrica que un consumidor utiliza y esta varía dependiendo a la actividad que realiza por



lo que la demanda de energía puede ser más elevada para el sector industrial que para el sector urbano. (Fernando, 2020)

Red neuronal: Una red neuronal es un modelo matemático el cual se inspiró del funcionamiento del cerebro humano, la estructura básica de una red neuronal son las capas de entrada, intermedia y salida a su vez cada capa posee cierta cantidad de neuronas.

Aprendizaje automático: Las redes neuronales también se pueden clasificar según el tipo de aprendizaje, dentro de los cuales tenemos el aprendizaje supervisado y no supervisado, el aprendizaje supervisado se basa en que al modelo se le entrega los datos de entrada asociado a una salida mientras que el aprendizaje no supervisado se diferencia en que no se le suministra una salida es por eso que este tipo de aprendizaje aprende solo con los datos de entrada. (Serrano, Soria, & Martín, 2009)

CAPITULO III

MATERIALES Y MÉTODOS

3.1. MATERIALES

- Laptop para el preprocesamiento de datos y elaboración de la red neuronal
- Software Python
- Librerías Keras, TensorFlow, pandas, etc
- Plataforma Google Colaboratory

3.2. CLASIFICACIÓN DE LA INVESTIGACIÓN

El trabajo de enfoque cuantitativo, su diseño de la investigación es del tipo no experimental, el tipo de investigación es predictiva ya que tiene como propósito prever o anticipar situaciones futuras, el enfoque es del tipo cuantitativo debido a que se toma como centro del proceso de investigación las mediciones numéricas a su vez se utiliza técnicas estadísticas para interpretar y comparar el porcentaje de error del modelo de predicción. (Sampieri R. H., 2014)

3.3. VARIABLES DE LA INVESTIGACIÓN

3.3.1. Definición de las variables

Tabla 1. *Definición de las variables*

| Variable | Tipo | Concepto |
|---|-------------------------------|--|
| Error de predicción de la demanda de energía eléctrica | Cuantitativa. Dependiente | Es la diferencia entre el valor real de la demanda de energía eléctrica y el valor como resultado de dicha predicción. |
| Arquitectura de la red neuronal | Cuantitativa Independiente | Se refiere al número de neuronas, capas, conexiones entre neuronas. |

Elaboración propia

3.3.2. Operacionalización de variables

Tabla 2. Operacionalización de variables

| Variable | Indicador | Técnica | Instrumento |
|--|---|-------------|---------------------------|
| Error de predicción de la demanda de energía eléctrica | Error absoluto porcentual promedio (MAPE) | Estadística | Python |
| Arquitectura de la red neuronal | Error absoluto porcentual promedio (MAPE) | Estadística | Python, TensorFlow, Keras |

Elaboración propia

3.4. POBLACIÓN Y MUESTRA

3.4.1. Población

Sistema eléctrico de la empresa Electro Puno S.A

3.4.2. Muestra

Barra de 10kv la cual se encuentra en la ciudad de Juliaca que a su vez influye sobre los alimentadores (5004, 5005, 5006, 5008, 5009) que se encuentran en la subestación de transformación Taparachi.

(Ver anexo 01 donde se muestra el diagrama unifilar de la subestación de transformación Taparachi)

3.5. TÉCNICAS DE RECOLECCIÓN DE DATOS

- Observación no experimental
- Análisis documental

3.5.1. Recolección de datos

Los pasos para la recolección de datos fueron los siguientes:

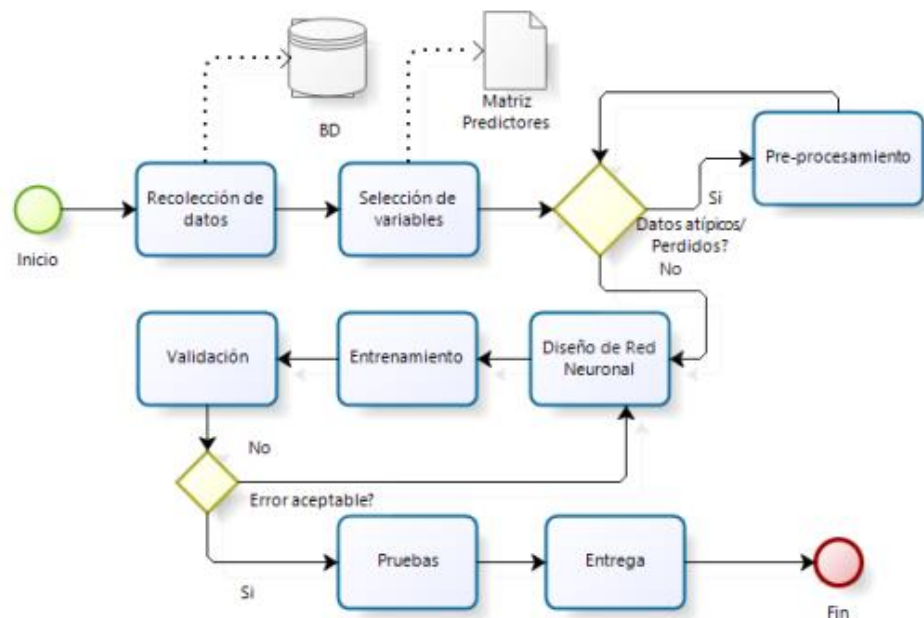
- Búsqueda de información en internet, libros y artículos

- Solicitud de la información del perfil de carga durante 3 años a la empresa Electro Puno

3.6. METODOLOGÍA PARA EL DISEÑO DE LA RED NEURONAL RECURRENTE

La metodología utilizada para este trabajo se puede resumir en la siguiente figura.

Figura 14: Metodología del trabajo de Investigación



Fuente: (Fernando, 2020, pág. 49)

3.6.1. Preprocesamiento de datos

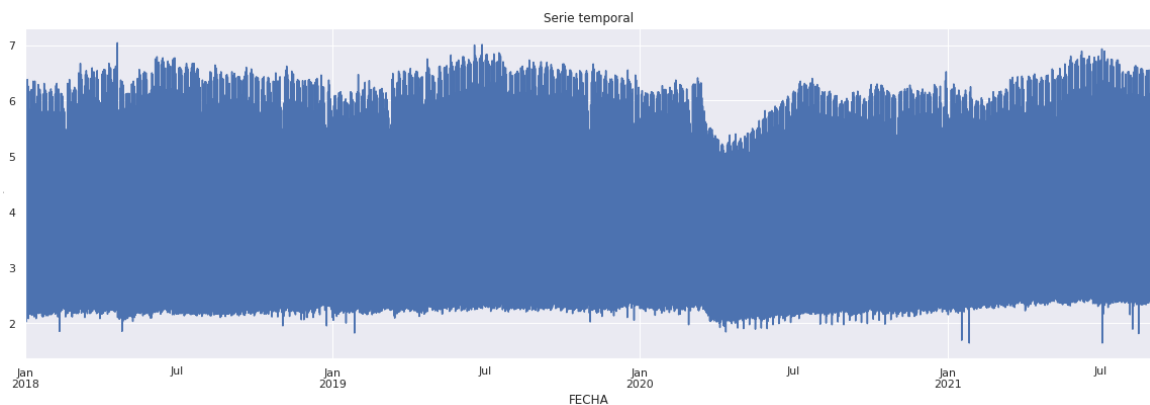
Consiste en realizar el análisis de la data de la demanda eléctrica suministrada por la empresa Electro Puno, visualizar su comportamiento, identificar valores atípicos y reemplazarlos para luego suministrar los datos a la red neuronal recurrente.

Los datos fueron suministrados por la empresa de distribución Electro Puno, se utiliza los datos de la barra de 10kv – Juliaca comprendiendo entre los años 2018, 2019, 2020 hasta el mes de agosto del 2021, se registran valores de cada 15 minutos, de los cuales 105216 valores comprenden los años 2018, 2019, 2020 los cuales serán utilizados para el entrenamiento y 23328 valores del año 2021, se utilizarán para probar la eficiencia de la red neuronal.

Análisis de los datos

En la figura 15 se muestra la serie de tiempo de la demanda de energía eléctrica desde el 2018 hasta el mes de agosto del 2021.

Figura 15: *Serie de tiempo de la demanda de energía eléctrica*

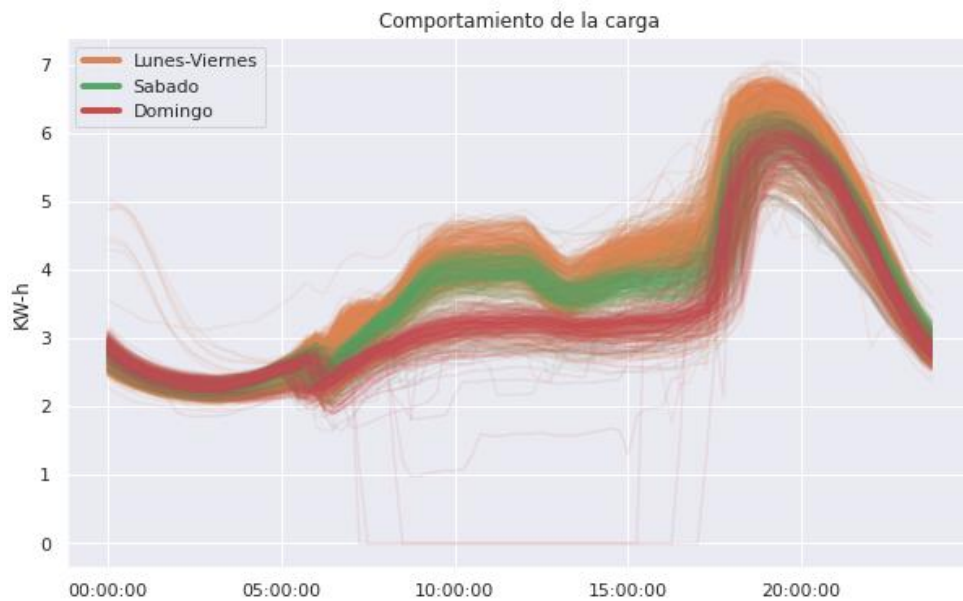


Elaboración propia

En la serie de tiempo también se puede observar que el pico máximo de la demanda de energía eléctrica está entre los meses de junio a agosto, esto se debe a que es la temporada más fría en el departamento de Puno, también se puede observar algo inusual con respecto a los anteriores años se registra una caída en la demanda de energía eléctrica a 5 MWh durante el año 2020 en el mes de abril.

Para poder visualizar patrones de la demanda de energía eléctrica se realiza otro grafico en periodos de 24 horas, donde se puede visualizar mejor la demanda dependiendo el día de la semana, siendo de lunes a viernes los días con más demanda seguidos por el sábado y domingo donde disminuye la demanda, a su vez también se observa la demanda pico que es alrededor de las 18 horas.

