

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Eléctrica



Eficiencia energética usando técnicas de predicción e inteligencia artificial

Francisco Javier Rau Andrade

Profesor Guía: Ismael Soto Gómez

Tesis para optar al grado académico de Magíster
en Ciencias de la Ingeniería, mención Ingeniería
Eléctrica

Santiago – Chile
2022

RESUMEN

Este trabajo presenta el ahorro de energía en los centros de datos de un proveedor de servicios de telecomunicaciones. En la red *core* existen varios enlaces de fibra óptica que se encuentran agrupados por la necesidad de tener una mayor capacidad de ancho de banda. El objetivo principal de este trabajo es predecir el tráfico futuro y en base a esta predicción, apagar y encender los enlaces agrupados. Los resultados muestran que respecto al caso base (condiciones actuales), el ahorro de energía contemplado en un día del algoritmo basado en umbrales fue de un 6.8 % y la del algoritmo basado en predicción fue de un 12.2%. Cabe destacar que en esta simulación sólo se ahorra en un lado del equipo (lado cliente o servicios).

En cuanto a los algoritmos de predicción y de la metodología aplicada, la mejor red neuronal es OS-ELM. En términos de predicción, las redes neuronales OS-ELM, LSTM y GRU superan a RNN en 26 %, 2 % y un 1 %, respectivamente para la métrica RMSE. En un 25 %, 3 % y un 2 %, respectivamente para la métrica MAE, y en un 23 %, 6 % y un 8 %, respectivamente para la métrica MAPE.

En términos de tiempo computacional OS-ELM, LSTM y GRU superan a RNN en un factor de 217.6, 7.2 y 3.9 veces, respectivamente.

Palabras Claves:

Eficiencia energética; Inteligencia artificial; Predicción de tráfico; Operador de servicios de telecomunicaciones.

ABSTRACT

This thesis presents energy savings in the data centers of a telecommunications services operator. In the core network there are several fiber optic links that are in bundle due to the need for higher bandwidth capacity. The main objective of this work is to predict the future traffic and based on this prediction, turn off and turn on the bundle links. The results show that with respect to the base case (current conditions), the energy savings contemplated in one day for the algorithm based on thresholds was 6.8% and that of the algorithm based on prediction was 12.2%. It should be noted that this simulation only saves on one side of the equipment (client or service side).

In terms of prediction algorithms and methodology applied, the best neuronal network is OS-ELM. In terms of prediction, the neural networks OS-ELM, LSTM and GRU outperform RNN by 26%, 2% and 1%, respectively for the metric RMSE. In a 25%, 3%, and 2%, respectively for the metric MAE, 23%, 6%, and 8%, respectively for the metric MAPE.

In terms of computational time OS-ELM, LSTM and GRU outperform RNN by a factor of 217.6, 7.2 and 3.9 times, respectively.

Keywords:

Energy efficiency; Artificial intelligence; Traffic prediction; Telecommunications services operator.

AGRADECIMIENTOS

Primero que todo quisiera agradecer a mis 2 hijos, Rafaela y Bastián, dado que son mi motor y empuje a todo lo que me propongo y realizo día a día.

A la Universidad de Santiago de Chile, dado que es la Universidad que me forjó primeramente como profesional, y la que me recibe nuevamente después de 12 años para realizar el posgrado en Ingeniería Eléctrica. Son muchos los recuerdos que poseo dentro de ella, y que aún recuerdo al ingresar con 17 años de edad en mi primera carrera profesional.

A mis profesores del Magíster, que cumplen una vital función en la realización de este programa, para poder posicionar a la Universidad en los primeros lugares en investigación dentro de Chile.

A mi compañero de Magíster, Raul Zamorano, por su buena onda y simpatía dentro del Laboratorio del Centro de Investigación Multidisciplinario en Tecnologías de Telecomunicaciones (CIMTT).

Al profesor David Zabala de la Universidad Católica del Maule, por su dedicación, tiempo y ayuda en corrección de *papers* presentados en distintas conferencias.

Por último, a mi profesor guía Ismael Soto, dado que él ha sido el precursor del término de esta tesis por su constante motivación, la confianza dada y principales enseñanzas en el ámbito de investigación y realización de documentos científicos. Muchas gracias por todo.

“El día en que la ciencia comience a estudiar los fenómenos no físicos, progresará más en una década que en todos los siglos previos a su existencia”.

Nikola Tesla

TABLA DE CONTENIDO

Índice de Tablas	5
Índice de Figuras	6
1 Introducción	8
1.1 Origen y Necesidad del Tema	8
1.2 Descripción del Problema	10
1.3 Hipótesis	10
1.4 Objetivos	11
1.4.1 Objetivo general	11
1.4.2 Objetivos específicos	11
1.5 Desarrollo y Alcances	11
1.6 Aporte Personal	12
1.7 Estructura del documento	12
2 Revisión Bibliográfica	13
2.1 Métodos de predicción de trafico basado en redes neuronales	13
2.2 Eficiencia energética en enlaces agrupados	18
3 Metodología	20
3.1 Metodología en predicción de tráfico	20
3.1.1 Colección de datos	21
3.1.2 Estructura de redes neuronales RNN, LSTM, GRU y OS-ELM	21
3.1.3 Procesamiento de datos para redes neuronales RNN, LSTM, GRU y OS-ELM	27
3.1.4 Datos de entrenamiento y de testeo	28
3.1.5 Hiper-parámetros de redes neuronales RNN, LSTM, GRU y OS-ELM	30
3.1.6 Métricas	31
3.2 Metodología en eficiencia energética en Bundle Ethernet	33
3.2.1 Algoritmos basado en umbral y en predicción	34
3.2.2 Métricas	36
4 Escenario de simulación y Resultados	37
4.1 Escenario de simulación	37
4.1.1 Topología de red	37
4.1.2 Descripción del tráfico	41
4.1.3 Características del equipamiento	43

4.2	Resultados en predicción de tráfico	44
4.2.1	Resultados simulación red neuronal RNN, LSTM y GRU	44
4.2.2	Resultados simulación red neuronal OS-ELM	51
4.2.3	Elección final de la red neuronal	55
4.3	Resultados en algoritmos de eficiencia energética	57
5	Conclusiones	61
5.1	Conclusión General	61
5.2	Perspectivas Futuras	62
5.3	Productos de Investigación	63
5.3.1	Publicados (SCOPUS)	63
5.3.2	Bajo revisión (WOS)	63
	Acrónimos	64
	Referencias	67
	Anexos	75
A	Equipamiento NE40E-X8A	1
B	Tabulación de valores de simulación para RNN	2
C	Tabulación de valores de simulación para LSTM	6
D	Tabulación de valores de simulación para GRU	10
E	Tabulación de valores de simulación para OS-ELM	14

ÍNDICE DE TABLAS

Tabla 3.1	Especificaciones del sistema	32
Tabla 4.1	Potencia por tarjeta y puertos de <i>router</i> CDN.	43
Tabla 4.2	Configuración de Hiper-parámetros	45
Tabla 4.3	Resumen de resultados para RNN, LSTM y GRU	50
Tabla 4.4	Resumen de resultados para OS-ELM	54
Tabla 4.5	Comparación de predicción de tráfico de las redes neuronales	55
Tabla 4.6	Métricas de predicción de tráfico para las mejores redes de LSTM y OS-ELM	55
Tabla B.1	First group - RNN results	3
Tabla B.2	Second group - RNN results	4
Tabla B.3	Third group - RNN results	5
Tabla C.1	First group - LSTM results	7
Tabla C.2	Second group - LSTM results	8
Tabla C.3	Third group - LSTM results	9
Tabla D.1	First group - GRU results	11
Tabla D.2	Second group - GRU results	12
Tabla D.3	Third group - GRU results	13
Tabla E.1	First group - OS-ELM results	15
Tabla E.2	First group - OS-ELM results	16
Tabla E.3	Second group - OS-ELM results	17
Tabla E.4	Second group - OS-ELM results	18
Tabla E.5	Third group - OS-ELM results	19
Tabla E.6	Third group - OS-ELM results	20

ÍNDICE DE FIGURAS

Figura 1.1. Agrupación de enlaces en un TSO.	10
Figura 3.1. Metodología de predicción de tráfico.	20
Figura 3.2. Estructura de la célula RNN	21
Figura 3.3. Estructura de la célula LSTM	22
Figura 3.4. Estructura de la célula GRU	24
Figura 3.5. Capas de ELM, Extreme learning machine	25
Figura 3.6. Conjunto de datos de entrenamiento y testeo.	29
Figura 3.7. Timesteps (lags) en redes neuronales recurrentes.	30
Figura 3.8. Metodología en eficiencia energética en Bundle Ethernet.	33
Figura 3.9. Algoritmo basado en umbrales.	34
Figura 3.10. Algoritmo basado en predicción	35
Figura 4.1. Agrupación de enlaces en un TSO.	38
Figura 4.2. Arquitectura de red CDN en un TSO.	39
Figura 4.3. Lado del <i>router</i> a ser considerado.	40
Figura 4.4. Tráfico CDN - Conjunto de datos final.	41
Figura 4.5. Métrica RMSE para redes RNN, LSTM y GRU.	45
Figura 4.6. Métrica MAE para redes RNN, LSTM y GRU.	47
Figura 4.7. Métrica MAPE para redes RNN, LSTM y GRU.	48
Figura 4.8. Métrica tiempo computacional o <i>lapsed time</i> para redes RNN, LSTM y GRU.	49
Figura 4.9. Métrica RMSE para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00	51
Figura 4.10. Métrica MAE para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00	52
Figura 4.11. Métrica MAPE para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00	53
Figura 4.12. Métrica tiempo computacional o <i>lapsed time</i> para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00	54
Figura 4.13. Comparación de predicción de tráfico de las mejores redes: LSTM y OS-ELM	56
Figura 4.14. Ahorro acumulado en un día para caso base, umbral y predicción	57
Figura 4.15. Número de puertos activos en un día para caso base, umbral y predicción	58
Figura 4.16. Capacidad disponible en el BE o LAG en un día para caso base, umbral y predicción	59

Figura 4.17. Pérdida de capacidad en el BE o LAG en un día para caso base, umbral y predicción 60

Figura A.1. Consumo de *router* CDN NE40E-X8A. 1

CAPÍTULO 1

INTRODUCCIÓN

1.1 ORIGEN Y NECESIDAD DEL TEMA

La creciente demanda de energía en la industria moderna, el calentamiento global del planeta Tierra y el desarrollo de nuevas tecnologías de comunicación, como el Internet de las Cosas (IOT) y el 5G, han hecho que en los últimos años se investigue cómo ahorrar energía en el sector de las telecomunicaciones, principalmente en los operadores de servicios de telecomunicaciones (TSO).

En [1] se indica que la demanda mundial de energía de los centros de datos creció de 194 TWh a 205 TWh durante 2010 - 2018. En [2] las redes de telecomunicaciones consumirán hasta el 51 % de la electricidad mundial en el peor de los casos para el año 2030, si no se mejora suficientemente su eficiencia energética. Por lo tanto, la eficiencia energética se está convirtiendo en algo crítico para las redes actuales.

Algunos TSO han diseñado redes con enlaces redundantes para evitar la congestión en esquemas de alta disponibilidad (configuración activa/pasiva) y de equilibrio de carga (configuración activa/activa) [3]. Desde el punto de vista de la eficiencia energética, estos diseños pueden considerarse como un gasto energético ineficiente, debido a que estos enlaces están siempre activos [4]. Los estudios [5], [6] muestran que los enlaces están infrutilizados sólo en un 40 % en la red central o comúnmente llamada *Core*, y que durante las horas valle es mucho menor. Además, no hay mucha diferencia en el consumo de energía entre los equipos a plena carga y en modo de espera [5].

Para aumentar la capacidad de ancho de banda, los TSOs conectan los *routers* a través de múltiples cables físicos que forman un único enlace lógico agrupado. Los enlaces lógicos

agrupados también se denominan grupos de agregación de enlaces (LAG) o Bundle-Ethernet (BE). La agregación de enlaces en Ethernet se estandarizó en la norma IEEE 802.3ad y posteriormente se renombró como IEEE 802.1ax para mantener la coherencia con otras normas 802.3 [7].

La actual red *Core* de una TSO se compone de múltiples redes y las líneas troncales suelen estar formadas por muchas fibras ópticas agrupadas para aumentar la capacidad y añadir resiliencia. Las conexiones con el número de sub-enlaces dentro de un BE o LAG van de 2 a aproximadamente 20 en un TSO típico [6]. Además, existe el problema del consumo energético de estos enlaces y más aún cuando están agrupados. En los últimos años se han publicado y propuesto varios trabajos para reducir el consumo energético de los enlaces simples (no agrupados) [8]. Uno de los estándares que ayuda a resolver este problema es el IEEE 802.3az, que especifica la Eficiencia Energética de Ethernet (EEE) que es un método para reducir la energía utilizada por un dispositivo Ethernet durante los períodos de baja utilización de enlace [9]. La premisa de EEE es que los enlaces Ethernet tienen tiempo de inactividad y, por tanto, la oportunidad de ahorrar energía en dicho periodo de tiempo. El método se denomina Low Power Idle (LPI), pero es sólo para interfaces de cobre [10].

En cuanto a los LAGs, se han realizado varios estudios de eficiencia energética, utilizando diferentes métodos de optimización [6], Además, hay trabajos que proponen algoritmos que colocan puntos de umbral como en [11], y otros [12] que trabajan con controladores de redes definidas por software (SDN). También en [13] ha utilizado técnicas predictivas para reducir la energía de los LAGs, utilizando una Media Móvil Simple (SMA).

Por otra parte, las redes neuronales han comenzado a ser utilizadas para predecir series de tiempo, debido a que las series de tiempo son el método óptimo para describir el comportamiento del tráfico de la red. En lo que respecta principalmente a las redes neuronales recurrentes (Recurrent Neural Network, RNN) y sus variantes, la red neuronal de memoria a largo plazo (Long Short-Term Memory, LSTM) y la red neuronal de unidades recurrentes cerradas (Gated Recurrent Unit, GRU) en [14–17]. Existe también una red neuronal que se destaca por su rapidez llamada máquina de aprendizaje extremo secuencial en línea (Online Sequential Extreme Learning Machine, OS-ELM), que ha mostrado ser eficiente y sobretodo sumamente rápida en comparación con las redes basadas en optimizar el gradiente [18–20]. En los trabajos asociados a eficiencia energética, una de las problemáticas que no se ha abordado es la de realizar un ahorro de energía en LAGs o BE con la predicción del tráfico utilizando redes neuronales recurrentes y/o secuenciales.

La principal motivación de este trabajo de tesis es la predicción de tráfico en un corto periodo de tiempo para activar y desactivar los puertos de la agregación de enlaces (LAG o BE) entre 2 nodos dentro de una red TSO, con el fin de ahorrar energía en los enlaces de fibra óptica.

Si los resultados son satisfactorios podrían extrapolarse a toda la red en *routers* que conlleven agregación de enlaces.

1.2 DESCRIPCIÓN DEL PROBLEMA

Los operadores de servicios de telecomunicaciones (TSO) por la creciente demanda de datos, tanto fijo y móvil, deben aumentar la capacidad de los enlaces dentro del *Core*. Este aumento de capacidad se realiza a través de aumento de tarjetas de línea, y por ende conexión de enlaces entre los equipos de red, *routers* principalmente. Como se dijo anteriormente, la agrupación lógica de varios enlaces físicos Ethernet, se denomina LAG por siglas al inglés, *Link Aggregation Group* o *BE*, *Bundle Ethernet*, esta agrupación se trata como un único enlace lógico y permite la suma de la velocidad nominal de cada puerto físico Ethernet utilizado para obtener un enlace troncal de alta velocidad. Como se muestra en la Fig. 1.1, se puede ver que Bundle-Ethernet esta compuesto de n interfaces de fibra óptica generando un LAG de n enlaces.