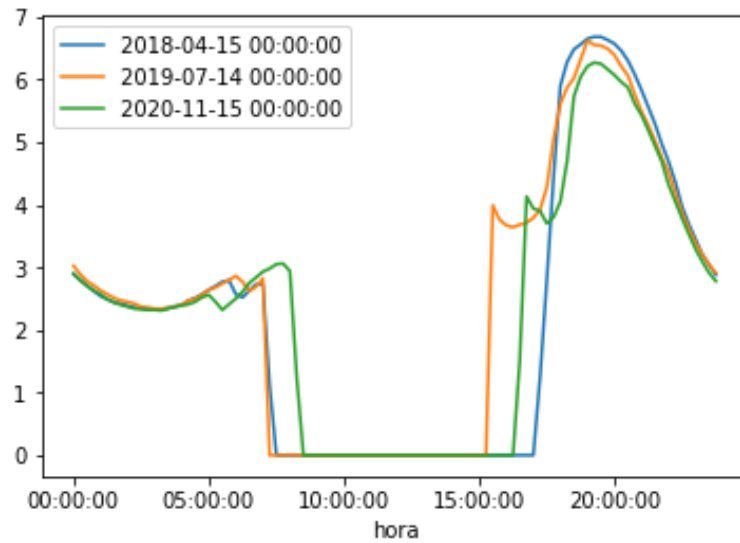
Figura 16: Perfil de carga diaria desde el 2018 hasta agosto del 2021



Elaboración propia

En la figura 16 también se puede ver valores atípicos que no siguen la tendencia de la gráfica y en la figura 17 se muestran los valores nulos, perdidos o no registrados, los cuales corresponden al 2018-04-15, 2019-07-14, 2020-11-15.

Figura 17: *Valores no registrados*



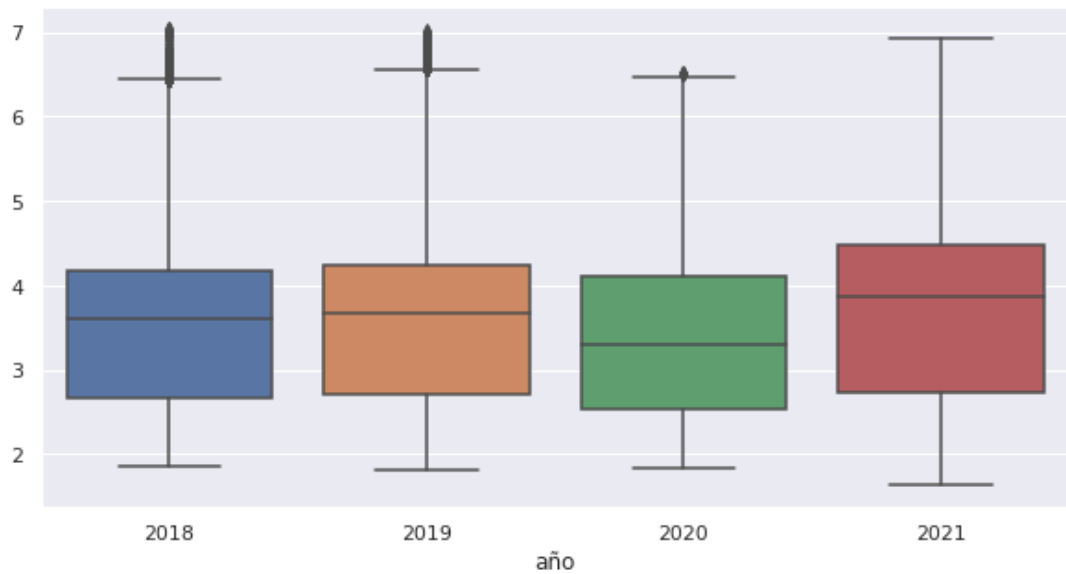
Elaboración propia

Detección y tratamiento de datos atípicos

Para evaluar la existencia de valores atípicos se utilizan los diagramas de cajas y el método de los cuartiles

- **Diagrama de cajas:** El diagrama de cajas es una presentación visual que describe como están distribuidas los valores por medio de sus cuartiles, de esta forma se pueden observar la mediana, cuartiles y valores atípicos. (Ojeda, 2014).

Figura 18: Diagrama de cajas



Elaboración propia

Se puede observar los valores atípicos moderados y extremos, a su vez el grafico nos muestra la media y los cuartiles

- Método de los cuartiles

El método de los cuartiles se utiliza para dividir los datos en partes que son Q1, Q2, Q3 que corresponde al 25%, 50%, 75% de los datos, a su vez también obtenemos los valores mínimos, y máximos de los datos y el cuartil dos es la media de los datos. (Sinaluisa Lozano, 2017)

Para realizar este método se utiliza la siguiente ecuación, donde:

$$P_Q = Q_{xn}/4$$

P_Q : Posición del cuartil

Q : Número del cuartil

P_Q : Numero de datos

Tabla 3. Cuartiles

| Cuartil | Q1 | Q2 | Q3 |
|------------|------|------|------|
| Porcentaje | 25% | 50% | 75% |
| Valor | 2.66 | 3.59 | 4.25 |

Elaboración propia

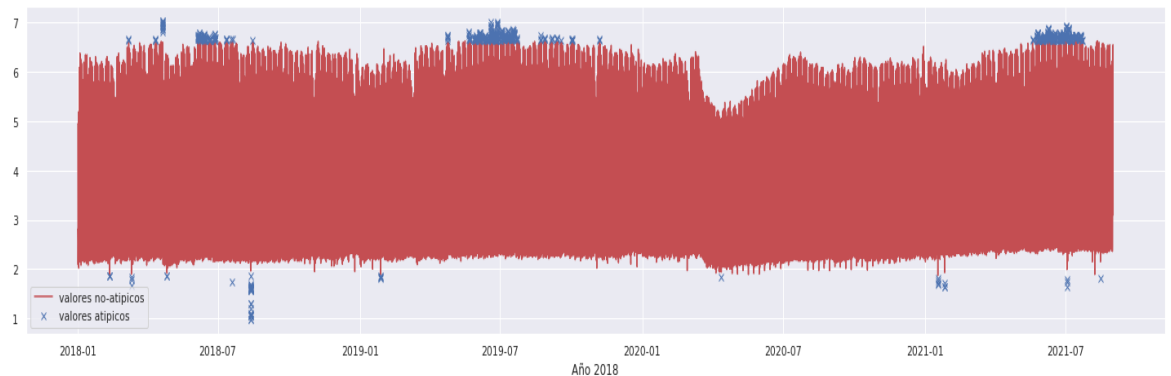
Para definir los límites superior e inferior de los valores atípicos de la serie de tiempo se aplica la siguiente fórmula.

$$L_{inf} = Q_1 - 0.5(Q_3 - Q_1)$$

$$L_{sup} = Q_3 + 1.5(Q_3 - Q_1)$$

Obteniendo así los valores de 6.635 y 1.865 como límites

Figura 19: Datos atípicos de la serie de tiempo

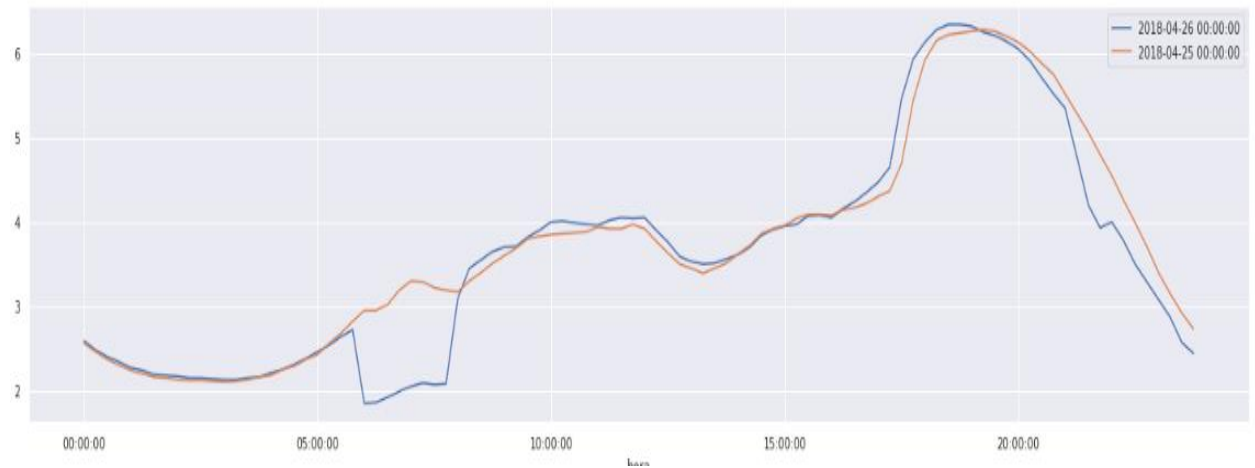


Elaboración propia

Se puede observar que este método está tomando como valores atípicos los valores entre los meses de junio, julio, agosto por lo que se decide conservar los datos ya que evidentemente en la región de Puno estos meses disminuye la temperatura y por ende debe aumentar la demanda de energía eléctrica debido al uso de calefactores.

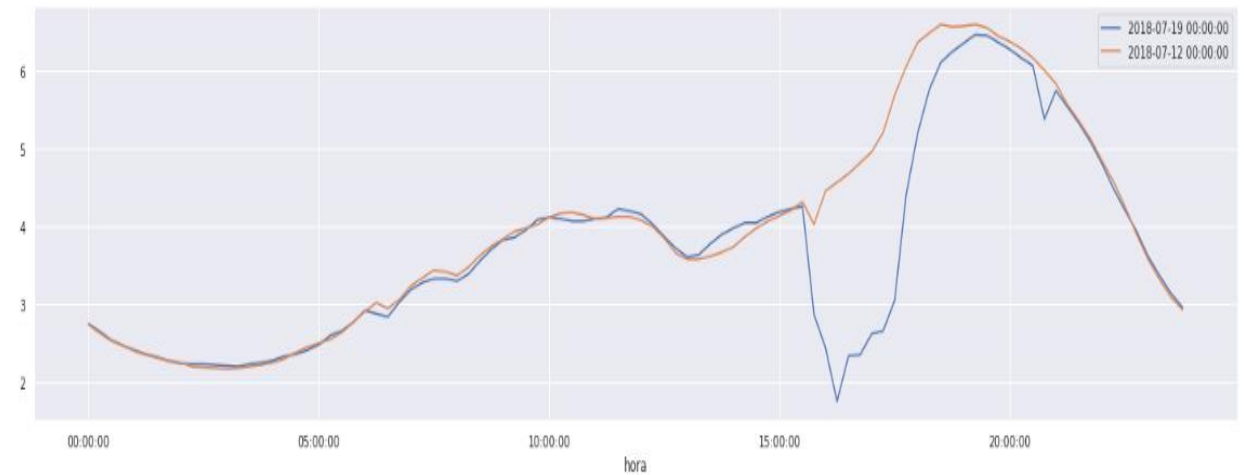
Las siguientes figuras muestran algunos de los valores atípicos encontrados, los cuales son de color azul y las figuras de color rojo representan los valores no atípicos de la serie de tiempo.

Figura 20: Valor atípico, correspondiente al 26-04-2018



Elaboración propia

Figura 21: Valor atípico, correspondiente al 19-07-2018



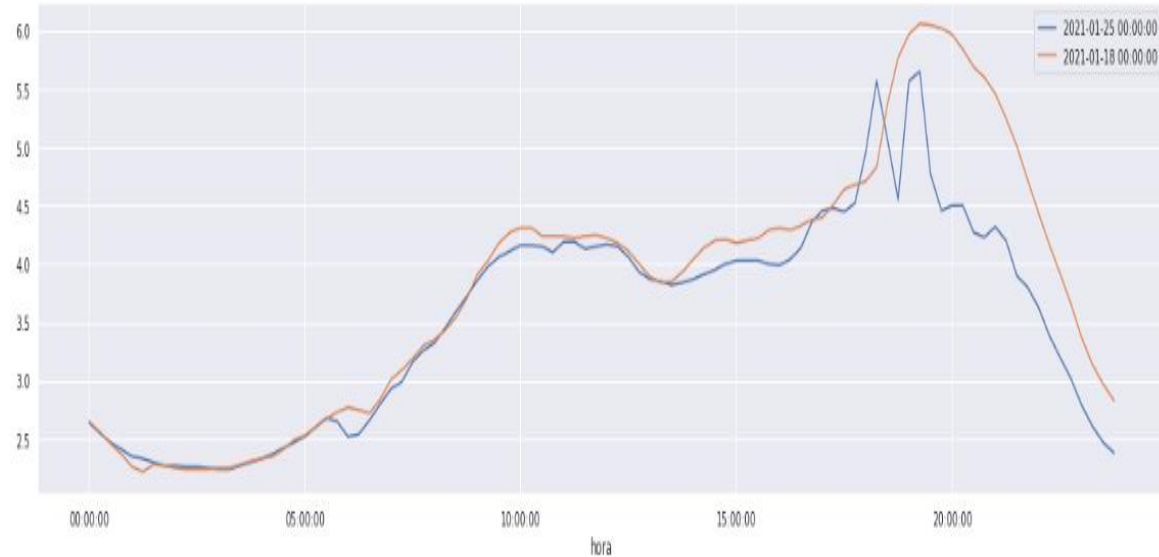
Elaboración propia

Figura 22: Valor atípico, correspondiente al 25-01-2021



Elaboración propia

Figura 23: Valor atípico, correspondiente al 12-08-2018



Elaboración propia



Normalización

La normalización es una técnica que se utiliza para normalizar los datos, reducir los datos a valores comprendidos entre 0 y 1 o -1 y 1, realizando esta técnica se logra una mejor etapa de entrenamiento logrando así que el algoritmo del descenso del gradiente encuentre el mínimo global de la función.

La siguiente formula describe la fórmula para realizar la normalización y obtener valores comprendidos entre 0 y 1

$$r_i = \frac{y_i - \min}{\max - \min}$$

Donde:

r_i = Dato normalizado

y_i = Dato original

min = valor mínimo de la serie

max = valor máximo de la serie

3.6.2. Desarrollo de las Redes

Como se vio en la etapa de preprocesamiento de datos la serie de tiempo es de una dimensión por lo que se realizara el desarrollo de la red neuronal recurrente del tipo univariante de varios pasos.

Preparación de datos de entrada

El modelo LSTM aprenderá una función que asigna una secuencia de observaciones pasadas como entrada hacia varias observaciones de salida. Como tal, la



secuencia de observaciones debe transformarse en múltiples ejemplos de los que el LSTM pueda aprender, por ejemplo, dada una serie temporal:

[10, 20, 30, 40, 50, 60, 70, 80, 90]

Podemos utilizar tres datos de entrada para pronosticar los siguientes dos datos de salida como se indica a continuación.

Entradas: [[10, 20,30]; [20, 30,40]; [30, 40,50]; [40, 50,60]]

Salidas: [[40, 50]; [50, 60]; [60, 70]; [70, 80]]

Entendido lo anterior se realizarán varios tipos de preparación de datos para ver con cual se tendrá mejores resultados el primero consiste en utilizar los datos del día actual para predecir la demanda del día siguiente y el segundo en utilizar los datos de la semana actual para predecir la demanda de la semana siguiente y el tercero será predecir el día actual con los datos de dos días anteriores

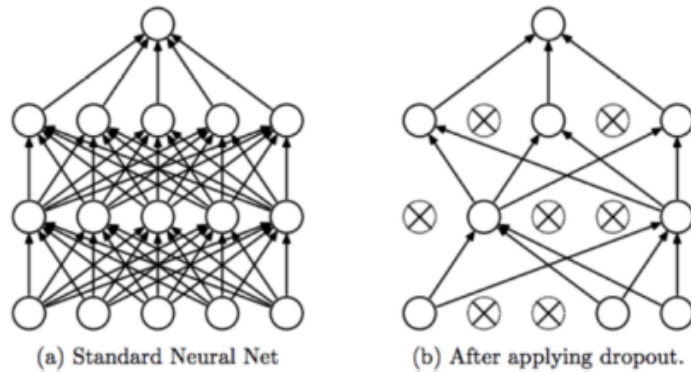
Teniendo los datos ya preparados se construye las redes neuronales recurrentes, se implementará usando células LSTM (Long Short Term Memory) con las siguientes arquitecturas, simple, apilada, bidireccional. Cabe recalcar que la implementación de cada red tiene varios parámetros como las funciones de activación, función de pérdida, y número de neuronas en cada capa con los que se experimentara hasta conseguir los mejores resultados. Existen algunas recomendaciones de cómo manejar estos parámetros, pero todo se basa en realizar pruebas y evaluar el error del modelo.

Dropout

La técnica del dropout se utiliza para eliminar las conexiones entre neuronas de modo que la red neuronal no sufra de sobreajuste, esto quiere decir que se memorice los datos y no sea capaz de generalizar que es lo que se espera de una red neuronal.

La siguiente imagen ejemplifica la técnica del dropout, se puede observar la pérdida de conexiones entre neuronas.

Figura 24: *Consecuencias del Dropout*



Fuente: (Manchado, 2018)

Se eligió un valor de dropout de 0.2 después de realizar varias pruebas

LSTM simple

El primer modelo tiene una sola capa de unidades LSTM con 48 entradas y una capa de salida con 48 valores que son la predicción demanda eléctrica de un día, se utiliza la técnica del dropout para tener mejores resultados, como métrica de error se utiliza la función de pérdida del error cuadrático medio (MSE) y como optimizador estimación del momento adaptativo (ADAM).

Figura 25: *Arquitectura LSTM simple*

```
# entrenar el modelo LSTM simple
def build_model(train_x,train_y, n_input,tensorboard):

    # parámetros
    epochs, batch_size = 50, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2], train_y.shape[1]
    # modelo
    model = Sequential()
    model.add(LSTM(100, activation='relu', input_shape=(n_timesteps, n_features)))
    model.add(Dropout(0.2))

    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # fit network
    history=model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size, callbacks=[tensorboard])
    return model,history
```

Elaboración propia

LSTM apilado

El segundo modelo tiene una capa LSTM con 48 entradas, una capa oculta LSTM y una capa de salida con 48 valores que son la predicción demanda eléctrica de un día, se utiliza la técnica del dropout para tener mejores resultados, como métrica de error se utiliza la función de pérdida del error cuadrático medio (MSE) como optimizador estimación del momento adaptativo (ADAM) y como función de activación el rectificador lineal unitario (RELU)

Figura 26: *Arquitectura LSTM apilado*

```
# entrenar el modelo LSTM apilado
def build_model_stacked_lstm(train_x,train_y, n_input,tensorboard):

    # parámetros
    epochs, batch_size = 50, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2], train_y.shape[1]
    # modelo
    model = Sequential()
    model.add(LSTM(100, activation='relu',return_sequences=True ,input_shape=(n_timesteps, n_features)))
    model.add(Dropout(0.2))
    model.add(LSTM(100, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # fit network
    history=model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size, callbacks=[tensorboard])
    return model,history
```

Elaboración propia

LSTM bidireccional

El tercer modelo tiene una capa LSTM bidireccional con 48 entradas y una capa de salida con 48 valores que son la predicción demanda eléctrica de un día, se utiliza la técnica del dropout para tener mejores resultados, como métrica de error se utiliza la función de pérdida del error cuadrático medio (MSE) como optimizador estimación del momento adaptativo (ADAM) y como función de activación el rectificador lineal unitario (RELU).

Figura 27: *Arquitectura LSTM bidireccional*

```
# entrenar el modelo
def build_model_bd_lstm(train_x,train_y, n_input,tensorboard):

    # parámetros
    epochs, batch_size = 50, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2], train_y.shape[1]
    # modelo
    model = Sequential()
    model.add(Bidirectional(LSTM(100, activation='relu'), input_shape=(n_timesteps, n_features) ))
    model.add(Dropout(0.2))
    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # fit network
    history=model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size, callbacks=[tensorboard])
    return model,history
```

Elaboración propia

Como se observa se utiliza como función de pérdida el error cuadrático medio (MSE) para calcular el error del modelo, posteriormente se utilizará el error porcentual medio (MAPE) para comparar los modelos, también se observa otro parámetro denominado optimizador, para este caso se eligió el tipo Adam.

3.6.3. Entrenamiento

Una vez creada la red se procede a entrenarla. Para ello se utilizarán los datos desde el año 2018 hasta el 2020 con registros de cada 30 minutos los cuales hacen un total de 52608 registros que se utilizaran como datos de entrenamiento y los datos de prueba hacen un total de 11664 registros los cuales se corresponden desde el mes de enero hasta el mes de agosto del 2021. Por lo tanto, se introduce la siguiente línea de código.

Figura 28: *Entrenamiento de la RNN*

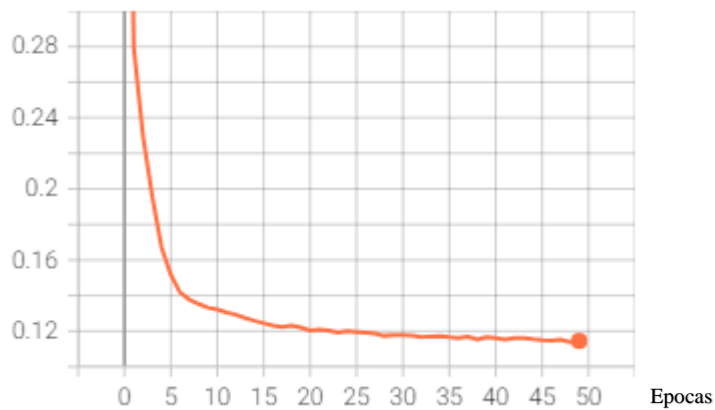
```
# fit network
history=model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size)
return model,history
```

Elaboración propia

Donde:

- Batch size: Cantidad de muestras con las que se actualiza el gradiente
- Epochs: Cantidad de épocas para entrenar el modelo

Figura 29: *Proceso de aprendizaje de la RNN, error vs épocas*



Elaboración Propia

La figura 29 muestra el proceso de aprendizaje de la red neuronal recurrente, se observa como disminuye la función de pérdida a medida que incrementa el número de épocas, además se observa que a partir de 20 épocas el error disminuye poco por lo que ya se considera un entrenamiento óptimo ya que el error se acerca a cero y converge. Cabe recalcar que si se incrementa el número de épocas se puede obtener mejores resultados, pero requiere más tiempo por lo que con 50 épocas y con un batch size de 13 es suficiente.