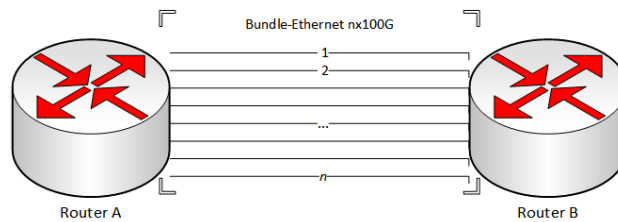


Figura 1.1. Agrupación de enlaces en un TSO.

Desde el punto de vista de eficiencia energética, la agrupación de enlaces no es óptima debido a la energía consumida de cada puerto activo. Aunque los investigadores han propuesto muchos esquemas para mejorar la eficiencia energética de los BEs o LAGs, todavía hay algunos problemas con la escalabilidad de la solución, debido a que están basados en un momento de la red estudiada, sin tener en cuenta la expansibilidad y la forma dinámica de la red que posee un operador de servicios de telecomunicaciones.

La solución que se propone en el presente trabajo de grado, es implementar una herramienta que sea eficiente desde el punto de vista energético cuando se dispone de LAGs en la red, pero en tiempo real y dinámica, y que en función de la predicción de la carga de tráfico, optimizar la cantidad de enlaces a utilizar.

1.3 HIPÓTESIS

Es posible aumentar la eficiencia energética por el uso de técnicas de predicción basada en inteligencia artificial.

1.4 OBJETIVOS

1.4.1 Objetivo general

Proponer y simular una herramienta en tiempo real mediante inteligencia artificial para el ahorro de energía en los enlaces ópticos Ethernet agrupados para ser aplicada en las redes de las empresas proveedores de servicios.

1.4.2 Objetivos específicos

- Presentar el estado del arte en herramientas de predicción de tráfico basada en inteligencia artificial y herramientas de eficiencia energética asociados a enlaces agrupados (LAG).
- Identificar técnicas de inteligencia artificial que permitan predecir tráfico de red en tiempo real.
- Evaluar técnicas de inteligencia artificial utilizando datos en tiempo real de un operador de servicios, en términos de tiempo y rendimiento con el fin de conocer la mejor herramienta para predecir el tráfico.
- Diseñar algoritmos para habilitar o deshabilitar puertos en los enlaces ópticos Ethernet agrupados en función del tráfico real y predicho.
- Evaluar los algoritmos diseñados en cuanto a su eficiencia energética simulándolos a una situación real de un proveedor de servicios.

1.5 DESARROLLO Y ALCANCES

Los datos serán colectados por una plataforma via Simple Network Management Protocol (SNMP) de manera *online*, la cual se generará una base de datos. Se debe desarrollar algoritmos que permita encender y apagar los puertos a un lado del *router*, según el tráfico que se predice en un *slot* de tiempo. Para que se pueda llevar a cabo, se debe determinar cuál es la técnica que mejor reúne las condiciones para realizar una correcta predicción con errores mínimos. Después de ello, las técnicas se evaluarán con 3 métricas las cuales son: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) y Mean Absolute Percentage Error (MAPE), variando los hiperparámetros de cada red neuronal a evaluar.

Ya elegida la red neuronal a utilizar, esta será de entrada al desarrollo del algoritmo con predicción de tráfico. Además, se realizarán comparaciones con el caso base actual y con un

algoritmo que se implementará con umbrales o *thresholds* basados en los datos anteriores (brutos) sin predicción.

Una vez ya teniendo los algoritmos de eficiencia energética, se desarrollará una comparativa, utilizando la métrica *W/h* día, simulándolos a una situación real en una empresa proveedora de servicios.

1.6 APORTE PERSONAL

El aporte personal en este trabajo es proponer e implementar un algoritmo capaz de habilitar o deshabilitar puertos en un LAG o BE en función del tráfico pronosticado, es decir, proponer una herramienta completa para el ahorro de energía en los LAGs para ser aplicada en las redes de las empresas proveedores de servicios.

Además, en paralelo a la elaboración de este trabajo, se han generado publicaciones, las cuales se encuentran indexadas en Scopus. Los trabajos realizados corresponden a “*Forescating Mobile Network Traffic based on Deep Learning Networks*” [15], realizado en Noviembre de 2021, en el cual se presenta a la comunidad científica tres redes neuronales profundas, el cual predicen con gran exactitud las ráfagas de tráfico producida por la tecnología 5G, medida en el nodo Evolved Packet Core (EPC). “*Network Traffic Prediction Using Online-Sequential Extreme Learning Machine*” [19], realizado en Noviembre 2021, el estudio se centra en la previsión con técnicas de redes neuronales, concretamente Long Short-Term Memory (LSTM) y Online-Sequential Extreme Learning Machine (OS-ELM), del tráfico real de un TSO de Chile. Los resultados muestran que OS-ELM supera a LSTM en términos de costo computacional debido a su simplicidad en su estructura en perjuicio de la predicción de la red.

1.7 ESTRUCTURA DEL DOCUMENTO

En el capítulo “estado del arte” se detallan los trabajos sobre ahorro de energía en LAG o BE y los tipos de redes neuronales recurrentes y secuenciales más utilizadas en predicción de tráfico. Luego, en el capítulo “metodología”, se presentan dos metodologías, la primera referida a las etapas a desarrollar para la elección de una red neuronal y la segunda las etapas y desarrollo de dos algoritmos de eficiencia energética para una red *core*. En el capítulo “escenario de simulación y resultados”, se presenta la topología de red, el escenario de simulación y el detalle de equipamiento de red. Además, se realizan los entrenamientos de las redes neuronales con datos de entrenamiento y testeo. Se muestran los resultados con las distintas métricas de evaluación. Por último, se presentan las “conclusiones” en el último capítulo.

CAPÍTULO 2

REVISIÓN BIBLIOGRÁFICA

En el presente capítulo se realiza una revisión bibliográfica de los principales puntos que abarca este trabajo. Se revisará información relacionada a la predicción tráfico basados en redes neuronales y eficiencia energética en enlaces agrupados (LAG) o (BE).

2.1 MÉTODOS DE PREDICCIÓN DE TRAFICO BASADO EN REDES NEURONALES

Analizar el tráfico histórico es un reto crítico para generar un modelo preciso que refleje la estructura de la serie, con el fin de permitir la predicción, y clasificación de eventos futuros [21], [22]. Las series temporales consisten en una sucesión de puntos numéricos de datos ordenados. El problema de la predicción de las series temporales es la previsión de la actividad futura a partir de los valores pasados y los patrones relacionados que estas presentan [22], [23].

Para la predicción de tráfico pueden utilizarse diversas técnicas de previsión de series temporales, agrupadas en dos tipos de métodos lineales y no lineales. Los métodos no lineales son más adecuados para predecir el tráfico, debido al ruido existente y a la naturaleza compleja que presenta el propio tráfico [24]. Las redes neuronales están siendo usadas ampliamente para predecir series temporales [25].

En el estudio [26] se indica que las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN) son los modelos de aprendizaje profundo o *deep learning* más utilizados en la predicción del tráfico a corto plazo. Las redes CNN son buenas para capturar las características espaciales y las redes RNN son buenas para capturar las características temporales de los datos de tráfico. Dentro de las redes de *deep learning* y recurrentes, existen

variaciones de ellas como la red neuronal llamada memoria prolongada de corto plazo (LSTM) y unidades recurrentes cerradas (GRU). En la actualidad, predecir tráfico con redes neuronales ha hecho del análisis de series temporales una parte esencial del modelado de datos en una amplia gama de industrias, incluyendo finanzas, salud, transporte, medio ambiente [27–32].

En [23] han utilizado la red neuronal de unidades recurrentes cerradas (GRU) variante de LSTM para predecir el flujo de tráfico de barcos dentro de un área de parques eólicos. Los resultados de GRU fueron comparados con el modelo autorregresivo integrado de media móvil (ARIMA), y de *machine leaning SVM* y *deep learning LSTM*, siendo GRU el ganador.

En el sector de logística, se ha utilizado la predicción del flujo de pasajeros en estaciones de metro [33], uso de modelado de estacionamientos en [34] y planificación de la urbanización según predicción de tráfico de vehículos [35].

En el ámbito de la salud, predicción del estrés y somnolencia del conductor para la prevención de accidentes [36] y predicción de la progresión mensual de la enfermedad de alzheimer [37].

Con respecto al sector de Telecomunicaciones, se ha utilizado la predicción en carga de tráfico de la red de TSO para evitar la sobrecargas, minimizar el tiempo de respuesta y optimizar el uso de los recursos [15], [19], [38, 39].

Con la llegada del estándar 5G, será mas difícil la predicción de tráfico por ser una red heterogénea. La coexistencia de diferentes redes y características significativamente diferentes hacen que la predicción, gestión y optimización del tráfico de red sea una tarea difícil. Por esta razón, cada vez más serán mayormente utilizadas el uso y adaptación de las tecnologías de inteligencia artificial (mayoritariamente redes neuronales) para analizar y administrar el tráfico de las redes a partir de los datos [40]. El variado uso de modelos de inteligencia artificial en los operadores de servicios de telecomunicaciones e investigaciones de redes neuronales han hecho que cada vez existen mayores comparaciones entre ellas.

En [41] hacen uso de predicción de trafico en la red celular para la optimización dinámica de recursos en redes *backhaul* inalámbricas.

En [14] autores indican que la red neuronal LSTM es un tipo de arquitectura de red neuronal recurrente RNN, que se entrena con un algoritmo de aprendizaje basado en el gradiente. El marco propuesto por los autores hace uso de trazas reales de un TIER-1 TSO. Con estas trazas realizaron predicciones con distintos *span* de tiempo comparándolas con una combinación del método clásico como ARIMA más la red neuronal recurrente RNN, los resultados obtenidos en comparación con el modelo ARIMA y RNN, el modelo LSTM funciona bien con un RMSE normalizado bajo para todo el conjunto de datos, y además LSTM genera predicciones a escalas de tiempo muy cortas (menos de treinta segundos).

En el trabajo [42], se compararon RNN, LSTM, GRU, Deep Neural Network (DNN) y Bidirectional LSTM (BLSTM), y el resultado fue que LSTM es una de las redes que mejor se comportó en términos de predicción de tráfico, debido a que es el que posee métricas de error más bajas.

Los autores en [43] muestran la predicción de tráfico de Internet en Telecom Italia, se comparan las prestaciones de LSTM, GRU y de dos arquitecturas de aprendizaje automático convencionales, Random Forest (RF) y Decision Tree (DT) para la predicción del tráfico de Internet móvil. La calidad predictiva de los modelos se evaluó mediante el error medio cuadrático (RMSE) y el error medio absoluto (MAE). Ambos algoritmos de aprendizaje profundo fueron eficaces para modelar la actividad de Internet y la estacionalidad, tanto dentro de los días como a lo largo de 2 meses. Los modelos de aprendizaje profundo superaron a los modelos de aprendizaje automático convencionales, colocando a la red LSTM como ganadora frente a la red neuronal GRU en los experimentos.

En el estudio [15] se muestra que las redes 5G pueden enfrentarse a *peaks* de tráfico de red, debido a las numerosas conexiones, por lo que se centra en la predicción de los *peaks* de tráfico a través de técnicas de aprendizaje profundo como RNN, LSTM y GRU de una red real. En términos de predicción, LSTM y GRU superan a RNN, en un 4,98 % y 4,56 %, respectivamente y en términos de complejidad computacional, GRU es el peor comparado con RNN y LSTM, en un 12,16 % y 0,13 %, respectivamente. Por último, dado que los tiempos entre GRU y LSTM son similares, buscándose mayor precisión para los *peaks* de tráfico, el modelo LSTM fue el de mejor rendimiento.

En [44] se comparó el rendimiento de predicción de modelos recurrentes RNN, LSTM y GRU, con modelos no recurrentes como XGBoost y RF, regresión lineal LASSO y modelos de predicción basado en medias móviles. Los resultados indicaron que RNN y sus variantes superaron a los demás métodos. El mejor modelo no recurrente fue XGBoost. En comparación a XGBoost, GRU y RNN redujo en un 15 % en la métrica RMSE y en un 8 % en MAPE.

Otro estudio en [45], las redes neuronales recurrentes y sus variantes se comparan con métodos estadísticos tradicionales, como ARIMA, ARIMA estacional (SARIMA) y la red neuronal CNN. Se prueban tres escenarios diferentes de congestión, día completo, hora *peak* en la mañana y hora *peak* en la tarde. Los resultados indican para la métrica RMSE que la red neuronal LSTM tiene rendimientos de 5,8, 7,9 y 10,2 y LSTM de 6,7, 8,6 y 10,9 según el escenario probado.

Como se ha descrito, las redes neuronales RNN, LSTM y GRU han ganado terreno en predicción de tráfico, frente a modelos de machine learning (XGBoost, RF, SVM, DT), convencionales como regresión lineal LASSO, ARIMA, SARIMA y otras redes neuronales CNN.

Uno de los factores que impiden que las redes de aprendizaje profundo RNN, LSTM y GRU

sean más rápidas en predecir, es el factor de buscar su optimización basado en la minimización del gradiente, por lo que se ha buscado encontrar una alternativa a esta, no perdiendo su *performance* en rendimiento predictivo.

Una de las redes neuronales que ha concentrado el interés en este último tiempo, son las redes neuronales de aprendizaje extremo (ELM, por sus siglas en inglés), debido a sus buenos resultados en predicción y su algoritmo de entrenamiento extremadamente rápido [46].

ELM es un tipo particular de red neuronal *feed-forward*. El mecanismo de aprendizaje permite ofrecer una velocidad de entrenamiento significativamente mayor en comparación con las redes neuronales clásicas en una variedad de escenarios.

Una variación de esta red neuronal es la llamada máquina de aprendizaje extremo secuencial en línea, en inglés Online-Sequential Extreme Learning Machine (OS-ELM). La ventaja de OS-ELM frente ELM es que permite al algoritmo aprender en línea datos secuenciales, parte por parte utilizando el método de mínimos cuadrados recursivos [47]. Autores muestran en estudio [19], la comparación de la red neuronal recurrente LSTM con la red neuronal OS-ELM. Los resultados muestran que OS-ELM supera a LSTM en términos de costo computacional por un factor de 2300, lo que es altísimo, y en términos de predicción de red, OS-ELM fue similar a LSTM. La red neuronal OS-ELM es más simple en tipo de arquitectura que LSTM, lo que hace que en términos de complejidad sean más rápidas que las recurrentes RNN y sus variantes. Estas últimas redes neuronales pueden tener en ciertos casos mejor precisión, pero dependerá de los datos de entrada, la arquitectura de la serie temporal y los hiper-parámetros que se le se configuren.

Para los modelos de predicción, el objetivo de las métricas de evaluación es minimizar el error. Para los modelos de regresión y predicción, las métricas error absoluto medio (MAE) y error de raíz cuadrada media (RMSE) se utilizan con mayor frecuencia para evaluar el rendimiento del modelo. Como ejemplo de algunos estudios en [15], [19], [38], [48–51] se ha trabajado con la métrica RMSE y en otros [37], [48], [38], [52] con la métrica MAE. El error porcentual absoluto medio (MAPE), también se utiliza en algunos casos, para evaluar la regresión y/o predicción [53], [54].

RMSE y MAE son métricas dependientes de la escala. MAPE es una métrica de error porcentual. La métrica RMSE es muy sensible a los valores atípicos, mientras que el impacto de éstos se reduce con MAE. Por otro lado MAE no puede indicar el sesgo de las predicciones en términos de sobreajuste (*overfitting*) o subajuste (*underfitting*). MAPE puede diferenciar este tipo de sesgo e impone una penalización a los valores predichos por encima de los reales.

En el presente estudio, se necesita una predicción lo más exacta posible, relajando el tiempo de procesamiento como máximo al tiempo de obtención de los datos del sistema. Como esto depende de los datos de la serie temporal, en este trabajo se hará una comparación de

OS-ELM frente a las redes neuronales recurrentes y sus variantes (RNN, LSTM, GRU) para comprobar cuál de estas redes es mejor en términos de precisión (RMSE, MAE y MAPE) y tiempo computacional.

2.2 EFICIENCIA ENERGÉTICA EN ENLACES AGRUPADOS

Para los trabajos de eficiencia energética en enlaces agrupados, se han propuesto diversos métodos y técnicas para reducir el consumo de energía de los LAGs o BE en las redes de telecomunicaciones, que se pueden dividir en cuatro categorías: métodos de optimización, puntos de umbral, controladores SDN y técnicas predictivas.

En [6] se propuso una formulación de Programación Lineal Entera (ILP) para optimizar la energía de los LAGs. El ahorro de energía fue significativo, reduciendo el consumo de energía en un 79% respecto a una red normal. Otro trabajo en [55] estudió la reducción de energía en los enlaces agrupados, utilizando un modelo de formulación de Problema Lineal Íntegro Mixto (MILP), mostrando una reducción del 50% con respecto al enrutamiento del camino más corto. En [56] se estudió el uso de la activación y desactivación de enlaces agregados con el método de optimización, utilizando la Programación Lineal (LP) y se mostró cómo encontrar la configuración de enlaces más eficiente en función de su consumo de energía, para todos los enlaces de un BE. El ahorro de energía osciló entre el 10% y el 30% dependiendo del tipo de configuración elegida. En [57] se utilizó la optimización convexa con el algoritmo de llenado de agua reduciendo el consumo de energía hasta un 50% en enlaces ethernet eficientes. *Energy Efficient Ethernet* (EEE), un estándar 802.3az del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), que reduce el consumo de energía de los dispositivos de la capa física (PHY) durante los periodos de baja utilización del enlace. EEE ahorra energía conmutando parte del circuito de transmisión en modo de bajo consumo, cuando el enlace está inactivo.

Un enlace Ethernet consume energía incluso cuando el enlace está inactivo. La EEE proporciona un método para utilizar la energía de forma que los enlaces Ethernet sólo la utilicen durante la transmisión de datos. EEE utiliza un protocolo de señalización, *Low Power Idle* (LPI), para lograr el ahorro de energía cuando un enlace Ethernet está inactivo. EEE permite a los PHYs intercambiar indicaciones LPI para señalar la transición al modo de bajo consumo cuando no hay tráfico. LPI indica cuándo un enlace puede quedar inactivo y cuándo debe reanudarse después de un retardo predefinido sin afectar a la transmisión de datos. Los siguientes PHY de cobre están estandarizados por IEEE 802.3az: 100BASE-T, 1000BASE-T y 10GBASE-T. Pero no así los estándares de fibra óptica.

Otros algoritmos en [11] como el umbral heurístico local fijo (FLHT) y el umbral heurístico local dinámico (DLHT), los cuales son dos algoritmos distribuidos optimizados localmente, pueden ajustar dinámicamente el número de subenlaces activos, para ahorrar el consumo de energía en aproximadamente un 80% en la red *Core* tanto para los casos de *bin packing* como de balanceo de carga.

En las redes de telecomunicaciones de un TSO que poseen una mayor inteligencia

ocupando controladores SDN, se aplica el algoritmo de puerto de espera (SPA) y el algoritmo de dos colas (TQA), logrando una eficiencia de consumo de energía de los enlaces agrupados de hasta el 50 % [12], [58].

En [13] el algoritmo propuesto por el autor fue capaz de reducir el número medio de enlaces activos hasta el 25.4%, utilizando un mecanismo basado en la técnica de predicción llamada Media Móvil Simple (SMA).

Aunque los investigadores han propuesto muchos esquemas para mejorar la eficiencia energética de los BEs o LAGs, aun hay problemas con la escalabilidad de la solución, porque se basan principalmente en un momento de la red, sin tener en cuenta la expansibilidad y la forma dinámica de una red TSO y la gran mayoría de ellos asociados a estándares de cobre.

CAPÍTULO 3

METODOLOGÍA

3.1 METODOLOGÍA EN PREDICCIÓN DE TRÁFICO

Para entender la metodología en predicción de tráfico, la Fig. 3.1 muestra los pasos para obtener la mejor predicción en función de la configuración y del modelo a utilizar. El diagrama consta de una entrada, en donde se alimentan con datos duros (*data raw*), obtenidos de una plataforma que monitorea en línea. Después de ello, se procesa los datos (limpieza y realización de ajustes para poder realizar las simulaciones). Posteriormente, se realiza un división de los datos, correspondiente a procesos de validación. Como siguiente paso, se realizan las simulaciones aplicando cada red neuronal, verificando su rendimiento con métricas, como la medición del *performance* de la predicción, en términos de error RMSE, MAE, MAPE y en tiempos computacionales. Finalmente se da la elección al mejor modelo de predicción.

A continuación se mostrará con detalle cada uno de estos pasos.

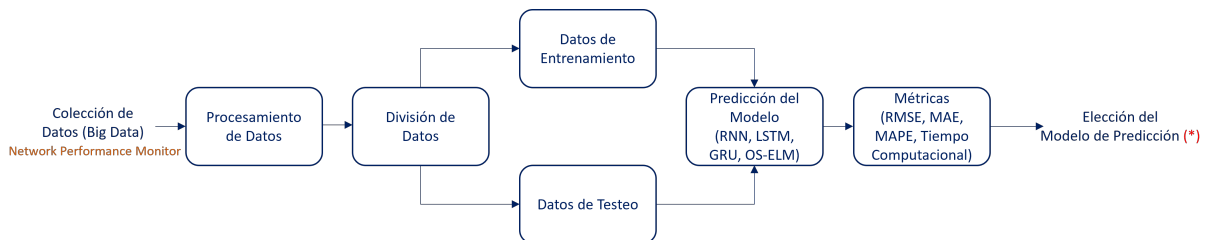


Figura 3.1. Metodología de predicción de tráfico.

3.1.1 Colección de datos

El tráfico de los *routers* de la red TSO es monitoreada por la plataforma *Network Performance Monitor* a través del protocolo simple de administración de red (SNMP), que proporciona estadísticas de tráfico en tiempo real, con *span* en minutos, horas, días y meses. Esta plataforma de monitoreo, guarda la información a lo menos 1 año por cada equipo suministrado en la red del operador de servicios de telecomunicaciones, generándose una gran colección de datos a nivel de *Big data*. Se puede formular que estos datos representan una serie temporal. El sistema *Network Performance Monitor* está monitoreando en tiempo real constantemente a los equipos que se encuentra activos. El mínimo *span* de tiempo es a cada 5 minutos, esta información se utilizará como *input* en nuestro modelo de predicción.

3.1.2 Estructura de redes neuronales RNN, LSTM, GRU y OS-ELM

A continuación se presentará las principales características de las redes neuronales RNN, LSTM, GRU y OS-ELM.

3.1.2.1 Estructura de red neuronal RNN, LSTM y GRU

La RNN es una red neuronal feed-forward que capta la información dinámica en los datos secuenciales a través de las conexiones periódicas de los nodos de la capa oculta [59]. El sistema de la RNN se representa en la Fig. 3.2.

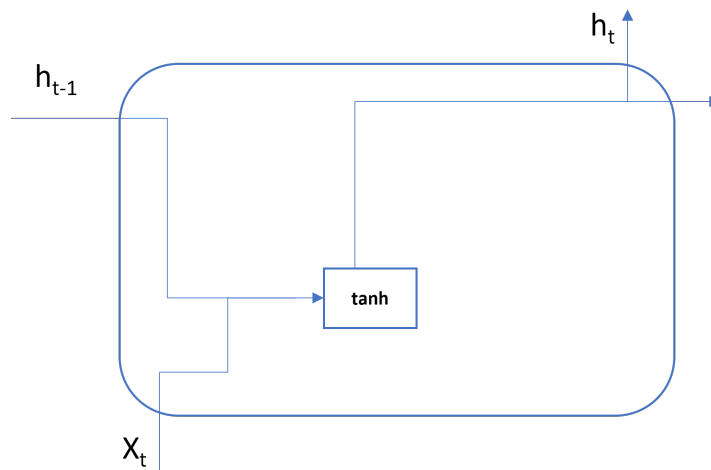


Figura 3.2. Estructura de la célula RNN

En el que x_t es la entrada de la secuencia en el tiempo t , h_t es la unidad de memoria de

la secuencia en el tiempo t y h_{t-1} almacena en caché la información anterior. En este sentido, h_t viene dado por la ecuación 3.1:

$$h_t = \tanh(x_t U + h_{t-1} W + b), \quad (3.1)$$

donde W , U y b representan el peso de entrada, el peso de la información previa y el sesgo respectivamente para un estado de célula actual de la RNN [59]. Aquí, las neuronas ocultas se ve afectada por la función tangente hiperbólica debido a es la función de activación. El rango de la función \tanh es de (-1 a 1). Las RNN se desarrollaron con el objetivo especial de superar los límites de las redes neuronales feed-forward en el modelado de secuencias, permitiendo modelar los datos de entrada en las observaciones actuales así como en las anteriores (h_{t-1}) [16].

Las principales desventajas de la RNN son la desaparición del gradiente y los problemas de explosión. La desaparición del gradiente radica que este llega a cero rápidamente y esto dificulta el aprendizaje de algunas dependencias a largo plazo. Con respecto a los problemas de explosión, este llega al infinito a una velocidad exponencialmente y su valor se convierte en *Not a Number*, como resultado del proceso inestable. Ambos problemas se deben a la naturaleza temporal del proceso de optimización, el cálculo de los gradientes del modelo tiene un efecto recurrente en los otros pasos. Así, el aumento o reducción de la norma de un gradiente tiene efectos exponenciales sobre el cálculo de los otros valores, lo que puede llevar a fenómenos de inestabilidad numérica en el proceso de optimización [24]. La solución al problema anterior es el modelo LSTM.

La estructura LSTM permite acumular información de forma selectiva en el tiempo. El cálculo del estado oculto de la célula viene dado por el sistema representado en la Fig. 3.3, donde tiene tres puertas: entrada, olvido y salida.

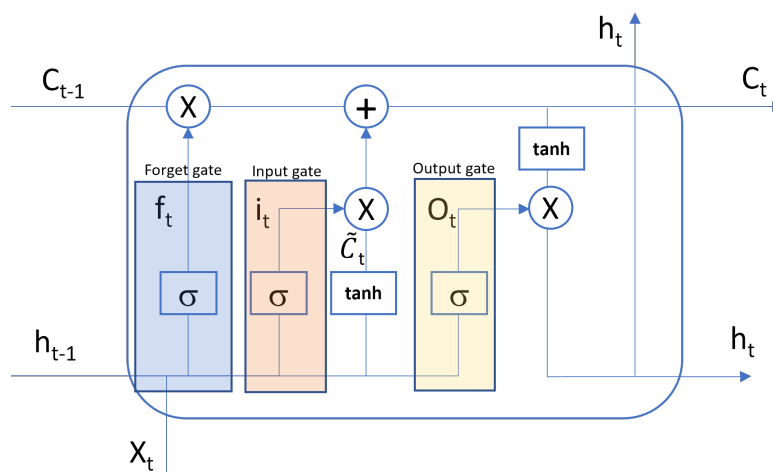


Figura 3.3. Estructura de la célula LSTM

En la ecuación 3.2, la puerta de entrada i_t gestiona el nuevo flujo de información en la unidad de memoria.

Para todas las ecuaciones x_t representa los datos de entrada, h_{t-1} representa el estado oculto del paso de tiempo anterior, y W^i , U^i son pesos de la puerta de entrada con b^i sesgo de entrada.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i + b^i), \quad (3.2)$$

La puerta de olvido f_t omite la información del estado real con la función sigmoide establecida en el estado oculto del paso de tiempo anterior h_{t-1} . La entrada real x_t junto con los pesos de la puerta de olvido W^f , U^f y el sesgo de la puerta de olvido b^f , se muestra en la ecuación 3.3.

$$f_t = \sigma(x_t U^f + h_{t-1} W^f + b^f), \quad (3.3)$$

En la ecuación 3.4, se representa la puerta de salida o_t que conduce la información necesaria para la activación de la salida de la unidad de memoria, y W^o , U^o son los pesos de la puerta de salida con b^o sesgo de la puerta de salida.

$$o_t = \sigma(x_t U^o + h_{t-1} W^o + b^o), \quad (3.4)$$

En la ecuación 3.5 la puerta de entrada i_t que se aplican al estado previamente existente de la unidad C_{t-1} . El estado actual de la unidad C_t se actualiza con la puerta de olvido f_t y \tilde{C}_t que es candidato a un estado de la célula en *timestamp* (t) y se representa en ecuación 3.6.

$$C_t = \sigma(i_t * \tilde{C}_t + f_t * C_{t-1}), \quad (3.5)$$

$$\tilde{C}_t = \tanh(h_{t-1} W^c + x_t U^c + b^c), \quad (3.6)$$

donde W^c , U^c y b^c son el peso y el sesgo del candidato para el estado de la unidad actual, respectivamente.

La etapa de salida h_t está calculado como se representa en la ecuación 3.7. La puerta de salida o_t y la función de activación \tanh se aplican al estado actual C_t y luego se le asigna a la siguiente unidad de memoria.

$$h_t = \tanh(C_t) * o_t, \quad (3.7)$$

El modelo LSTM controla el flujo de información de entrada. La ecuación 3.5 permite a la célula aprender dependencias de mayor duración. El cálculo del gradiente es ponderado selectivamente por los parámetros de las diferentes puertas lógicas como se muestra en las ecuación 3.4 y 3.6. Esto reduce el efecto recurrente de los valores extremos, lo que estabiliza el proceso de optimización. Además, permite que el modelo retenga la información en pasos de tiempo más largos [24].

Por otro lado, la red neuronal Gated Recurrent Units, GRU es análogo al LSTM pero con sólo dos puertas y menos parámetros. El cálculo del estado oculto de la célula viene dado por el sistema representado en la Fig. 3.4, donde tiene dos puertas: reinicio y actualización representados por r y z respectivamente. x representa la entrada, h_{t-1} representa la salida del paso anterior, h_t representa la salida real, por lo tanto, como se mostró en el modelo LSTM, W , U son pesos y el símbolo b indican el sesgo para las siguientes ecuaciones [60].

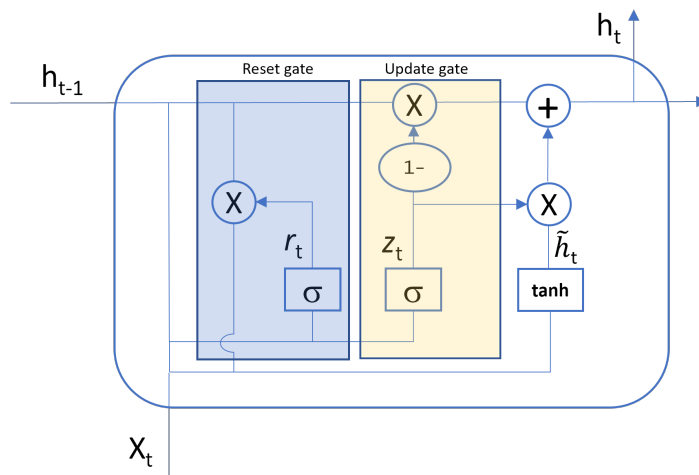


Figura 3.4. Estructura de la célula GRU

En la ecuación 3.8, la salida es la puerta de reinicio. Esta puerta es similar a la puerta de olvido de la célula LSTM. El resultado del vector de reinicio r representa los datos que se eliminarán de los pasos de tiempo ocultos anteriores.

$$r_t = \sigma(h_{t-1}U^r + x_tW^r + b^r), \quad (3.8)$$

La puerta de actualización ayuda al modelo GRU a determinar la cantidad de información anterior que debe pasar al futuro. Esto es muy valioso porque el modelo puede determinar la copia de toda la información pasada (anterior) y eliminar el problema del gradiente de fuga. En la ecuación 3.9 se representa la ecuación de salida de la puerta de actualización.

$$z_t = \sigma(h_{t-1}U^z + x_tW^z + b^z), \quad (3.9)$$

En la ecuación 3.10 se muestra la salida implícita y no la salida objetivo de la red GRU. La salida implícita en este punto, se verá afectada por la salida de la puerta de reinicio.

$$\tilde{h}_t = \tanh(x_tW + r_t * h_{t-1}U), \quad (3.10)$$

La ecuación 3.11 se utiliza para determinar la salida objetivo, que se establece a partir de la última salida y la salida real implícita. Si el valor de la puerta de actualización es cercano a cero, esto indica que los datos de las características del paso anterior son necesarios [60].