3.6.4. Comparación de modelos

Se utilizan los datos de prueba de la demanda de energía eléctrica desde enero del 2021 hasta agosto del 2021 para compararlos con los resultados obtenidos por el modelo de RNR LSTM para realizar el cálculo del desempeño se utilizará el Porcentaje de Error Medio Absoluto (MAPE)

$$MAPE = \frac{100}{N} \sum_{t=1}^n \left| \frac{y'_t - y_t}{y_t} \right|$$

Donde:

n : Numero de observaciones

y_t : Valor real



y'_t : Valor pronosticado por el modelo

Predicción con RNR LSTM Simple

Como se indicó en el apartado anterior sobre la preparación de datos que serán ingresados a la red LSTM, se realizaron dos diferentes matrices de entrada de los cuales el primero era, utilizar los datos del día anterior para predecir la demanda del día actual, del cual se obtuvo un MAPE de 5.5% y el segundo era utilizar los datos de la semana anterior para predecir la demanda de la semana actual, del cual se obtuvo un MAPE de 6.69%, por lo que podemos decir que el modelo predice mejor la demanda de energía eléctrica del día actual cuando se le suministra datos del día anterior, esto tiene sentido en cierta parte ya que el comportamiento de la demanda es similar los días lunes a viernes pero los sábados y domingos disminuye el comportamiento de la demanda de energía eléctrica. Los resultados del MAPE por días se detallan en la siguiente tabla

Tabla 4. *Resultados del MAPE*

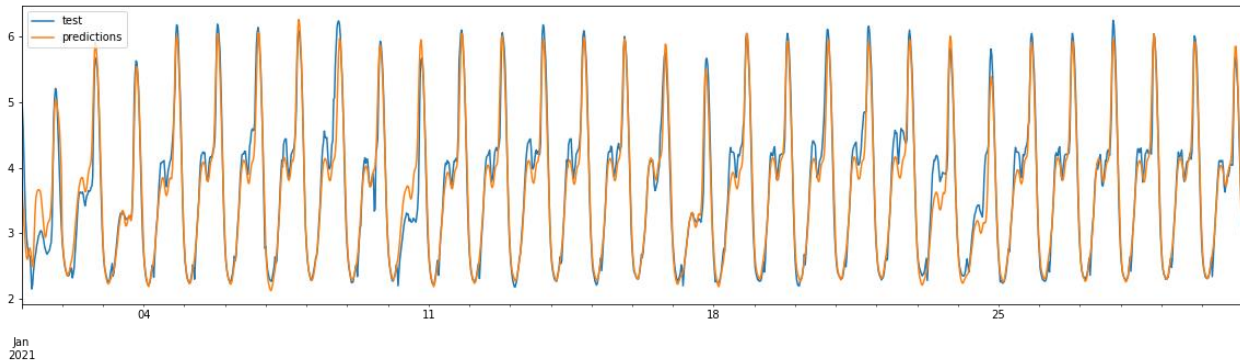
| Día de la semana | MAPE – día anterior | MAPE – semana pasada |
|-------------------------|----------------------------|-----------------------------|
| Lunes | 5.13 % | 11.55 % |
| Martes | 6.36 % | 4.71% |
| Miércoles | 5.48 % | 4.59 % |
| Jueves | 5.72 % | 4.86 % |
| Viernes | 6.82 % | 5.64 % |
| Sábado | 4.16 % | 4.09 % |
| Domingo | 4.89 % | 11.29 % |
| Global | 5.56 % | 6.69 % |

Elaboración propia

Se puede observar en la tabla que el modelo que utiliza los datos de la semana pasada para pronosticar la demanda de la semana actual tiene un MAPE menor, durante los días martes, miércoles, jueves, viernes, sábado, pero los días lunes y domingo

incrementa el MAPE, esto quiere decir que el modelo aprender mejor cuando se le suministra datos del día anterior para predecir la demanda eléctrica del día actual.

Figura 30: *Predicción de la demanda para el mes de enero del 2021 – LSTM simple*

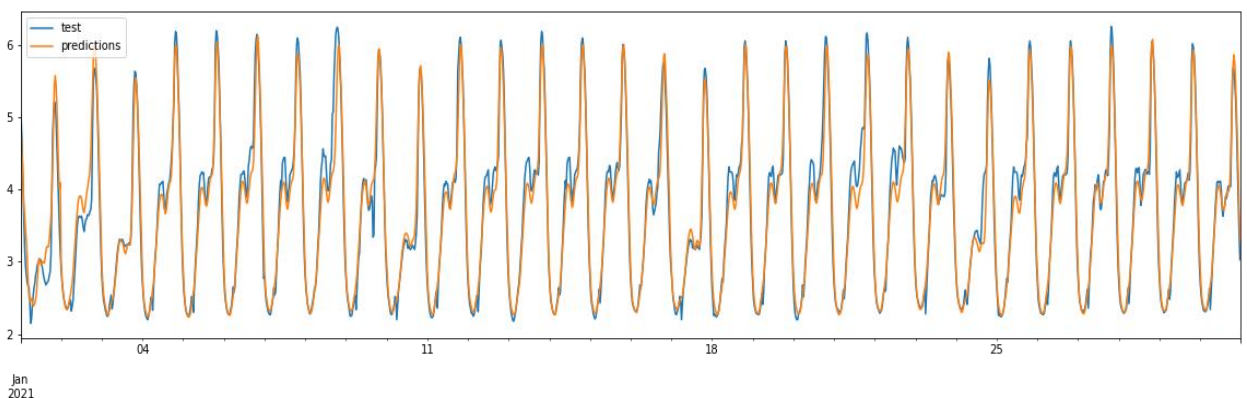


Elaboración Propia

La figura 30 muestra los resultados de las predicciones del modelo LSTM simple comparado con los datos de test del 2021, se puede observar que el modelo realiza una buena predicción además que puede seguir la estacionalidad natural de la demanda eléctrica, pero no llega a cubrir los picos de la demanda eléctrica.

Predicción con RNR LSTM Apilado

Figura 31: *Predicción de la demanda para el mes de enero del 2021 – LSTM apilado*



Elaboración propia

La figura 31 muestra los resultados de las predicciones del modelo LSTM apilado, se puede observar que tiene un mejor desempeño ya que mejora la predicción en los picos

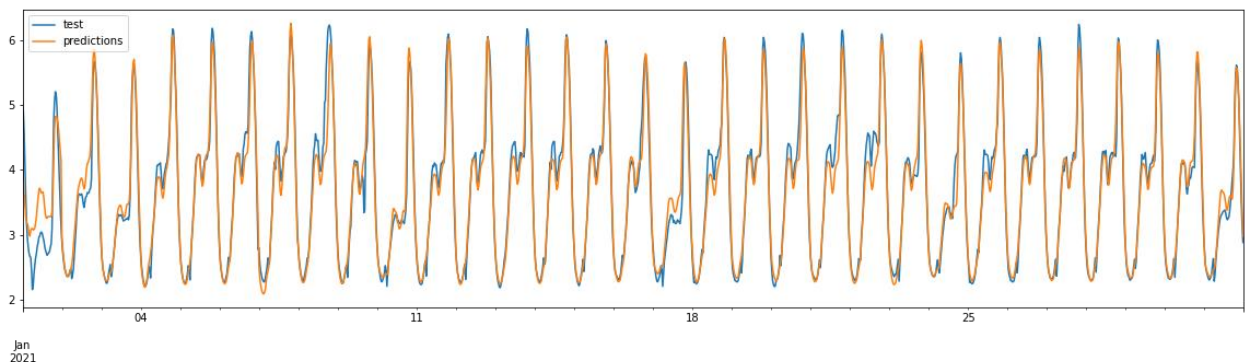
de la demanda de la energía eléctrica, a su vez la tabla 5 se observa como el MAPE disminuye

Tabla 5. Resultados MAPE LSTM Apilado

| Día de la semana | LSTM – Apilado |
|------------------|----------------|
| Lunes | 5.07 % |
| Martes | 5.31% |
| Miércoles | 5.26 % |
| Jueves | 5.32 % |
| Viernes | 5.83 % |
| Sábado | 3.43 % |
| Domingo | 3.92 % |
| Global | 4.88 % |

Predicción con RNR LSTM Bidireccional

Figura 32: Predicción de la demanda para el mes de enero del 2021 – LSTM Bidireccional



Elaboración propia

La figura 32 muestra los resultados de las predicciones del modelo LSTM bidireccional, no se observa una mejora notable, se puede comprobar en la tabla 6 donde se observa que el MAPE no disminuye significativamente.

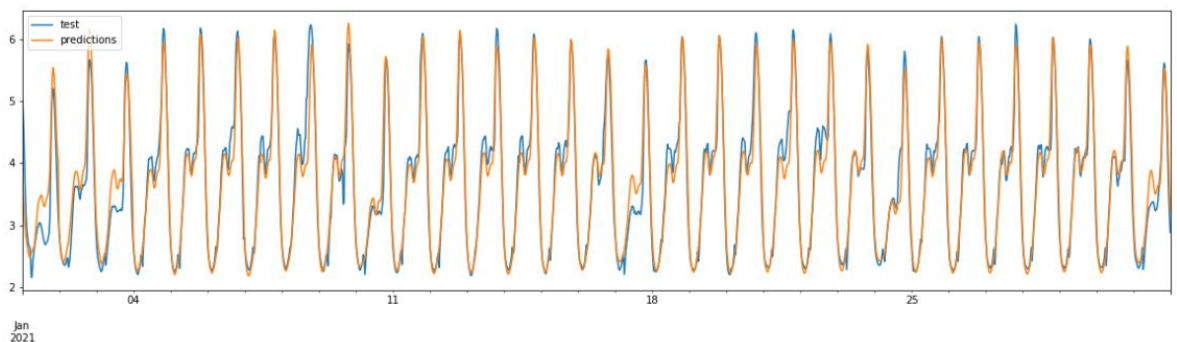
Tabla 6. Resultados MAPE LSTM Bidireccional

| Día de la semana | LSTM – Bidireccional |
|------------------|----------------------|
| Lunes | 5.14 % |
| Martes | 4.91% |
| Miércoles | 4.55 % |
| Jueves | 4.96 % |
| Viernes | 5.83 % |
| Sábado | 3.70 % |
| Domingo | 4.47 % |
| Global | 4.80 % |

Elaboración propia

Como se observan los resultados del MAPE del modelo de la Red neuronal recurrente LSTM bidireccional tiene mejores resultados se procede a realizar pruebas con este modelo como variar el número de entradas hacia la red hasta obtener un menor valor del MAPE, se realizan varias pruebas de donde se obtiene mejores resultados con 96 valores de entrada hacia la red.