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \quad (3.11)$$

En las ecuaciones 3.4, 3.6, 3.10 y 3.11 el símbolo * indica el producto de Hadamard. En los tres modelos RNN, LSTM y GRU, se tienen dos salidas iguales h_t , porque una es la retroalimentación a la célula, debido a que los modelos son recurrentes.

3.1.2.2 Estructura de red neuronal OS-ELM

Para comprender a OS-ELM, se revisará a la estructura base que posee. Extreme Learning Machine (ELM) se compone de tres partes: capas de entrada, oculta y de salida, presentadas en la Fig. 3.5. Los pesos de entrada y el sesgo oculto se atribuyen al azar y mantienen el mismo valor durante todo el proceso de entrenamiento. Mientras tanto, los pesos de salida se determinan de forma analítica [61]. Los pesos de salida se calculan utilizando la matriz pseudoinversa [47].

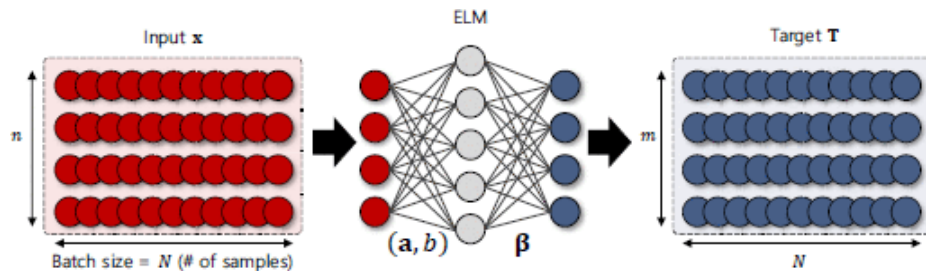


Figura 3.5. Capas de ELM, Extreme learning machine

Los datos de entrada mapean un espacio de características aleatorias ELM de L -dimensional, y la salida de la red es:

$$\sum_{i=1}^L \beta_i h_i(x) = h(x)\beta, \quad (3.12)$$

En la que $\beta = [\beta_1, \dots, \beta_L]^T$ es la matriz de peso de salida entre los nodos ocultos. Los nodos de salida, $h(x) = [g_1(x), \dots, g_L(x)]$ son las salidas de los nodos ocultos para la entrada x ; y $g_i(x)$ es la salida del i -ésimo del nodo oculto. Si se dispone de N muestras de entrenamiento $(x_i, t_i)_{i=1}^N$, ELM es capaz de resolver la solución del problema [47]:

$$H\beta = T, \quad (3.13)$$

$T = [t_1, \dots, t_N]^T$ son etiquetas de destino, y $H = [h^T(x_1), \dots, h^T(x_N)]^T$. Los pesos de salida β se pueden calcular a partir de:

$$\beta = H^\ddagger T, \quad (3.14)$$

En la ecuación 3.13 H^\ddagger es la matriz inversa generalizada de Moore-Penrose. Para tener mejores resultados de la generalización y mejorar la robustez de la solución, se introduce un término de regularización (C), como se muestra en la ecuación 3.15 :

$$\beta = \left(\frac{I}{C} + H^T H \right)^{-1} H^T T, \quad (3.15)$$

donde I indica la matriz de identidad.

Una red neuronal que se basa en la red ELM es la llamada Online-Sequential Extreme Learning Machine, la cual se presentará a continuación:

Online-Sequential Extreme Learning Machine, OS-ELM es una red ELM que permite al algoritmo aprender en línea datos secuenciales, parte por parte utilizando el método de mínimos cuadrados recursivos [47].

OS-ELM consta de la fase de inicialización y de la fase de aprendizaje secuencial.

Inicialización: Se comienza con la matriz H_0 , el cual es creada para ser utilizada en la fase de aprendizaje.

Fase de aprendizaje secuencial: En esta parte, los datos pueden ser aprendidos parte por parte (de diferentes tamaños). Cuando los datos son aprendidos, ya no son necesarios y son eliminados. OS-ELM se implementa de la siguiente manera [47]:

Comienza el aprendizaje de la utilización de un pequeño trozo de datos iniciales para el entrenamiento $(x_i, t_i)_{i=1}^{N_0}$ del conjunto de entrenamiento asignado, $n \leq N_0 < N$:

1. Los valores se generan al azar del nodo oculto $(a_i, b_i), I = 1, \dots, n$.
2. Cálculo de la matriz de salida inicial de la capa oculta H_0 .
3. Determinar el peso de salida inicial $\beta_0 = P_0 H_0 T_0$ with $P_0 = (H_0^T H_0 + \frac{I}{C})^{-1}$ and $T_0 = [t_1, \dots, t_{N_0}]^T$.

4. Establecer $k = 0$.
5. Establecer $(k + 1)$ i-ésimo bloque de entrada de datos nuevos.
6. Cuantifica la matriz de salida parcial de la capa oculta H_{k+1} and set T_{k+1} .
7. Cuantifica el peso de la salida β_{k+1} :

$$P_{k+1} = P_k - P_k H_{k+1}^T (I + H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k, \quad (3.16)$$

$$\beta_{k+1} = \beta_k + P_{k+1} H_{k+1} (T_{k+1} - H_{k+1} \beta_k), \quad (3.17)$$

8. Establecer $k = k + 1$. Ir al paso 5, donde $k + 1$ indica la $(k + 1)$ i-ésima de datos de entrada con k creciente desde cero, y H_{k+1} simboliza la salida de la capa oculta para la $(k + 1)$ i-ésima de datos de entrada.

La precisión de la predicción a corto plazo del modelo OS-ELM se puede potenciar utilizando el valor de olvido, que se incorpora al OS-ELM en [62]. Con un valor de olvido constante $\lambda \in (0, 1]$, ecuación 3.16: cambia como sigue:

$$P_{k+1} = \frac{1}{\lambda} P_k - P_k H_{k+1}^T (\lambda^2 + \lambda H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k, \quad (3.18)$$

Si $\lambda = 1$, la ecuación 3.18 es la misma que la ecuación 3.16, esto significa que el modelo OS-ELM no olvida nada. El valor de olvido λ permite que el OS-ELM siga olvidando los datos de entrada en el proceso de aprendizaje para reducir el efecto negativo de la siguiente fase [47].

3.1.3 Procesamiento de datos para redes neuronales RNN, LSTM, GRU y OS-ELM

En esta etapa, la base de datos a utilizar o el tren de datos es el que nos entrega el sistema *Network Performance Monitor*, se debe adecuar a lo que se requiere buscar. En este caso se adecuará los datos según la red neuronal a utilizar, el cual se muestra a continuación.

3.1.3.1 Procesamiento de datos para redes neuronales RNN, LSTM, GRU

La transformación de estos tres pasos será necesaria para procesar mejor los datos en la implementación de las redes neuronales recurrentes RNN, LSTM y GRU.

1. Transformar los datos en un problema de aprendizaje supervisado: En el problema de las series temporales, los datos se modifican de la siguiente manera: La observación en el

último paso de tiempo ($t-1$) como la entrada y la observación en el paso de tiempo actual (t) como la salida. Esto representa la ventana de deslizamiento de un solo paso. Principalmente es la única variable para comparar por lo que se trata de un problema univariado [42].

2. Datos de series temporales, dependientes del tiempo: La tendencia se puede eliminar de las observaciones y luego volver a la escala de predicción original. Una forma estándar de eliminar una tendencia es diferenciar los datos.
3. Normalizar las observaciones: La función de activación por defecto del modelo LSTM es la tangente hiperbólica (\tanh), que tiene valores entre -1 y 1. Se normalizará las observaciones de la misma manera, es decir, entre -1 y 1. Esta regularización ayuda a evitar corromper el conjunto experimental con información del conjunto de datos de prueba.

3.1.3.2 Procesamiento de datos para redes neuronales OS-ELM

La transformación de estos tres pasos será necesaria para procesar mejor los datos en la implementación de las redes neuronales OS-ELM.

1. Transformar los datos en un problema de aprendizaje supervisado: Al preparar los datos, el modelo se instruye de la misma manera que las redes neuronales recurrentes, es decir, la ventana deslizante o paso de predicción será de un solo paso.
2. Función de activación: La función de activación de la red neuronal OS-ELM es la función sigmoide [63]. La función de activación ReLU (*Rectified Linear Unit*), también conocida como función de rampa se prueba en [19], teniendo malos resultados en comparación con la función sigmoide.
3. Normalizar las observaciones: En el modelo OS-ELM la recomendación de escala es normalizar los datos restando la media y dividiendo por la desviación estándar.

3.1.4 Datos de entrenamiento y de testeo

El conjunto de datos se divide en dos partes: conjuntos de datos de entrenamiento y de testeo. Los conjuntos de datos de entrenamiento y de testeo se utilizan para construir y validar el modelo, respectivamente. Para comparar y validar los modelos, típicamente se valida con el método llamado encadenamiento hacia adelante (*forward-chaining*) [64], [65]. Cuando se trata de datos de series temporales, la validación cruzada tradicional no debe utilizarse por dos razones, las dependencias temporales y la elección arbitraria del conjunto de pruebas. La primera se refiere

a los datos de series temporales, en los que se debe tener especial cuidado al dividir los datos para evitar la fuga de datos [66]. Mientras que el segundo se refiere a que en la validación cruzada tradicional el conjunto de prueba es bastante arbitrario, y esa elección puede significar que el error del conjunto de prueba es una mala estimación del error en un conjunto de prueba independiente. Para resolver este problema, se utiliza un método llamado validación cruzada anidada [67].

Para este caso, se aplicará que el conjunto de testeo (*testing*) sea el mismo para todas las simulaciones, por lo que será posible extraer conclusiones en función de variaciones de las cuatro redes neuronales y de los distintos hiper-parámetros.

En el caso del conjunto de entrenamiento (*training*) se agregarán datos, para verificar si al agregar mas datos de entrenamiento, mejora las métricas de error. Como la serie temporal de datos es estacional, se agregaran días anteriores ($t-x$ días), como se muestra en la Fig. 3.6. La cantidad de datos que se agreguen serán dependiendo del *span* de tiempo que tenga el sistema de monitoreo.

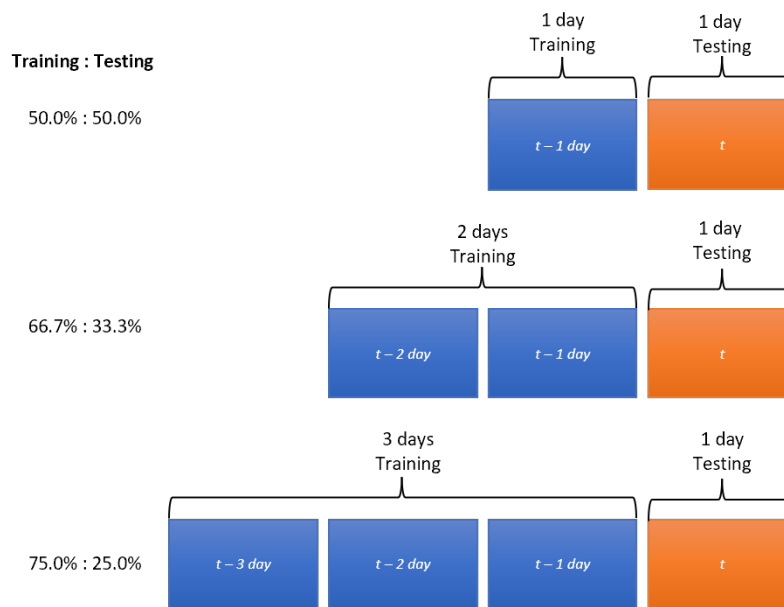


Figura 3.6. Conjunto de datos de entrenamiento y testeo.

Teniéndose así tres conjuntos de datos en porciones diferentes para las simulaciones, los cuales son - *Training* : *Testing*:

- Primer conjunto = 50.0 % : 50.0 %
- Segundo conjunto = 66.7 % : 33.3 %
- Tercer conjunto = 75.0 % : 25.0 %

3.1.5 Hiper-parámetros de redes neuronales RNN, LSTM, GRU y OS-ELM

3.1.5.1 Hiper-parámetros de redes neuronales RNN, LSTM, GRU

Estos pasos explican los hiper-parámetros fundamentales del modelo RNN, LSTM y GRU que se van a implementar.

1. Número de neuronas: Es el número de capas ocultas añadidas a la célula RNN, LSTM y GRU.
2. Épocas (*epochs*): Es el número de veces que cada dato de entrenamiento pasará por la red neuronal.
3. Timesteps (*lags*): El número de pasos de tiempo especificados. Determina el número de variables de entrada x utilizadas para pronosticar el siguiente paso de tiempo h , como se muestra en la Fig. 3.7.

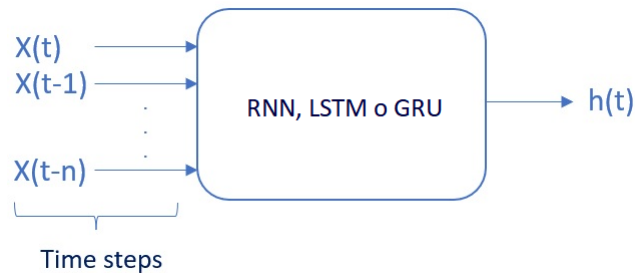


Figura 3.7. Timesteps (lags) en redes neuronales recurrentes.

4. Optimizador Adam: El algoritmo Adam [68], es aquel que combina RMSProp con momentum. Hasta el momento no hay un algoritmo que tenga un desempeño superior al de los demás en distintos escenarios [69], por lo que se recomienda usar el algoritmo de optimización con el que el usuario se sienta más cómodo al momento de ajustar los hiper-parámetros. Para ejecutar las simulaciones se configurará para las tres redes neuronales RNN, LSTM y GRU el algoritmo de optimización basado en el gradiente de Adam. En [70] se indica que es un optimizador muy popular en las redes de aprendizaje profundo en los últimos años.

3.1.5.2 Hiper-parámetros de redes neuronales OS-ELM

Estos pasos explican los hiper-parámetros fundamentales del modelo OS-ELM que se van a implementar.

1. Número de neuronas: Es el número de capas ocultas añadidas a la célula OS-ELM.
2. Valor de olvido (*Forgetting Factor*): El valor de olvido, también llamado *Forgetting Factor* permite a la red neuronal OS-ELM olvidar continuamente los datos de entrada obsoletos en el proceso de entrenamiento, para reducir su mal efecto al siguiente aprendizaje. Si *Forgetting Factor* = 1, significa que la red neuronal OS-ELM no olvida nada. Si *Forgetting Factor* es menor a 1, comienza a olvidar datos.

3.1.6 Métricas

3.1.6.1 Error de raíz cuadrada media (RMSE)

El error de raíz cuadrada media RMSE, muestra la desviación media de los valores predichos respecto a los valores objetivo (determina la similitud entre dos conjuntos) y viene dada por:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (3.19)$$

donde, y_i es el valor real, \hat{y}_i es el valor predicho producido por el modelo y n es el número total de muestras.

3.1.6.2 Error Absoluto Medio (MAE)

Es el promedio sobre la muestra de prueba de las diferencias absolutas entre la predicción y la observación real, donde todas las diferencias individuales tienen el mismo peso. Mide la magnitud media de los errores en un conjunto de predicciones, y se define por:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3.20)$$

donde, y_i es el valor real, \hat{y}_i es el valor predicho producido por el modelo y n es el número total de muestras.

3.1.6.3 Error Porcentual Absoluto Medio (MAPE)

También conocido como Desviación Media Absoluta Porcentual (MAPD), es una métrica de evaluación para problemas de regresión. La idea de esta métrica es ser sensible a los errores

relativos, se define por:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{\max(\epsilon, |y_i|)}, \quad (3.21)$$

donde, y_i es el valor real, \hat{y}_i es el valor predicho producido por el modelo, n es el número total de muestras y ϵ es un número arbitrario pequeño pero estrictamente positivo para evitar resultados indefinidos cuando y es cero.

Para todos las redes neuronales, se realizan 100 ejecuciones y se obtiene su media para cada métrica, ya que en estudio [71] se indica que en los problemas de optimización se deben realizar más de 100 pruebas para encontrar un valor verdadero u óptimo de la solución, por ser un problema heurístico.

3.1.6.4 Tiempo computacional

El tiempo computacional es el tiempo necesario para realizar un modelo computacional a implementar, se mide en segundos (s).

El hardware y el software que se utilizarán en las simulaciones referente a las redes neuronales se especifican en la Tabla 3.1.

Tabla 3.1. Especificaciones del sistema

Hardware	
CPU	Intel(R) Core(TM) 8600K at 5.1Ghz
RAM	32 Gb
Graphics card	NVIDIA GeForce(R) RTX 2080.
Software	
Python	3.7.10
Tensorflow	2.2.0
Keras	2.3.0
Pandas	1.2.4
Scikit-Learn	0.24.1

3.2 METODOLOGÍA EN EFICIENCIA ENERGÉTICA EN BUNDLE ETHERNET

Para entender la metodología en eficiencia energética en Bundle Ethernet, la Fig. 3.8 muestra como se compararán los tres casos. El primer caso corresponderá al base (sin eficiencia energética), el cual será cuando el sistema se encuentra en las condiciones actuales, es decir, como esta funcionando el sistema al día de hoy. El segundo y tercer caso será el desarrollo de algoritmos que se encargarán de encender (*turn ON*) o apagar (*turn OFF*) los puertos correspondientes al lado que da servicios el *router*, con la finalidad de reducir el consumo de energía de la red mientras se satisface la demanda de tráfico.

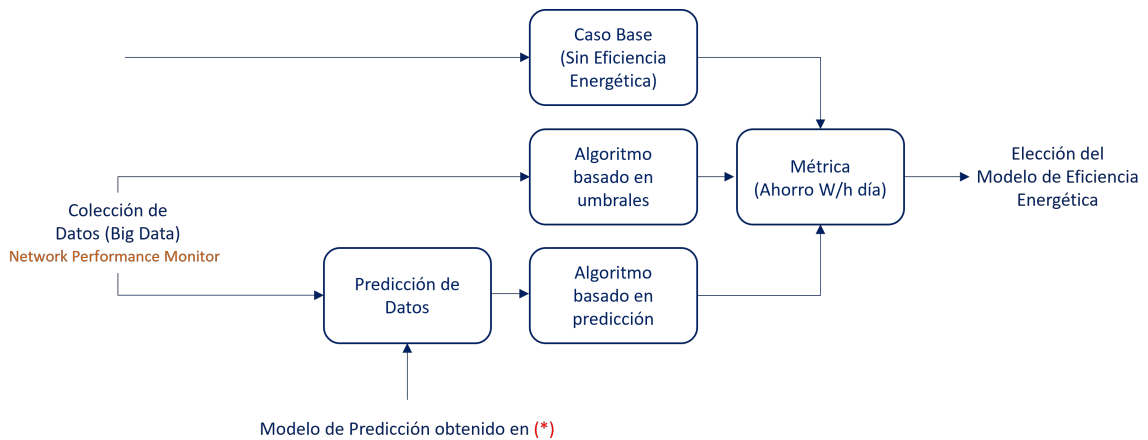


Figura 3.8. Metodología en eficiencia energética en Bundle Ethernet.

El primer algoritmo estará basado en el tiempo pasado de los valores en brutos (*raw value*, (*rv en t-1*)) del sistema de monitoreo *Network Performance Monitor*, en el cual se agregará un umbral o factor de seguridad del 100%, es decir, si el tráfico en el tiempo *t-1* fue de 100Gbps, se aplicará al tiempo *t* un capacidad de al menos 200Gbps, y en función de esa capacidad se dispondrá el BE requerido. Este algoritmo será llamado “algoritmo basado en umbrales”.

El segundo algoritmo tendrá como *input* el modelo de predicción que se seleccionará en la metodología de predicción de tráfico, vista en la Fig. 3.1. Con este modelo de predicción se tendrá los valores predichos (*prediction value*, (*pv en t*)) para cada punto en el tiempo *t*, con esto se activará la capacidad en el BE. Este algoritmo será llamado “algoritmo basado en predicción”.

3.2.1 Algoritmos basado en umbral y en predicción

A continuación se presentarán la lógica de ambos algoritmos escrito en lenguaje pseudocódigo.

3.2.1.1 Algoritmo basado en umbrales

El algoritmo basado en umbrales, que se muestra en la Fig. 3.9, requiere las siguientes entradas para que se pueda ejecutar, estas son:

- rv_{t-1} : valor en bruto de la velocidad del enlace en Gbps en un *timestamp* anterior, es decir *raw value*, (*rv en t-1*), este valor se obtiene de la plataforma *Network Performance Monitor* y es una variable continua.
- nl : número de enlaces que posee inicialmente el LAG o BE, variable discreta entera, adimensional.
- pb : ancho de banda del puerto (*port bandwidth*) y es una variable continua medida en Gbps.

Algorithm 1 : Algorithm based on Threshold with raw value in $t - 1$

Require: raw value in $t-1$: rv_{t-1} ; numbers of links in LAG: nl ; port bandwidth: pb

```
 $x \leftarrow rv_{t-1} / pb$  ▷  $x$  is defined as a ratio variable
 $pa \leftarrow 0$  ▷  $pa$  is defined as ports active
 $pd \leftarrow nl - pa$  ▷  $pd$  is defined as ports deactivate
if  $x > 0$  then
   $pa_u = (\text{integer}(x) + 1) * 2$  ▷  $pa_u$  is defined as ports active update
  if  $pa_u > nl$  then
     $pa = nl$ 
     $pd = nl - pa$ 
    activate  $pa$  ports ▷ set in router activate ports
    deactivate  $pd$  ports ▷ set in router deactivate ports
  else
     $pd = nl - pa_u$ 
     $pa = pa_u$ 
    activate  $pa$  ports ▷ set in router activate ports
    deactivate  $pd$  ports ▷ set in router deactivate ports
  end if
end if
```

Figura 3.9. Algoritmo basado en umbrales.

Como variables que inicializan el algoritmo, se define x como una variable de razón entre las variable rv_{t-1} y pb . Además de pa que representa los puertos activos del enlace, que se inicializa en 0, y es una variable entera discreta. pd como la diferencia de nl y pa . Esta última variable, se obtienen de los puertos que se tienen desactivados al momento de ejecutar el algoritmo.

Ya dentro de la ejecución del algoritmo, mientras la variable x sea mayor a 0, es decir, el *router* este presente con tráfico, ejecuta la variable pa_u , el cual corresponde a dar un umbral del 100% más al valor obtenido del tráfico anterior, debido a que se están utilizando los valores del pasado para esta acción. En otras palabras pa_u corresponde a los puertos que deben estar activos, según el tráfico anterior más el factor de seguridad que se define como el doble, debido a no tener certeza del tráfico futuro.

Después de esto, se encuentra la condición de que si el número de puertos activos sobrepasa a la cantidad de número de enlaces del LAG o BE, definido como nl , se debe limitar al máximo definido por este *port-channel*. En caso contrario, calcula la diferencia de puertos a activar y desactivar. En ambas condiciones se ejecutan la cantidad de puertos a activar y desactivar al *router* para el próximo *timestamp*.

3.2.1.2 Algoritmo basado en predicción

Para este algoritmo se utiliza la misma lógica que el anterior (basado en umbrales), la única e importante diferencia es que se tiene como entrada o *input* el valor futuro de tráfico obtenido en la predicción. Como se muestra en la Fig. 3.10 las entradas requeridas son el valor de predicción pv , el número de enlaces en el LAG nl y el ancho de banda del puerto pb .

Algorithm 2 : Algorithm based on Prediction model

Require: prediction value: pv ; numbers of links in LAG: nl ; port bandwidth: pb

```

 $x \leftarrow pv / pb$                                 ▷  $x$  is defined as a ratio variable
 $pa \leftarrow 0$                                     ▷  $pa$  is defined as ports active
 $pd \leftarrow nl - pa$                                ▷  $pd$  is defined as ports deactivate
if  $x > 0$  then
   $pa_u = integer(x) + 1$                              ▷  $pa_u$  is defined as ports active update
  if  $pa_u > nl$  then
     $pa = nl$ 
     $pd = nl - pa$ 
    activate  $pa$  ports                                ▷ set in router activate ports
    deactivate  $pd$  ports                               ▷ set in router deactivate ports
  else
     $pd = nl - pa_u$ 
     $pa = pa_u$ 
    activate  $pa$  ports                                ▷ set in router activate ports
    deactivate  $pd$  ports                               ▷ set in router deactivate ports
  end if
end if

```

Figura 3.10. Algoritmo basado en predicción

Para que ambos algoritmos funcionen correctamente, las unidades de medida de las variables pv y pb deben ser iguales.

3.2.2 Métricas

Para poder comparar ambos algoritmos más el caso base, se utilizará la unidad de medida de energía Watt-hora (Wh). Watt-hora es una forma de medir la cantidad de trabajo generado o realizado. El ahorro se producirá en la diferencia del consumo en el caso base a un día respecto a cada algoritmo. Lo que es lógico, el algoritmo elegido será el ahorro que se produzca más en comparación al caso base.

CAPÍTULO 4

ESCENARIO DE SIMULACIÓN Y RESULTADOS

4.1 ESCENARIO DE SIMULACIÓN

Escenario de simulación se refiere a describir la topología de red a utilizar en este estudio, la descripción del tráfico para elegir la mejor red neuronal en términos de predicción y tiempo computacional, y las características del equipamiento a utilizar, para modelar el ahorro de energía que se requiere como objetivo.

4.1.1 Topología de red

Como se menciono anteriormente, los BE o LAGs permiten la agrupación lógica de varios enlaces físicos Ethernet, esta agrupación se trata como un único enlace y permite la suma de la velocidad nominal de cada puerto físico Ethernet utilizado para obtener un enlace troncal de alta velocidad. En un TSO se puede encontrar diferentes routers que se conectan entre sí. Como se muestra en la Fig. 4.1, se puede ver que el LAG se conecta con n interfaces de fibra óptica generando un LAG de n enlaces. Con el paso del tiempo, las interfaces de la red de transporte óptico (OTN) y las interfaces de los routers han ido creciendo en las redes centrales o *Core* IP ópticas de un TSO hasta llegar a interfaces de 400GE en otros operadores [72], por lo que va en aumento a medida que la demanda de tráfico crece y en función del desarrollo de esta tecnología.

Además, muchas de las implementaciones en los TSO son redundantes a nivel geográfico como parte del diseño de redes de comunicaciones fiables (DRCN) [73], lo que implica en este

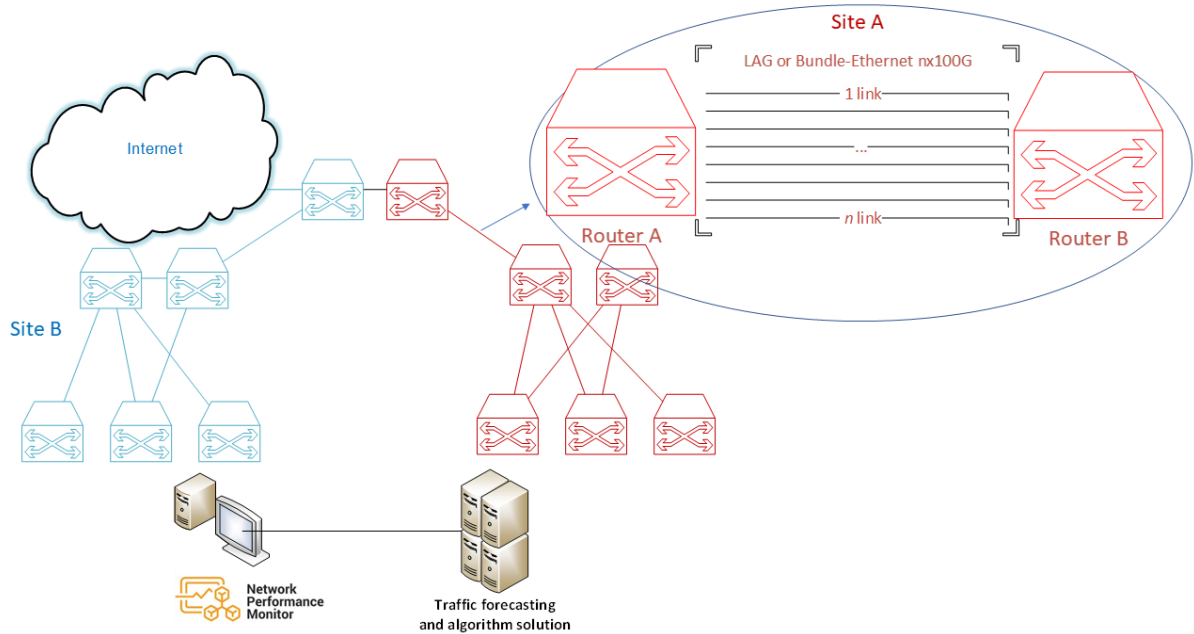


Figura 4.1. Agrupación de enlaces en un TSO.

caso, como se muestra en la Fig. 4.1 que el sitio A es igual al sitio B, esto significa que si el router del sitio A se apaga (falla), el tráfico de red conmutará al sitio B por el enlace llamado interplano (el cual conecta a ambos sitios). Lo mismo sucede en caso contrario.

En la Fig. 4.2 se muestra la arquitectura de una red de distribución de contenidos (CDN). El *router* CDN agrega diferentes proveedores de contenidos, como Google, Netflix, Facebook, Microsoft, Akamai, entre otros. La red CDN es un grupo de servidores distribuidos geográficamente e interconectados. Proporcionan contenidos de Internet en caché desde la ubicación de red más cercana al usuario para acelerar la entrega de información.

La salida a Internet la proporciona el *router* llamado comúnmente en inglés *Internet Gateway Router* (IGR). IGR es el nodo que agrega todo el tráfico de una red TSO y se comunica con Internet. Ambos *routers* IGR y CDN están conectados con 8 interfaces de 100GE de fibra óptica que generan un LAG o BE de 8 enlaces, el cual es de 800 Gbps de capacidad. El tráfico real entre ambos *routers* es de aproximadamente 400 Gbps, dejando para respaldo la misma capacidad para conmutar a otro sitio (espejo) en caso de falla.