Figura 33: Predicción de la demanda para el mes de enero del 2021 – LSTM Bidireccional con 96 valores de entrada



Elaboración propia

La figura 33 muestra los resultados obtenidos por ese modelo y la tabla número 7 muestra los resultados del MAPE donde se observa la gran mejora de este modelo donde se obtiene el mejor valor de predicción con un valor de 97.18% de precisión.



Tabla 7. Resultados MAPE LSTM Bidireccional

| Día de la semana | LSTM – Bidireccional |
|-------------------------|-----------------------------|
| Lunes | 2.58 % |
| Martes | 2.14% |
| Miércoles | 2.93 % |
| Jueves | 3.28 % |
| Viernes | 3.73 % |
| Sábado | 1.79 % |
| Domingo | 3.33 % |
| Global | 2.82 % |

Elaboración propia

CAPITULO IV

RESULTADOS Y DISCUSIÓN

Este capítulo está destinado a analizar los resultados obtenidos del modelo de predicción basado en los modelos de redes neuronales recurrentes, contra los datos de test que corresponde al mes de enero hasta el mes de agosto del 2021

4.1. DATOS DE ENTRADA A LA RED NEURONAL

Primero se tuvieron dos planteamientos, primero predecir la demanda de energía del día actual suministrándole a la red neuronal los datos de la demanda de energía del día anterior, el segundo fue predecir la demanda de energía del día actual suministrándole a la red neuronal los datos de la demanda de energía de la semana anterior, estos planteamientos tienen sentido ya que por ejemplo la demanda del miércoles es parecido a la demanda de martes de igual manera es parecido al martes de la semana anterior. Como resultados se obtuvo que el modelo predice mejor cuando se le suministraron los datos de la demanda de energía del día anterior.

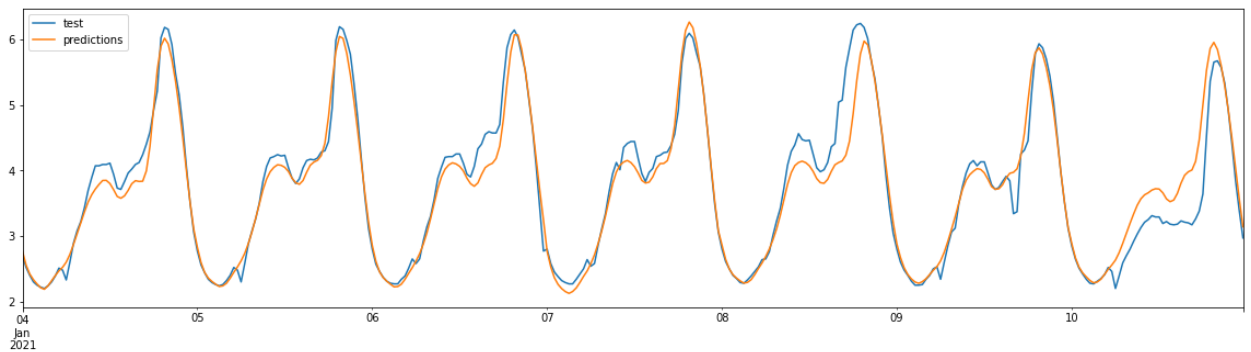
Tabla 8. Comparación de MAPE dependiendo los datos de entrada

| Día de la semana | MAPE – día anterior | MAPE – semana pasada |
|------------------|---------------------|----------------------|
| Lunes | 5.13 % | 11.55 % |
| Martes | 6.36 % | 4.71% |
| Miércoles | 5.48 % | 4.59 % |
| Jueves | 5.72 % | 4.86 % |
| Viernes | 6.82 % | 5.64 % |
| Sábado | 4.16 % | 4.09 % |
| Domingo | 4.89 % | 11.29 % |
| Global | 5.56 % | 6.69 % |

Elaboración propia

Para ambos modelos se usaron 48 valores de entrada hacia una RNN LSTM con 100 neuronas, 48 valores de salida y se utilizó dropout con un valor de 0.2.

Figura 34: Predicción de la demanda de energía eléctrica para una semana



Elaboración Propia

La figura 34 muestra la predicción del modelo para una semana se puede observar que el modelo de predicción basado en redes neuronales recurrentes es capaz de seguir la trayectoria de la curva original, evidenciando como la demanda disminuye los días sábados y domingos.

4.2. MODELO RNR LSTM SIMPLE VS LSTM APILADO VS LSTM

BIDIRECCIONAL

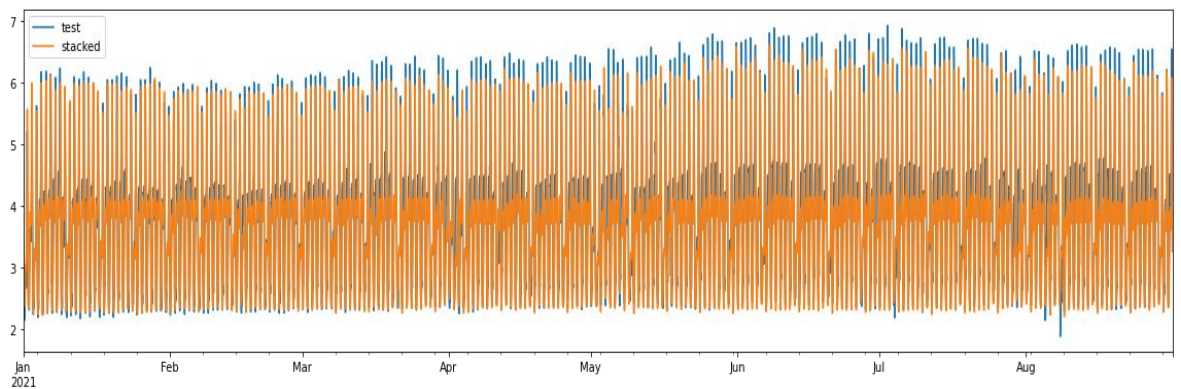
La tabla número 9 muestra los resultados obtenidos al realizar los tres modelos de redes neuronales recurrentes LSTM, a la RNN LSTM Apilada se le agregó una capa más con 100 neuronas y dropout de 0.2 se evidencia como disminuyó el MAPE en un 0.68% al agregar una capa más, por otra parte la RNN LSTM Bidireccional con 1 capa de 100 neuronas y dropout de 0.2 disminuye el MAPE en un 0.76%, vistos estos resultados se puede decir el modelo de la RNN LSTM bidireccional realiza una mejor predicción de la demanda de energía eléctrica, pero el entrenamiento demora más en realizarse por otra parte un modelo LSTM Apilado tarda menos en su entrenamiento, por lo que al agregarse más capas se podría obtener un menor valor del MAPE.

Tabla 9. Comparación de MAPE entre modelos LSTM

| Día de la semana | LSTM – Simple | LSTM – Apilado | LSTM – Bidireccional |
|------------------|---------------|----------------|----------------------|
| Lunes | 5.13 % | 5.07 % | 5.14 % |
| Martes | 6.36 % | 5.31 % | 4.91 % |
| Miércoles | 5.48 % | 5.26 % | 4.55 % |
| Jueves | 5.72 % | 5.32 % | 4.96 % |
| Viernes | 6.82 % | 3.83 % | 5.83 % |
| Sábado | 4.16 % | 3.43 % | 3.70 % |
| Domingo | 4.89 % | 3.92 % | 4.47 % |
| Global | 5.56 % | 4.88 % | 4.80 % |

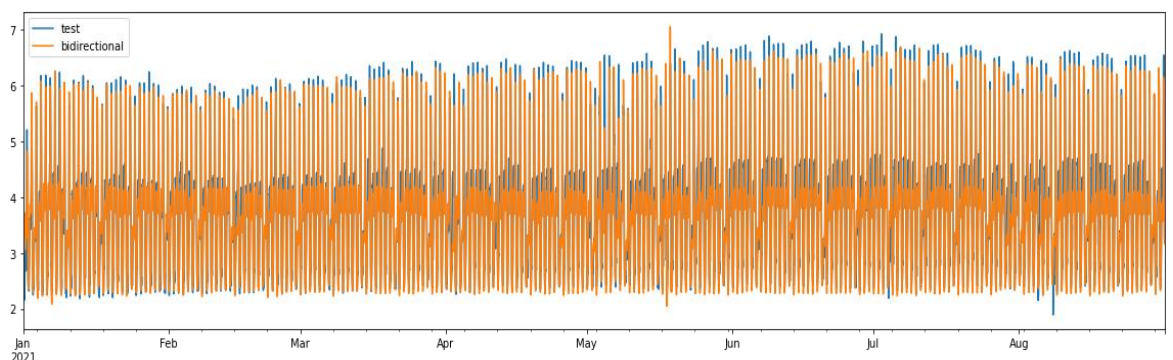
Elaboración propia

Figura 35: Predicción de la demanda de energía con LSTM Apilado



Elaboración propia

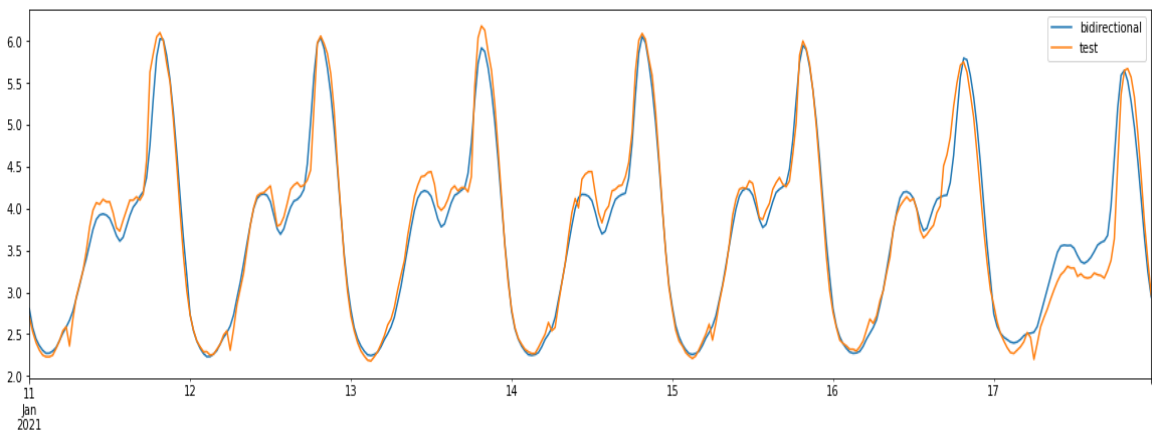
Figura 36: Predicción de la demanda de energía con LSTM Bidireccional



Elaboración propia

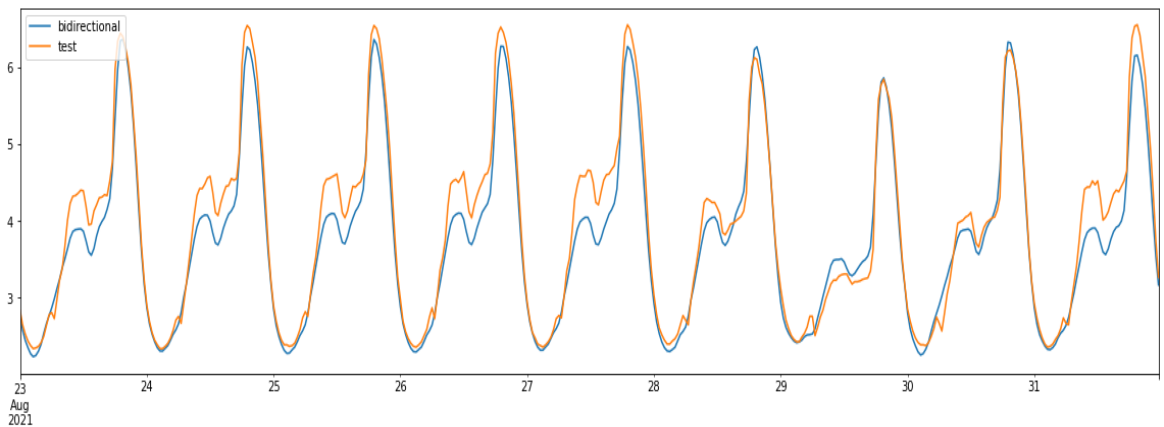
Las figuras número 35 y 36 muestran los resultados obtenidos, se puede observar que el modelo LSTM bidireccional logra abarcar los picos de la demanda de energía, pero también se observa que en ciertas partes de la gráfica sobrepasa los valores reales.