En cuanto al protocolo que regula a los proveedores de hardware y proporciona orientación sobre la práctica de la agregación de enlaces para las conexiones de datos, la IEEE definió al protocolo de control de agregación de enlaces (LACP, Link Aggregation Control Protocol) dentro del IEEE 802.3ad [74]. El cual es un método basado en un estándar para controlar la agregación de enlaces físicos de red. LACP modo activo es el protocolo configurado generalmente en

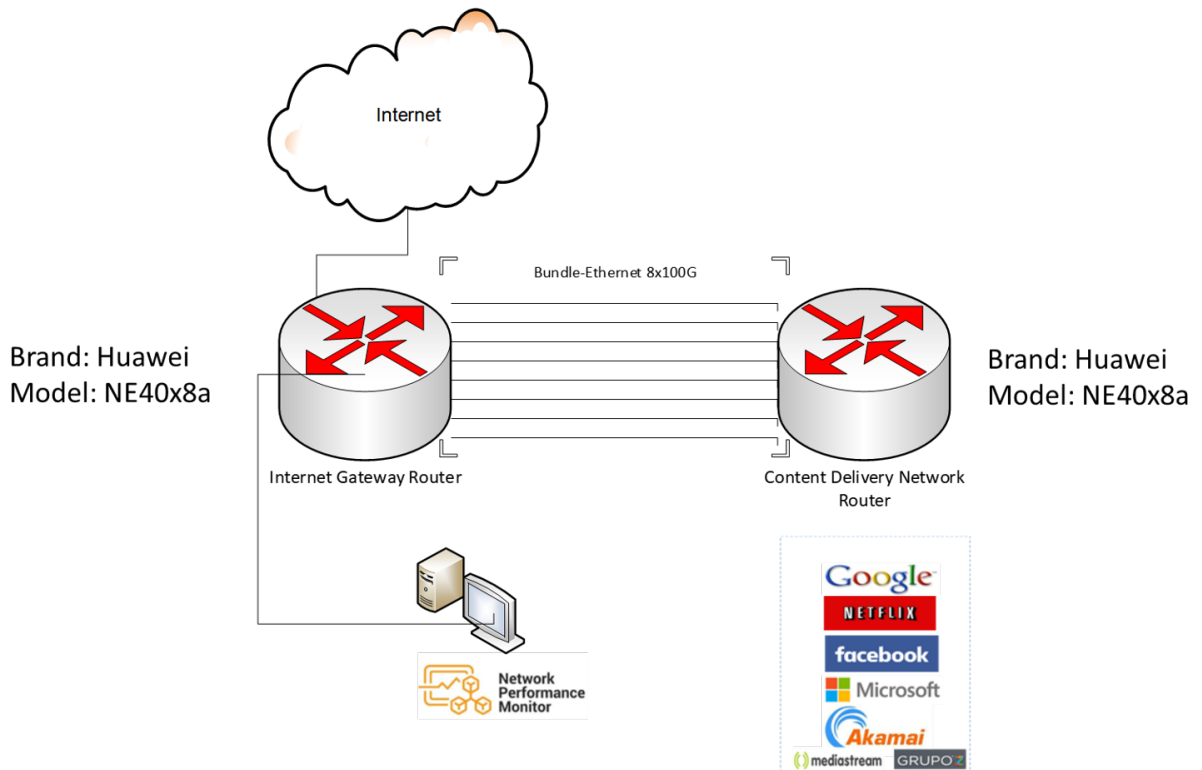


Figura 4.2. Arquitectura de red CDN en un TSO.

los equipos para gestionar los enlaces agregados. Esto significa que la interfaz se encuentra en un estado de negociación activa permanentemente. LACP se ejecuta en cualquier enlace que esté configurado para estar en estado activo. El puerto en modo activo también inicia automáticamente las negociaciones con otros puertos iniciando paquetes LACP. En él se configura LACP estático, el cual aumenta el ancho de banda de la interfaz y proporciona fiabilidad. Cuando un enlace Eth-Trunk o miembro de BE falla o no se encuentra activo, el tráfico se reparte automáticamente a otros enlaces disponibles, evitando así la interrupción del tráfico. Además, las interfaces Eth-Trunk que funcionan en modo LACP estático pueden implementar el balanceo de carga.

Cuando se tiene presente un BE o LAG, todos los enlaces están activos. En consecuencia, el consumo de energía se produce en todos los enlaces ópticos, debido a que el mecanismo de LAG envía constantemente paquetes de prueba para verificar si el enlace o miembro está activo o inactivo. A nivel energético, es un desperdicio de energía debido a que la capacidad del enlace debe estar proyectado a la demanda de tráfico *peak* de la red y en otros casos a la redundancia establecida geográficamente.

Para modelar el consumo de energía de un *router*, primero se debe conocer cuanto es la potencia. En la Fig. 4.3, se muestra el lado de servicios que se considerará en el presente estudio.

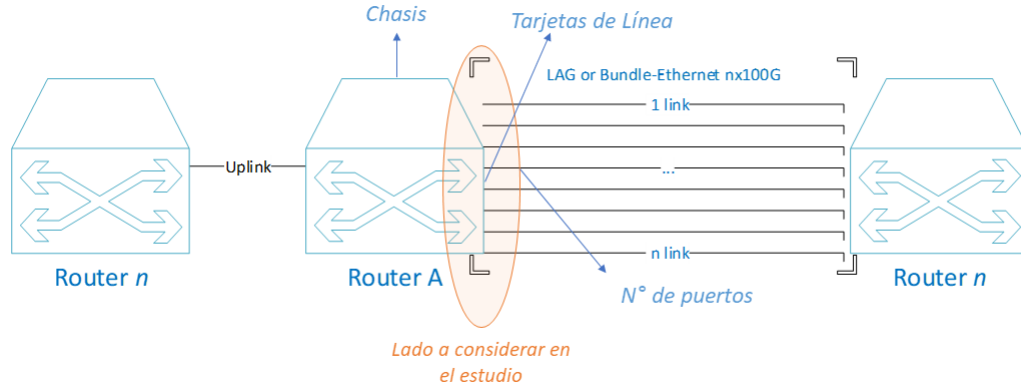


Figura 4.3. Lado del *router* a ser considerado.

Según [5] para calcular el consumo del equipo, este consta principalmente de un chasis, número de tarjetas de línea y el número de puertos. El consumo de energía del chasis y las tarjetas de línea es fijo, independientemente de la carga de tráfico, debido a que siempre estarán activas, por lo que la potencia de un *router* P_r puede expresarse mediante la ecuación 4.1:

$$P_r = P_{ch} + N_c * P_c + \sum_{i=0}^n N_{pi} * P_{pi} * fu_i, \quad (4.1)$$

Con P_{ch} como potencia del chasis, que es la potencia base del equipo. N_c es el número de tarjetas de línea y P_c la potencia de la tarjeta de línea. N_p es el número de puertos, P_p es la potencia de los puertos y fu es el factor de utilización de los puertos. El factor de utilización del puerto es el porcentaje entre el tráfico utilizado y la capacidad del puerto.

Cuando se habla de interfaces de 100GE, existen estándares que establecen distintas normativas técnicas para propósitos diferentes. Cuando se refiere a conexiones dentro del *Core*, el estándar 100GBASE-LR4 es el más utilizado [75] dado que corresponde a la especificación de la capa física del IEEE 802.3 para 100Gb/s con codificación 100GBASE-R a través de cuatro carriles WDM en fibra monomodo, con un alcance de al menos 10 km [76]. En el estándar 100GBASE-LR4, 100GBASE-SR10, 100GBASE-SR4 y 100GBASE-ER4, la eficiencia energética (EEE) no está disponible [77]; todos estos estándares el medio común por donde se transmiten los datos es fibra óptica. En la norma IEEE 802.3bm se indica que EEE en 100GBASE-LR4 es opcional. Además, actualmente muchos proveedores aún no han integrado esta normativa en su fabricación. Cuando la EEE no está activa, los estándares de Ethernet funcionan todo el tiempo a plena potencia, por lo que consumen el 100% de la energía, independientemente de la carga de tráfico [78]. Por lo que, para nuestro caso se puede suponer que el factor de utilización será siempre igual a 1.

4.1.2 Descripción del tráfico

Para efecto del estudio, se utilizará el tráfico real de la arquitectura que se muestra en la Fig. 4.2. El conjunto de datos representa la actividad del 13 de noviembre de 2021 al 16 de noviembre de 2021 (4 días), que consiste en el tráfico del *router* CDN ubicado en Santiago, Región Metropolitana, Chile (-33.444285499124504, -70.65611679943314). Los datos históricos fueron capturados desde el sistema de monitoreo, teniéndose datos brutos o *data raw* que contenían la hora, el tráfico medio entrante, el tráfico medio saliente, el tráfico máximo entrante, el tráfico máximo saliente, etc. En el conjunto de datos el *peak* de tráfico de salida es la variable que se utilizará, dado que es el tráfico máximo que puede tener la red. En el sistema de monitoreo, el tiempo mínimo de recolección de datos es cada 5 minutos. En consecuencia, el conjunto de datos constará de una variable de tráfico medida en Gbps, cada cinco minutos.

Para efecto del entrenamiento, y según lo mostrado en la metodología, se dividirán tres grupos según el número de días para entrenar la red neuronal. Como se muestra en la Fig. 4.4, cada día corresponde a 288 observaciones y cada punto representa cinco minutos con su respectivo valor en Gbps, debido a que este conjunto de datos funciona como una serie temporal univariante. Por lo tanto, la división de grupos de entrenamientos queda como sigue, manteniendo la proporción indicada en la Fig. 3.6 :

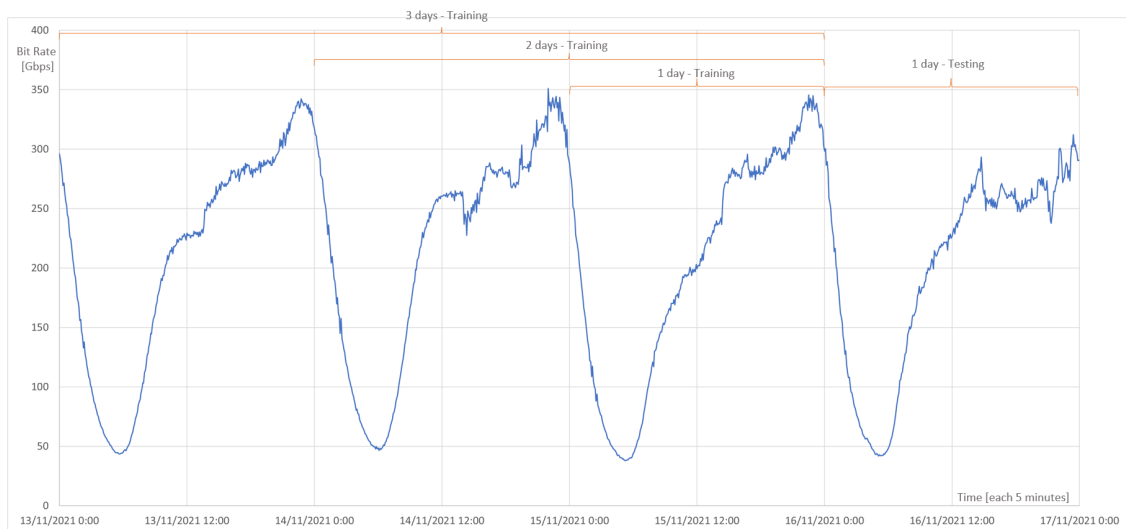


Figura 4.4. Tráfico CDN - Conjunto de datos final.

- Primer conjunto: 576 observaciones
 - Observaciones de entrenamiento: 288
 - Observaciones de testeo: 288
- Segundo conjunto: 864 observaciones

- Observaciones de entrenamiento: 576
- Observaciones de testeo: 288

- Tercer conjunto: 1152 observaciones
 - Observaciones de entrenamiento: 864
 - Observaciones de testeo: 288

Como la arquitectura de red *Core* agrega tráfico y se la entrega a los distintos clientes mediante el *router* CDN, el tipo de tráfico es estacional debido a que posee la característica horaria del uso nocturno de principales CDN como Netflix, Facebook, Google.

4.1.3 Características del equipamiento

En la arquitectura de red que se muestra en la Fig. 4.2 existe configurado un BE o LAG de 8 enlaces de 100GE de interfaz, indicado normalmente como 8x100GE. De acuerdo con el protocolo LACP, estos se configuran con el mismo *weight*. Esto significa que la repartición de carga es igual para todos los enlaces al momento de realizar la asignación de tráfico, y por ende para efecto de consumo de energía, están todos los enlaces activos.

El equipo estudiado es de marca Huawei, modelo NE40E-X8A, con una configuración base que consume 784W (potencia típica a 25°C), sin añadir tarjetas de servicio, ni enlaces *uplinks*. Si sólo se analiza el consumo del BE en un lado del equipo NE40E-x8A, como se muestra en la Fig. 4.3, se tiene la siguiente configuración y consumo que se muestra en la Tabla 4.1.

Tabla 4.1. Potencia por tarjeta y puertos de *router* CDN.

Slot	Board Info	Typical Power At 25°C(W)
Slot1	LPUF-480-E	290
Slot1-PIC0	PIC-2*100GBase-QSFP28	73
Slot1-PIC1	PIC-2*100GBase-QSFP28	73
Slot2	LPUF-480-E	290
Slot2-PIC0	PIC-2*100GBase-QSFP28	73
Slot2-PIC1	PIC-2*100GBase-QSFP28	73

Los datos de la configuración base, más el detalle de cada una de las tarjetas que componen al equipo, fueron obtenidos por la configuración actual del *router* en la red y se presenta como resumen en la Fig. A.1 del Anexo A. Por lo tanto, como todos los puertos se encuentran activos, el factor de utilización (f_{u_i}) será igual a 1. La potencia base se calcula con la ecuación 4.1, quedando como sigue:

$$P_r = 784 + 2 * 290 + \sum_{i=1}^8 36,5 * 1 = 1656W, \quad (4.2)$$

Por lo tanto, la potencia total base es de 1656W, cuando se encuentran todos los puertos activos.

4.2 RESULTADOS EN PREDICCIÓN DE TRÁFICO

Se presentarán los resultados de ambas simulaciones tanto para las redes *deep learning* RNN, LSTM, GRU y red neuronal OS-ELM.

4.2.1 Resultados simulación red neuronal RNN, LSTM y GRU

Se realizan las simulaciones variando tres hiper-parámetros: pasos de tiempo (numero de entradas) o más comúnmente conocidos como *lags*, número de neuronas en la capa oculta y épocas. Las variaciones de cada hiper-parámetro será la siguiente:

- Timesteps (*lags*): 1, 4, 8, 16 y 32.
- Número de neuronas: 1, 10 y 50.
- Epocas (*epochs*): 1, 10 y 100.

Una vez que se varié cada hiper-parámetro, se obtendrá los resultados de las métricas RMSE, MAE, MAPE y tiempo computacional. Notar que las métricas RMSE, MAE y MAPE son errores, por lo que se busca que sean cercanas a cero. Para una mejor comprensión de estos resultados, se realizarán gráficas para cada métrica, mostrando cada red neuronal en los diferentes conjuntos o días de entrenamiento. Recordar que los días de entrenamiento, son 1, 2 y 3 días, con 1 día de testeo, ya explicado en el punto anterior.

La tabulación de los resultados se presentan en los anexos B, C y D. Los valores de mejor rendimiento se muestran en negrita. El criterio fue tener al menos 2 métricas con menor valor en la configuración de los hiper-parámetros *lags*, número de neuronas y *epochs*, por cada grupo y red neuronal de aprendizaje profundo.

Para poder interpretar de mejor forma los gráficos, a continuación se detallará en la Tabla 4.2 el número de configuración de hiper-parámetro señalada en el eje *x*. Cada hiper-parámetro tiene la siguiente descripción (*lags*, número de neuronas, épocas), es decir, el hiper-parámetro número 1, tiene la configuración de *lags*: 1, número de neuronas: 1, épocas: 1; y así sucesivamente.

En la Fig. 4.5 se muestra los resultados de la métrica RMSE. En el eje *x* se representa el número de configuración de los hiper-parámetros indicado en la Tabla 4.2. A partir del número 16 cambia de el número de épocas a 10, al igual que en el número 31 a 100 épocas. En el eje *y* se representa el error cuadrático medio (RMSE). Los datos están en Mbps, por lo que el rango de estos errores van desde los 5.5 a 9.0 Gbps.