Figura 37: *Predicción de la demanda de energía eléctrica para la primera semana de enero del 2021 con el modelo LSTM Bidireccional*



Elaboración Propia

Figura 38: *Predicción de la demanda de energía eléctrica para los últimos 9 días del mes de agosto del 2021 con el modelo LSTM Bidireccional*



Elaboración Propia

Las figuras 37 y 38 muestran los resultados del modelo de la predicción basado en redes neuronales recurrentes LSTM bidireccional se puede observar que el modelo fue capaz de predecir la demanda de energía eléctrica para el año 2021 desde el mes de enero

hasta el mes de agosto que eran los datos que se tenían como test, la predicción fue de un 95.2% de exactitud.

4.3. LSTM BIDIRECCIONAL CON 96 VALORES DE ENTRADA

La siguiente tabla muestra los resultados obtenidos de la RNR LSTM Bidireccional con 48 valores de entrada contra el mismo modelo, pero con 96 valores de entrada, se puede observar cómo influye la cantidad de entradas hacia la red sobre el resultado obtenido, teniendo una diferencia de 1.98% del MAPE. Cabe recalcar que para ese modelo solo se modificó el número de entradas con respecto al modelo LSTM Bidireccional anterior.

Tabla 10. Comparación de MAPE LSTM Bidireccional de 48 entradas vs 96 entradas

| Día de la semana | MAPE – 48 entradas | MAPE – 96 entradas |
|------------------|--------------------|--------------------|
| Lunes | 5.14 % | 2.58 % |
| Martes | 4.91 % | 2.14% |
| Miércoles | 4.55 % | 2.93 % |
| Jueves | 4.96 % | 3.28 % |
| Viernes | 5.83 % | 3.73 % |
| Sábado | 3.70 % | 1.79 % |
| Domingo | 4.47 % | 3.33 % |
| Global | 4.80 % | 2.82 |

Elaboración propia

4.4. MODELO RNA – LSTM CON DATOS SIN PROCESAR Y PROCESADO

Una vez obtenidos los resultados de los modelos anteriores se procede a comparar los resultados del menor MAPE obtenido por el modelo LSTM Bidireccional con preprocesamiento de datos y sin preprocesamiento de datos, es decir con valores atípicos, perdidos.

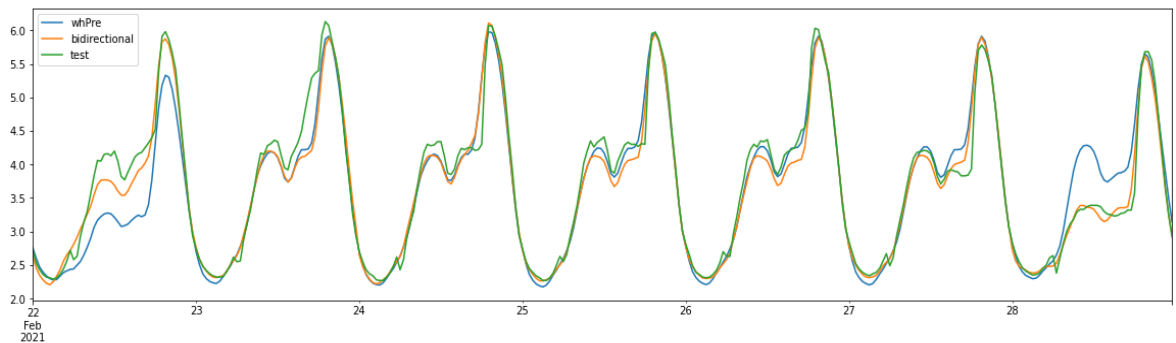
Tabla 11. Comparación del MAPE LSTM Bidireccional

| Día de la semana | Con procesamiento | Sin preprocesamiento |
|------------------|-------------------|----------------------|
| Lunes | 2.58 % | 13.29 % |
| Martes | 2.14% | 5.18% |
| Miércoles | 2.93 % | 5.17 % |
| Jueves | 3.28 % | 5.42 % |
| Viernes | 3.73 % | 6.67 % |
| Sábado | 1.79 % | 4.87 % |
| Domingo | 3.33 % | 11.44 % |
| Global | 2.82 % | 7.45 % |

Elaboración propia

La tabla número 11 evidencia que realizar el preprocesado de los datos disminuye el error del modelo habiendo una diferencia de 4.63 % del MAPE, cabe recalcar que ambos modelos tienen una capa con 100 neuronas y que en ambas se utilizó dropout de 0.2

Figura 39: Comparación de resultados con preprocesamiento y sin preprocesamiento de datos



Elaboración Propia

La figura 39 muestra los resultados de pronóstico de la demanda de energía eléctrica para una semana donde la gráfica de color azul es el resultado del modelo LSTM sin preprocesamiento de datos, la gráfica de color naranja es el modelo LSTM con preprocesamiento de datos, la gráfica de color verde son los valores reales correspondientes al año 2021. Por lo que se puede concluir que realizar el



preprocesamiento de datos mejora los resultados obtenidos por del modelo de predicción basado en redes neuronales recurrentes LSTM.

4.5. DISCUSIÓN DE RESULTADOS

Habiendo obtenido los resultados de los diferentes modelos de predicción basados en RNR LSTM, se pudo observar que el error de predicción varía dependiendo al modelo, tratamiento de datos, arquitectura de la red neuronal y las variables de entrada hacia la RNR. Las variables de entrada son un punto clave que puede definir el resultado del error de predicción por lo que lo más importante al realizar las diferentes consultas bibliográficas fue encontrar estas variables es así que en la tesis de (Fernandez, 2014) realiza un modelo tipo perceptrón multicapa con 48 valores de carga de un día anterior al pronosticado y 48 valores de carga de la semana pasada, días feriados, días festivos, de estas variables mencionadas se utilizó los 48 valores de carga de un día anterior y los 48 valores de carga de la semana pasada del cual se obtuvo mejores resultados con los 48 valores de carga de un día anterior al pronosticado, con respecto a las demás fuentes bibliográficas todas utilizan este patrón de ingreso de datos hacia la red neuronal excepto (Manchado, 2018) que realiza una red neuronal LSTM donde indica que utiliza como variables de entrada la data histórica de la demanda de energía y la temperatura, para este proyecto de investigación solo se contaba con los registros históricos de la demanda de energía eléctrica por lo que se realizó el modelo LSTM con una sola variable pero de varios pasos y se obtuvieron los mejores resultados con 96 valores de entrada que corresponden a valores de la demanda de energía de dos días consecutivos y como salida de la red se obtiene 48 valores que corresponden a un día de predicción.



V. CONCLUSIONES

Se implemento un modelo de predicción basado en redes neuronales recurrentes LSTM que puede predecir la demanda de energía eléctrica con un 97.18% exactitud, a su vez se demostró con los resultados obtenidos del MAPE que el modelo puede realizar predicciones a corto y mediano plazo, también puede realizar predicciones a largo plazo lo cual se considera de 1 año a más pero como solo se tenía 8 meses de datos para realizar las pruebas no se pudo ver la eficiencia del modelo para predecir un año completo.

Se realizo el preprocesamiento de datos donde se pudo confirmar que un adecuado preprocesamiento de datos minimiza el error de predicción del modelo basado en redes neuronales recurrentes LSTM, lo cual se evidencia en la tabla número 11 donde se obtuvo una diferencia de 4.63% del MAPE por lo que la etapa del preprocesamiento de datos fue de vital importancia para obtener mejores resultados sobre el modelo LSTM.

Se realizaron varias arquitecturas LSTM, dentro de las cuales se experimentó variando el número de capas, neuronas y entradas hacia la red neuronal y como resultado se obtuvo que la arquitectura LSTM bidireccional mejora la predicción del modelo basado en RNR, lo cual se evidencia en la tabla número 9 esto es debido a que esta arquitectura aumenta efectivamente la información de la serie temporal hacia la red, mejorando así el contexto del algoritmo.



VI. RECOMENDACIONES

PRIMERO: Es recomendable evaluar otras técnicas de preprocesamiento de datos para poder obtener mejores resultados, especialmente sobre los datos atípicos.

SEGUNDO: Añadir la temperatura como variable, esto ayudaría a la RNN LSTM a establecer mejores relaciones entre el consumo de energía y la temperatura en una determinada hora

TERCERO: Incrementar el número de capas en la RNN LSTM Bidireccional puede ayudar a obtener mejores resultados, un inconveniente es que demora más el proceso de aprendizaje.

CUARTO: Es recomendable tener grandes cantidades de datos para poder suministrarlos a la RNR ya que es un punto clave para el proceso de aprendizaje.

QUINTO: Al momento de realizar el entrenamiento es mejor empezar con pocas neuronas y cantidad de épocas reducidas ya que con las RNR LSTM se pueden obtener buenos resultados rápido esto es de gran ayuda ya que en vez de preocuparse por la cantidad de neuronas, capas y épocas se puede tratar de enfocarse en cambiar las variables de entrada para obtener los mejores resultados he aquí que ya teniendo las variables de entrada definidas se puede estar intentado cambiar la cantidad de neuronas, capas y épocas.



VII. REFERENCIAS BIBLIOGRÁFICAS

Ariza, A. M. (2013). Métodos utilizados para el Pronóstico de la demanda de Energía Eléctrica en sistemas de Distribución. (*Tesis*). Universidad Tecnológica de Pereira, Pereira.

Brownlee , J. (s.f.). *machinelearningmastery*. Obtenido de machinelearningmastery:
<https://machinelearningmastery.com/>

Bruner. (2001). *El proceso mental en el aprendizaje*. Madrid: Narcea, S. A.

Chambers. (1979). *Diccionario Científico y Tecnológico* . Barcelona: Omega.

Cortina, J. (2012). *Aplicación de técnicas de inteligencia artificial a la predicción de contaminantes atmosféricos*. Madrid: McGraw-Hill.