Se puede observar que a nivel general, la red neuronal RNN para las épocas en 1 y 10 se

Tabla 4.2. Configuración de Hiper-parámetros

Number	Hyperparameters Setting	Number	Hyperparameters Setting	Number	Hyperparameters Setting
1	1,1,1	16	1,1,10	31	1,1,100
2	4,1,1	17	4,1,10	32	4,1,100
3	8,1,1	18	8,1,10	33	8,1,100
4	16,1,1	19	16,1,10	34	16,1,100
5	32,1,1	20	32,1,10	35	32,1,100
6	1,10,1	21	1,10,10	36	1,10,100
7	4,10,1	22	4,10,10	37	4,10,100
8	8,10,1	23	8,10,10	38	8,10,100
9	16,10,1	24	16,10,10	39	16,10,100
10	32,10,1	25	32,10,10	40	32,10,100
11	1,50,1	26	1,50,10	41	1,50,100
12	4,50,1	27	4,50,10	42	4,50,100
13	8,50,1	28	8,50,10	43	8,50,100
14	16,50,1	29	16,50,10	44	16,50,100
15	32,50,1	30	32,50,10	45	32,50,100

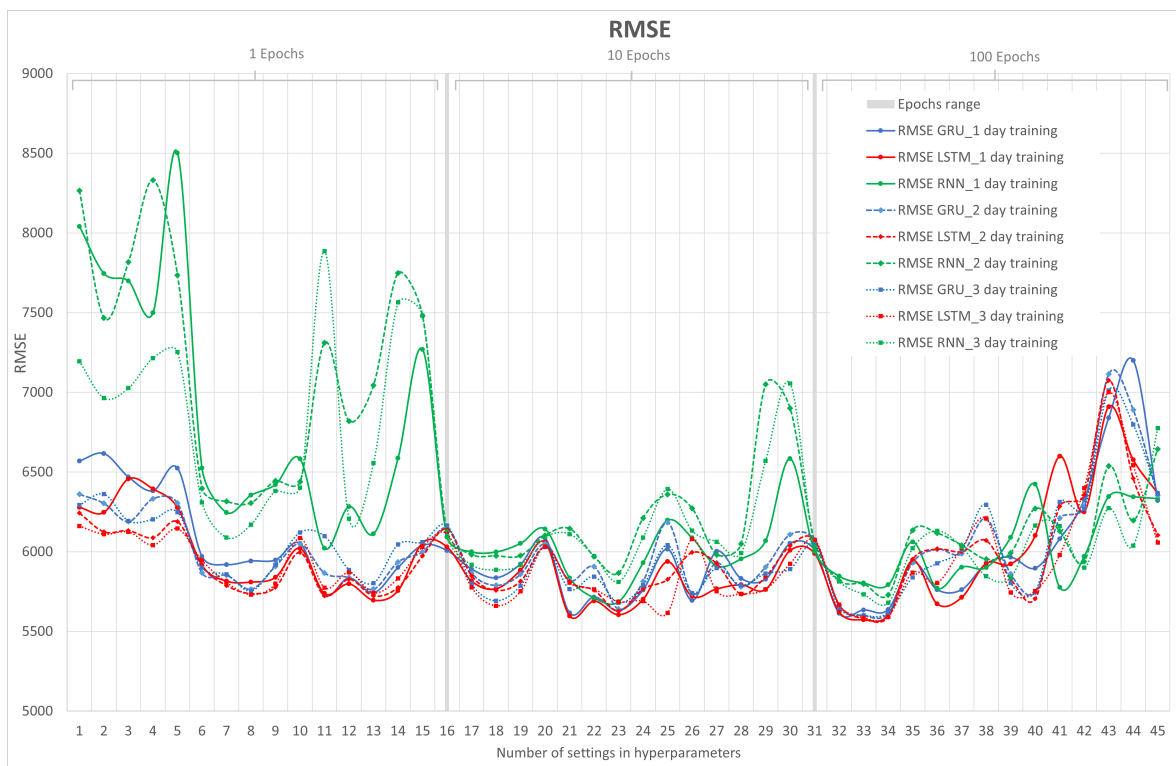


Figura 4.5. Métrica RMSE para redes RNN, LSTM y GRU.

comporta con un alto grado de error a diferencia de las redes neuronales LSTM y GRU para 1, 2 y 3 días de entrenamiento. Esto sucede debido a que presentan un menor número de épocas. Es por eso que a 100 épocas la red RNN mejora, incluso es mejor que otras redes en algunos casos. Recordar que cada ciclo de corrección de propagación hacia atrás y hacia adelante para reducir la pérdida se denomina época. La propagación hacia atrás consiste en determinar las mejores ponderaciones y sesgos de entrada para obtener un resultado más preciso o minimizar las pérdidas.

Otra conclusión de esta Fig. 4.5 es que debido a que la red neuronal RNN no tiene efecto memoria y que en 1 día de entrenamiento posee una menor cantidad de datos, por tanto, tiene una peor *performance* que en los 2 y 3 días de entrenamiento. Esto se minimiza en las otras redes neuronales como LSTM y GRU al tener un mayor efecto en la memoria que RNN.

Además se puede observar que desde el hiper-parámetro 40 al 45, en el cual están fijados los números de neuronas a 50 y épocas a 100, al variar el número de *lags* hace empeorar a las redes LSTM y GRU, respecto a la red RNN. Se puede inferir que puede estar influyendo de alguna manera el efecto memoria de estas 2 redes con muchas épocas y entradas lo que haría un empobrecimiento de estas 2 redes neuronales por un exceso de datos.

Finalmente en la métrica RMSE, como se puede observar el punto más bajo es el hiper-parámetro 33 de la red neuronal LSTM de 1, 2 y 3 día de entrenamiento. Dentro de esta red, la configuración del hiper-parámetro número 33, es (8,1,100) cuyo valor de RMSE más cercano a cero es de 1 día de entrenamiento (grupo 1 ó *first group*) que se puede encontrar en la Tabla C.1.

La métrica de error MAE, se calcula como un promedio de diferencias absolutas entre los valores objetivo y las predicciones. MAE es una puntuación lineal, lo que significa que todas las diferencias individuales se ponderan por igual en el promedio. Desde el punto de vista de la interpretación, la métrica MAE es preferible, dado que RMSE tiene la ventaja de penalizar más los errores grandes (valores atípicos), por lo que centrándose en el límite superior, significa que RMSE tiene una tendencia a ser cada vez más grande en número que MAE, a medida que aumenta el tamaño de la muestra de prueba. Es decir, como los datos que se analizan representan una serie de tiempo, en la mayoría de las veces no muestra valores atípicos, por lo que se prefiere graficar esta métrica.

En la Fig. 4.6 se muestra los resultados de la métrica MAE para las redes neuronales RNN, LSTM y GRU en sus 1, 2 y 3 días de entrenamiento. En el eje x se representa el número de configuración de los hiper-parámetros, ya conocidos por el gráfico anterior, y en el eje y se representa el error absoluto medio (MAE).

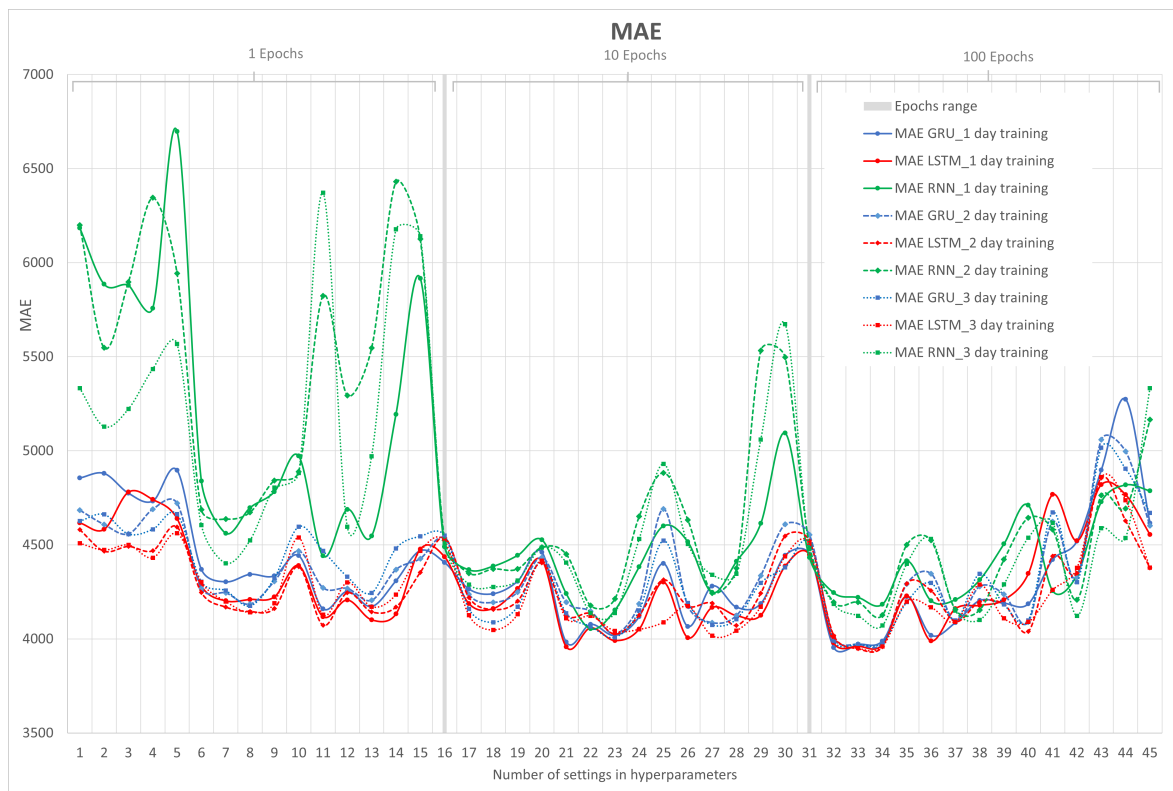


Figura 4.6. Métrica MAE para redes RNN, LSTM y GRU.

Se puede destacar que las curvas de la Fig. 4.6 se parecen al resultado de la métrica RMSE con algunas pequeñas diferencias. Se resalta que la red neuronal RNN en sus 3 días de entrenamiento, no tiene un buen *performance*, al igual que la métrica RMSE. Finalmente el valor RMSE de error más cercano a cero, es la red neuronal LSTM de 2 días de entrenamiento (grupo

2 ó *second group*) con su configuración de hiper-parámetros (8,1,100) que se puede encontrar en la Tabla C.2.

Dado que MAPE es un desarrollo posterior del cálculo MAE, existe similitud entre ambas métricas. Ambas no son sensibles a los valores atípicos ya que usan la diferencia absoluta. MAPE es más comprensible que MAE para el usuario final, debido a que el valor del error se encuentra en términos de porcentaje.

En la Fig. 4.7 se muestra los resultados de la métrica MAPE para las redes neuronales RNN, LSTM y GRU en sus 1, 2 y 3 días de entrenamiento. En el eje x se representa el número de configuración de los hiper-parámetros, y en el eje y se representa el error porcentual absoluto medio (MAPE).

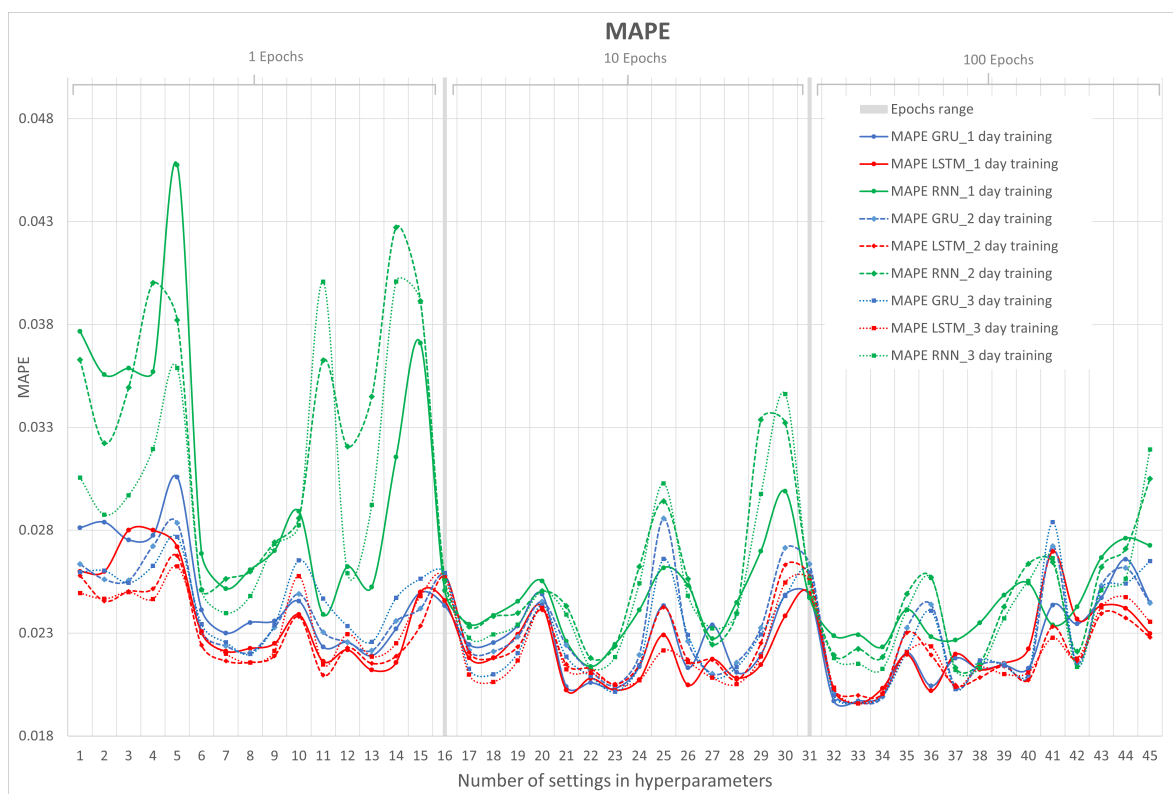


Figura 4.7. Métrica MAPE para redes RNN, LSTM y GRU.

Aquí se representa el error en términos porcentuales, lo que nos hace pensar que se están trabajando con errores mínimos cercanos a 1.9%, lo que se traduce que la predicción en las 3 redes neuronales son de un buen *performance*. Ahora, en la Fig. 4.7, se puede apreciar que con 100 épocas es donde las 3 redes predicen con menos error y que la red neuronal con buenos resultados continua siendo LSTM.

En la Fig. 4.8 se muestra los resultados de la métrica tiempo computacional o *lapsed time* para las redes neuronales RNN, LSTM y GRU en sus 1, 2 y 3 días de entrenamiento. En el eje x

se representa el número de configuración de los hiper-parámetros, y en el eje y se representa el tiempo transcurrido o *lapsed time*).