Fernando, M. T. (2020). Modelo para el pronóstico de la demanda de energía eléctrica de EMELNORTE S.A - IBARRA aplicando redes neuronales artificiales. (*Tesis*). Universidad Técnica del Norte, Ecuador.

Larico Capia, J. E. (14 de 05 de 2022). *github.com*. Obtenido de github.com:
<https://github.com/Jhonatan-Larico/prediccion>

Manchado, M. C. (2018). Predicción de la demanda eléctrica española. Implementacion de de redes neuronales recurrentes en Python. (*Tesis*). Universidad Complutense de Madrid, Madrid.

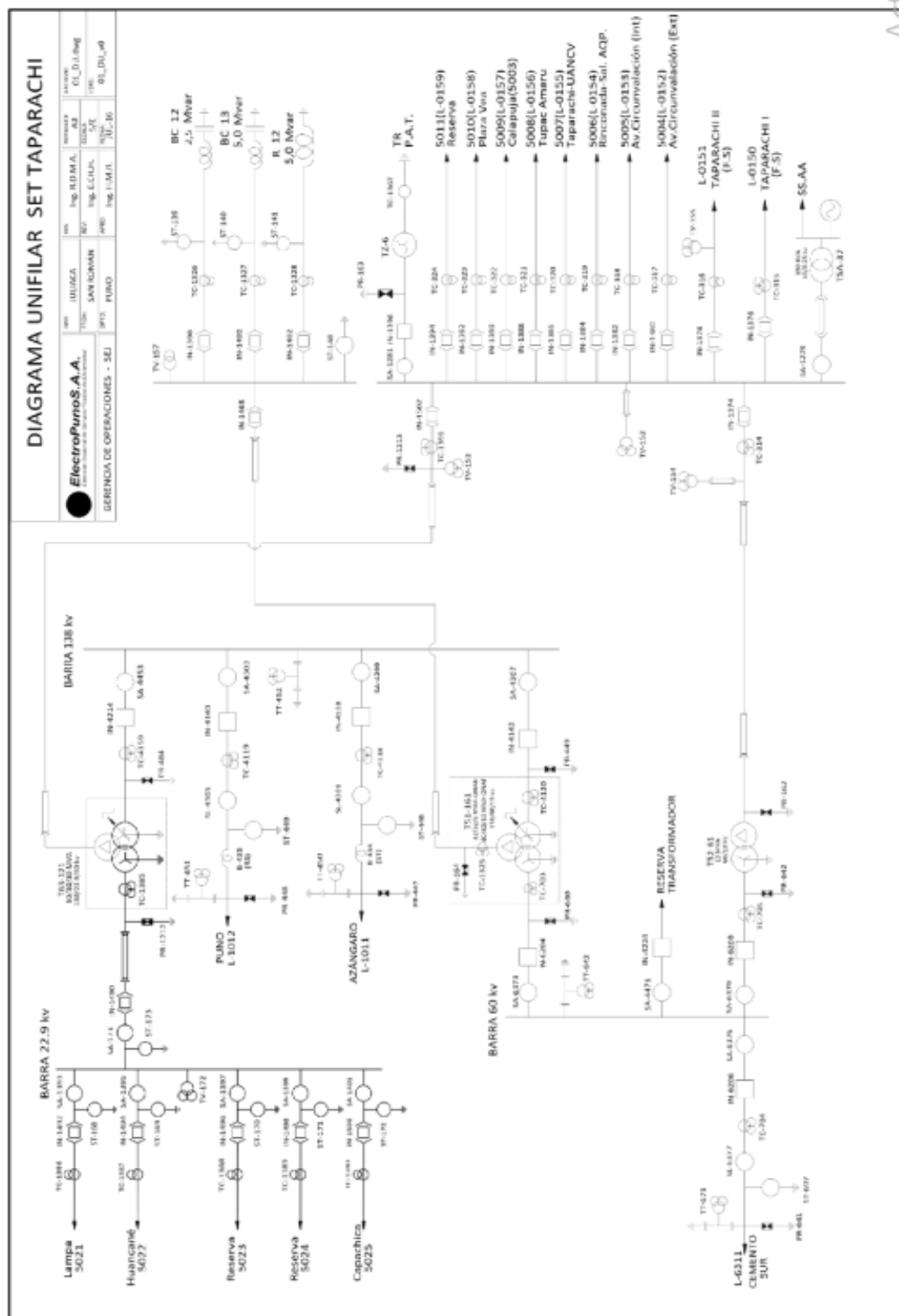
Moreno. (1994). *Aprendizaje automático*. Barcelona: Ediciones UPC.



- Morillo Tarapues, D. (2020). Modelo para el pronóstico de la demanda de energía eléctrica de EMELNORTE S.A - IBARRA aplicando redes neuronales artificiales. *Tesis*. Universidad Técnica del Norte, Ecuador.
- Orellana, R. J. (2012). *Modelación y Pronóstico de la Demanda de Energía Eléctrica de Mediano Plazo de El Salvador*. Maestría en Estadística. Ciudad Universitaria: Universidad de el Salvador.
- Quintela, & Redondo. (2005). *Redes Eléctricas*. España: Limusa.
- Quiñones, E. R. (2014). Prónostico de la demanda eéctrica a corto plazo utilizando redes neuronales artificiales aplicado al sistema interconectado nacional. (*Tesis de maestria*). Universidad Nacional de Ingenieria, Lima.
- Sampieri, R. H. (2014). *Metodología de la Investigación*. Mexico: McGRAW-HILL.
- Serrano, A., Soria, E., & Martín, J. (2009). *Redes neuronales artificiales*. Madrid: ANAGRAMA.
- Sinaluisa Lozano, I. F. (2017). Aplicacion de la red neuronal artificial FeedForward Backpropagation para la predicción de la demanda de energía eléctrica en la empresa eléctrica Riobamba S.A. (*Tesis*). Escuela Superior Politécnica de Chimborazo, Ecuador.
- Universidad Tenológica de Pereira. (2013). *Métodos utilizados para el pronóstico de demanda de energía eléctrica en sistemas de distribución*. Pereira: Universidad Tecnológica de Pereira.
- Villanueva Ure, R. (2010). *Centrales Hidroeléctricas*. Lima: Universidad Nacional de Ingeniería.

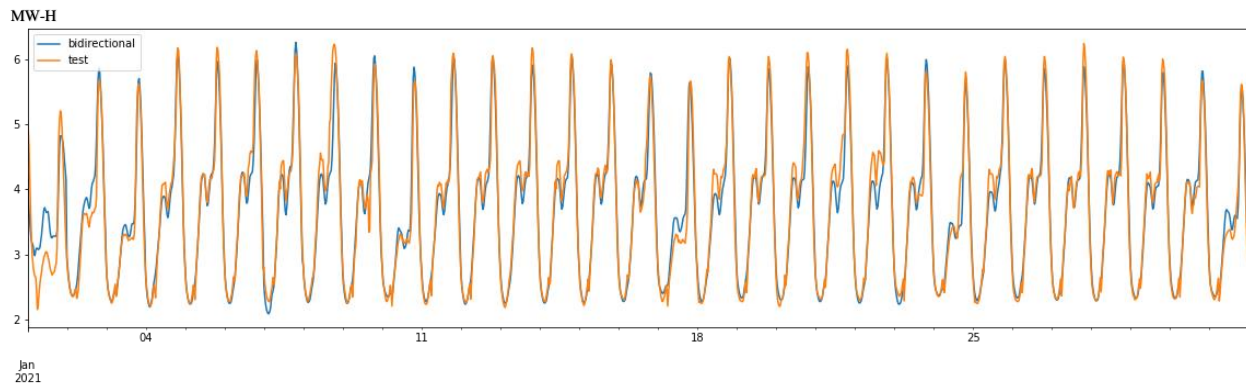
ANEXOS

ANEXO A. DIAGRAMA UNIFILAR DE LA SET TAPARACHI

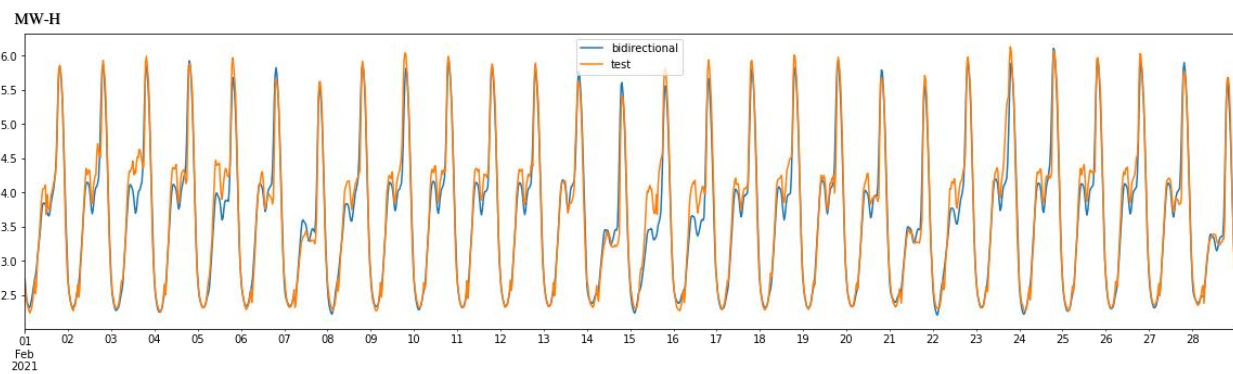


ANEXO B. RESULTADOS DE LAS PREDICCIONES MEDIANTE RNR LSTM DESDE ENERO HASTA AGOSTO DEL 2021

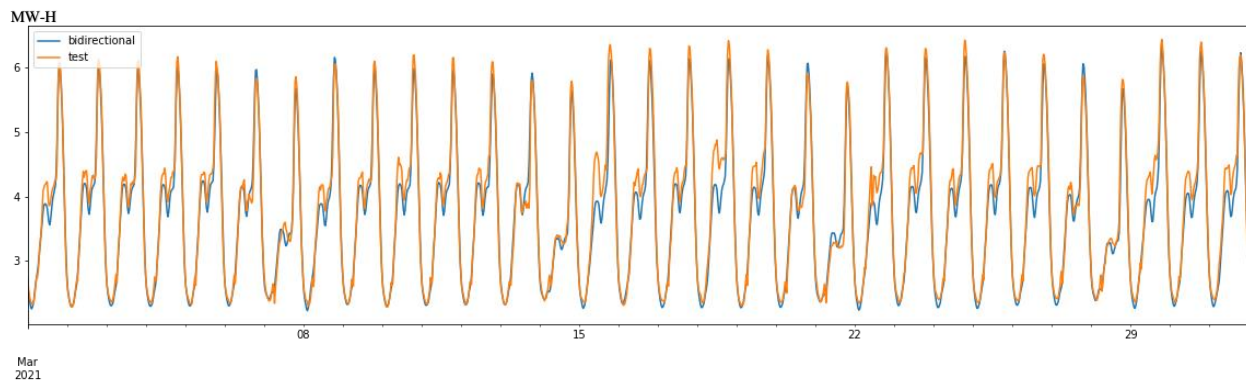
Enero 2021



Febrero 2021

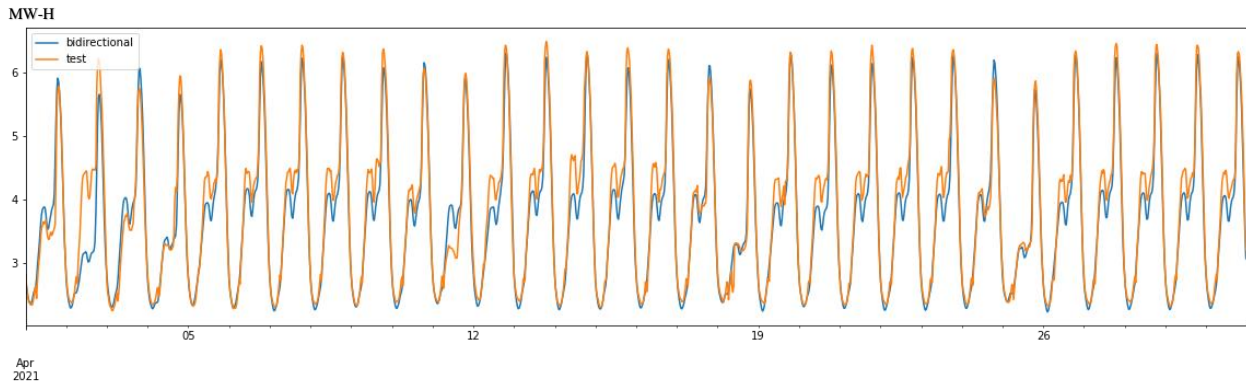


Marzo 2021

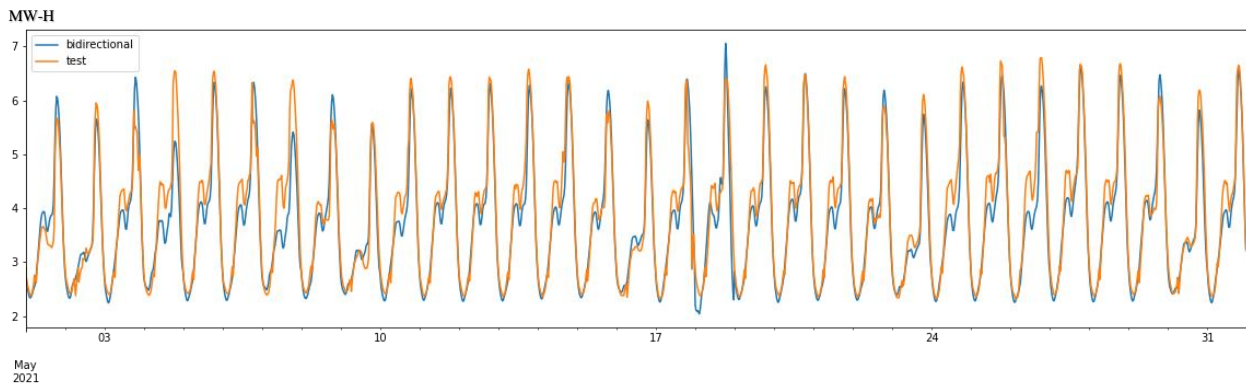




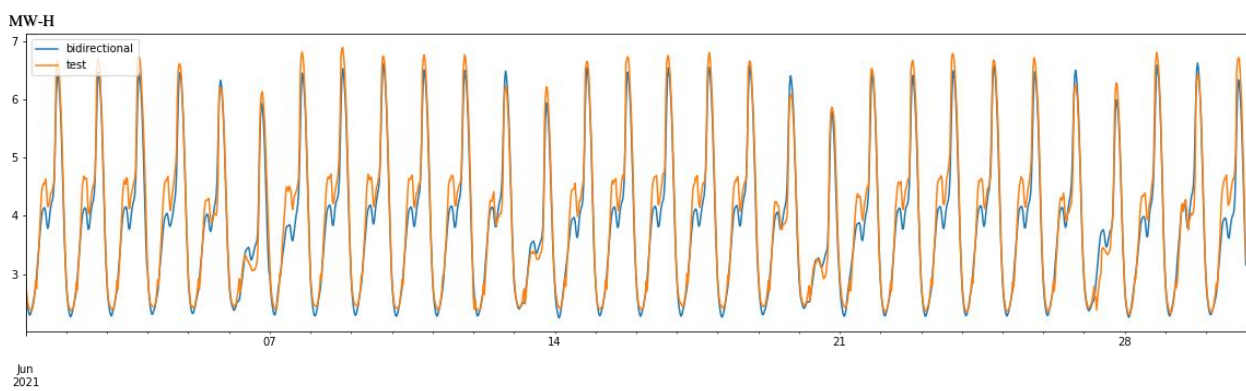
Abril 2021



Mayo 2021

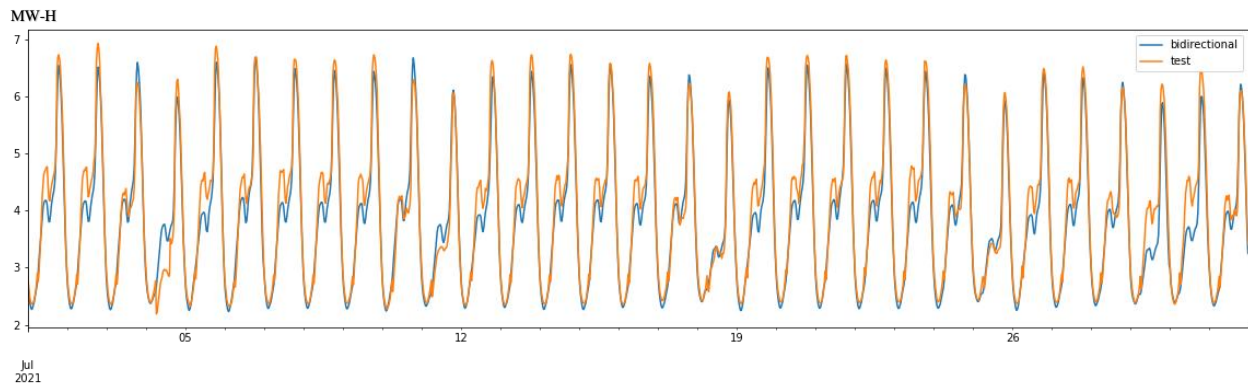


Junio 2021

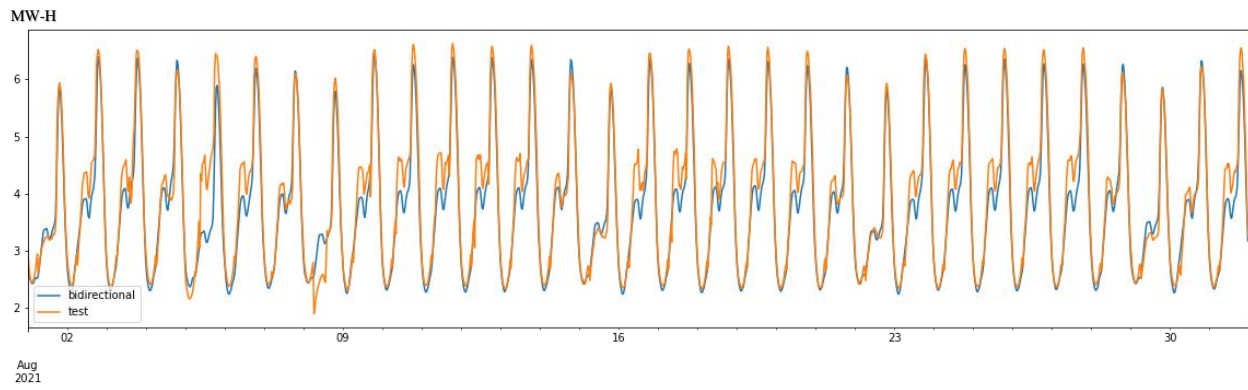




Julio 2021



Agosto 2021





ANEXO C: PROGRAMACIÓN DE LOS MODELOS RNN LSTM

LSTM Simple

```
# Entrenar el modelo LSTM simple
def build_model(train_x, train_y, n_input, tensorboard):

    # parámetros
    epochs, batch_size = 50, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1],
    train_x.shape[2], train_y.shape[1]
    # modelo
    model = Sequential()
    model.add(LSTM(100, activation='relu',
    input_shape=(n_timesteps, n_features)))
    model.add(Dropout(0.2))

    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # ajustar la red
    history=model.fit(train_x, train_y, epochs=epochs,
    batch_size=batch_size, callbacks=[tensorboard])
    return model, history
```

LSTM Apilado

```
# Entrenar el modelo LSTM apilado
def build_model_stacked_lstm(train_x, train_y, n_input, tensorboard):

    # parámetros
    epochs, batch_size = 50, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1],
    train_x.shape[2], train_y.shape[1]
    # modelo
    model = Sequential()
    model.add(LSTM(100, activation='relu', return_sequences=True
    , input_shape=(n_timesteps, n_features)))
    model.add(Dropout(0.2))
    model.add(LSTM(100, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # ajustar la red
    history=model.fit(train_x, train_y, epochs=epochs,
    batch_size=batch_size, callbacks=[tensorboard])
    return model, history
```

LSTM Bidireccional

```
# entrenar el modelo
def build_model_bd_lstm(train_x,train_y, n_input,tensorboard):

    # parámetros
    epochs, batch_size = 50, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1],
    train_x.shape[2], train_y.shape[1]
    # modelo
    model = Sequential()
    model.add(Bidirectional(LSTM(100, activation='relu'),
    input_shape=(n_timesteps, n_features)  ) )
    model.add(Dropout(0.2))
    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # ajustar la red
    history=model.fit(train_x, train_y, epochs=epochs,
    batch_size=batch_size, callbacks=[tensorboard])
    return model,history
```

ANEXO D: RESULTADO DEL MAPE DE TODOS LOS MODELOS

| | MAPE LSTM1 | LSTM 2 | LSTM Apilado | LSTM Bidireccional-1 | LSTM Bidireccional-2 | LSTM Bidireccional sin Preprocesamiento |
|------------------|------------|--------|--------------|----------------------|----------------------|---|
| Lunes | 5.13 | 11.55 | 5.07 | 5.14 | 2.58 | 13.29 |
| Martes | 6.36 | 4.71 | 5.31 | 4.91 | 2.14 | 5.18 |
| Miercoles | 5.84 | 4.59 | 5.26 | 4.55 | 2.93 | 5.17 |
| Jueves | 5.72 | 4.86 | 5.32 | 4.96 | 3.28 | 5.42 |
| Viernes | 6.82 | 5.64 | 5.83 | 5.83 | 3.73 | 6.67 |
| Sabado | 4.16 | 4.09 | 3.43 | 3.7 | 1.79 | 4.87 |
| Domingo | 4.89 | 11.29 | 3.92 | 4.47 | 3.33 | 11.44 |
| Total | 5.56 | 6.69 | 4.88 | 4.8 | 2.82 | 7.45 |