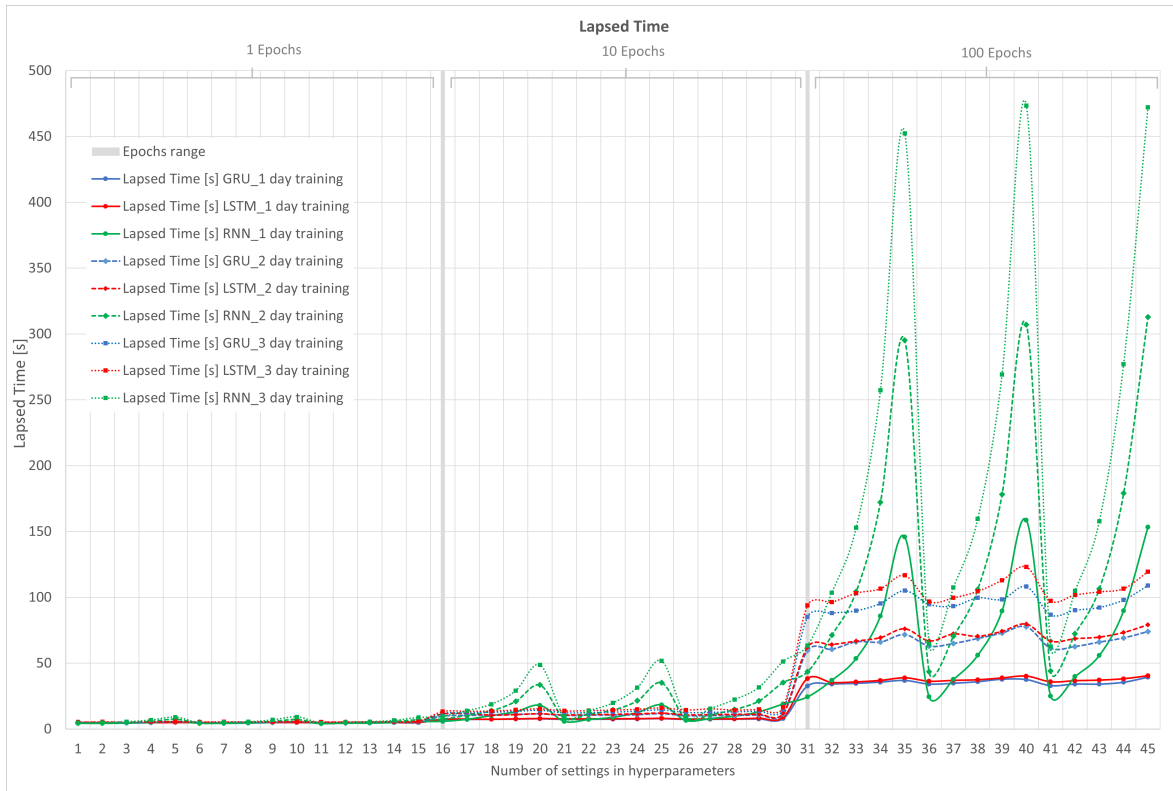


Figura 4.8. Métrica tiempo computacional o *lapsed time* para redes RNN, LSTM y GRU.

Uno de los factores preponderantes es el tiempo que demora la red neuronal en predecir el próximo valor. Este tiempo debe ser menor que el tiempo del valor de *input* del sistema de monitoreo, el cual se estableció en 5 minutos (300 segundos). En la Fig. 4.8, se visualiza claramente que a medida que aumenta las épocas, estas son determinantes a la hora del cálculo computacional. Notar que estos valores varían de acuerdo al sistema computacional. En la tabla 3.1 se muestra el equipamiento utilizado. Para este caso se utiliza tarjeta gráfica GPU para acelerar los cálculos vectoriales del gradiente de cada unas de las redes neuronales.

A parte de las épocas, el número de días de entrenamiento hace aumentar el tiempo transcurrido. Esto se visualiza claramente en donde los resultados de las métricas RMSE, MAE, MAPE son más cercanos a cero, es decir, centrado en el valor de 100 épocas.

Otro punto a resaltar es la estabilidad en términos de tiempo computacional que posee las redes neuronales LSTM y GRU, el cual no varía significativamente si se aumenta el número de *lags*, como sí lo realiza la red neuronal RNN, que aumenta el tiempo en forma exponencialmente.

Finalmente en la métrica tiempo computacional o *lapsed time*, sería incorrecto si la elección fuera el valor más cercano a cero, dado que tiene que ser acompañado por las métricas que se

vieron anteriormente. Ahora, se sabe que los tiempos computacionales mayores a 300 segundos no pueden ser seleccionados.

Recapitulando, en la Tabla 4.3 se muestra a continuación un resumen de los resultados que contienen las mejores métricas de las redes neuronales seleccionadas.

Tabla 4.3. Resumen de resultados para RNN, LSTM y GRU

Type	Group	Training	TimeStep (Lags)	Number Neurons	Epochs	RMSE	MAE	MAPE	Lapsed Time [s]
RNN	First	1 day	4	10	10	5711.161	4054.292	0.02137	7.177
RNN	Second	2 days	16	1	100	5728.889	4126.839	0.02184	172.208
RNN	Third	3 days	16	1	100	5679.286	4070.981	0.02126	257.146
LSTM	First	1 day	8	1	100	5573.399	3961.516	0.01959	35.745
LSTM	Second	2 days	8	1	100	5581.399	3947.681	0.01998	66.765
LSTM	Third	3 days	8	1	100	5585.884	3950.845	0.01958	103.195
GRU	First	1 day	4	1	100	5612.749	3953.799	0.01970	34.093
GRU	Second	2 days	8	1	100	5600.504	3972.234	0.01962	66.179
GRU	Third	3 days	8	1	100	5600.595	3971.039	0.01962	89.822

Los resultados indican que las predicciones de la red neuronal LSTM son considerablemente más precisas que las redes RNN y GRU. Si se visualiza el número de días de entrenamiento que se le esta entregando a cada red neuronal, en RNN al aumentar estos días, se incrementa considerablemente el tiempo de cálculo. En el caso de LSTM hay un empeoramiento de las métricas RMSE, MAE, MAPE y a su vez un aumento en el tiempo computacional. Para GRU se comporta similar a LSTM.

Por otro lado, se puede observar que los resultados de las métricas RMSE, MAE, MAPE en LSTM son más consistentes que en las redes RNN y GRU, debido a que presentan en distintos días de entrenamientos la misma configuración de hiper-parámetros (8 entradas, 1 capa oculta y 100 épocas).

La redes neuronales LSTM y GRU sorprenden dado que aún teniendo una estructura compleja sobre la red RNN, presenta tiempos computacionales bajos al ajustar los hiper-parámetros a valores altos. La mejor combinación de la red neuronal y días de entrenamiento a elegir, es la red LSTM con 1 día de entrenamiento con una configuración de 8 *lags*, 1 neurona y 100 épocas, como se muestra en la Tabla 4.3.

4.2.2 Resultados simulación red neuronal OS-ELM

Para la red OS-ELM se utiliza el mismo método que las redes neuronales RNN, LSTM y GRU, el cual consta de variar el valor de los hiper-parámetros. Se realizan las simulaciones variando 2 parámetros: número de neuronas en la capa oculta y valor de olvido (*Forgetting Factor*). El valor que tomará cada hiper-parámetro será el siguiente:

- Número de neuronas: 10, 110, 210, 310, 410, 510, 610, 710, 810, 910, 1010, 1110, 1210, 1310, 1410, 1510, 1610, 1710, 1810 y 1910

- Valor de olvido (*Forgetting Factor*): 0.9, 0.95, 0.99 y 1.00.

Una vez que se varié cada hiper-parámetro, se obtendrá los resultados de las métricas RMSE, MAE, MAPE y tiempo computacional.

En la Fig. 4.9 se muestra los resultados de cuatro gráficos de la métrica RMSE variando el factor de olvido. En el eje x se representa el número de neuronas. En el eje y se representa el error cuadrático medio (RMSE).

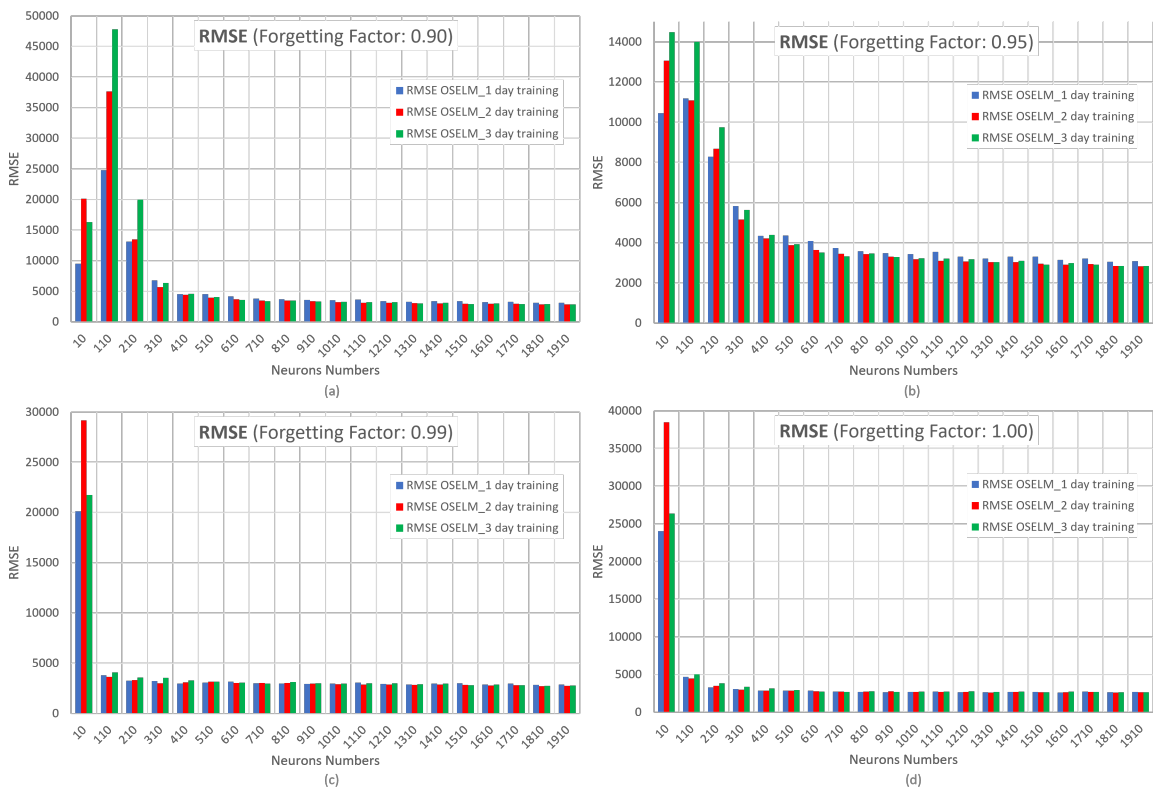


Figura 4.9. Métrica RMSE para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00

Se visualiza que para distintos valores de días de entrenamiento de la red neuronal OS-ELM, la métrica RMSE puede variar en valores de 50000 a 2700, por lo que refleja que

el comportamiento de esta red es más sensible a la variación de sus hiper-parámetros en comparación a las redes RNN, LSTM y GRU. La tabulación de los resultados se presentan en los anexos E. Recordar que si el factor de olvido es 1.00, significa que la red OS-ELM no olvida nada. El factor de olvido permite olvidar continuamente los datos de entrada obsoletos en el proceso de entrenamiento, para reducir su mal efecto al siguiente aprendizaje.

Cuando el factor de olvido es igual a 0.90, RMSE es máximo cuando se configuran 110 neuronas. Si el factor de olvido es igual a 0.95, el error disminuye a medida que aumenta el número de nodos ocultos, y converge después de 410 neuronas. Cuando el factor de olvido es 0.99 e igual a 1.00, en 10 neuronas la red neuronal presenta un error enorme, el cual disminuye bruscamente al aumentar el número de neuronas. Además, se visualiza que al momento de aumentar el factor de olvido y el número neuronas, la red OS-ELM no presenta mayor diferencia para los diferentes días (1, 2 y 3) que se le están dando como *input* para datos de entrenamiento.

En la Fig. 4.10 se muestra los resultados de la métrica MAE variando el factor de olvido. En el eje *x* se representa el número de neuronas. En el eje *y* se representa el error absoluto medio (MAE).

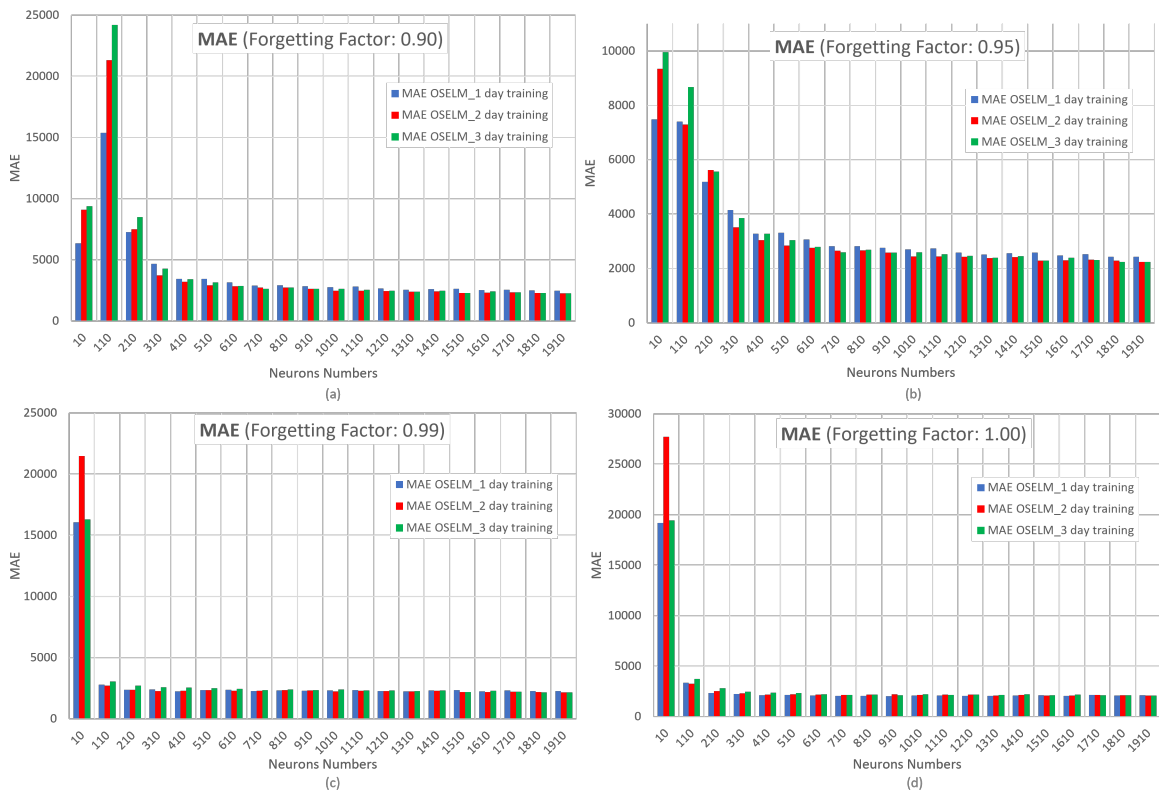


Figura 4.10. Métrica MAE para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00

Estos resultados muestran el mismo comportamiento ya indicado en la métrica RMSE.

Recordar que MAE es más robusto para los valores atípicos y no penaliza los errores tan extremadamente como RMSE. Es debido a eso que se visualiza la red OS-ELM entrenada a 3 días (barra de color verde) en la Fig. 4.10 con menor error que en la Fig. 4.9. Esta conducta se debe a que presenta valores extremos.

En la Fig. 4.11 se muestra los resultados de la métrica MAPE variando el factor de olvido. En el eje x se representa el número de neuronas. En el eje y se representa el error porcentual absoluto medio (MAPE).



Figura 4.11. Métrica MAPE para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00

Se visualiza en la Fig. 4.11 que en términos porcentuales, influye el *input* de datos a la red OS-ELM, destacándose el entrenamiento de 1 día frente a los otros días, dado que es el que presenta menor error. También se destaca la convergencia de esta red pasado las 410 neuronas con un factor de olvido de 0.95.

Ahora bien, en la red OS-ELM puede ver sobre-ajuste cuando se aumentan el número de neuronas y el factor de olvido es mayor a 0.95. El sobre-ajuste es un comportamiento de las redes neuronales no deseado que se produce cuando el modelo de aprendizaje automático proporciona predicciones precisas para los datos de entrenamiento, pero no para los nuevos datos. Si se tiene un factor de olvido cercano a 1.00, no olvidarán los datos anteriores y puede provocar que cuando haya un nuevo comportamiento la red neuronal pueda proporcionar predicciones inexactas.

En la Fig. 4.12 se muestra los resultados de la métrica tiempo computacional o *lapsed time* para las red neuronal OS-ELM en sus 1, 2 y 3 días de entrenamiento. En el eje *x* se representa el número de neuronas, y en el eje *y* se representa el tiempo transcurrido o *lapsed time*).

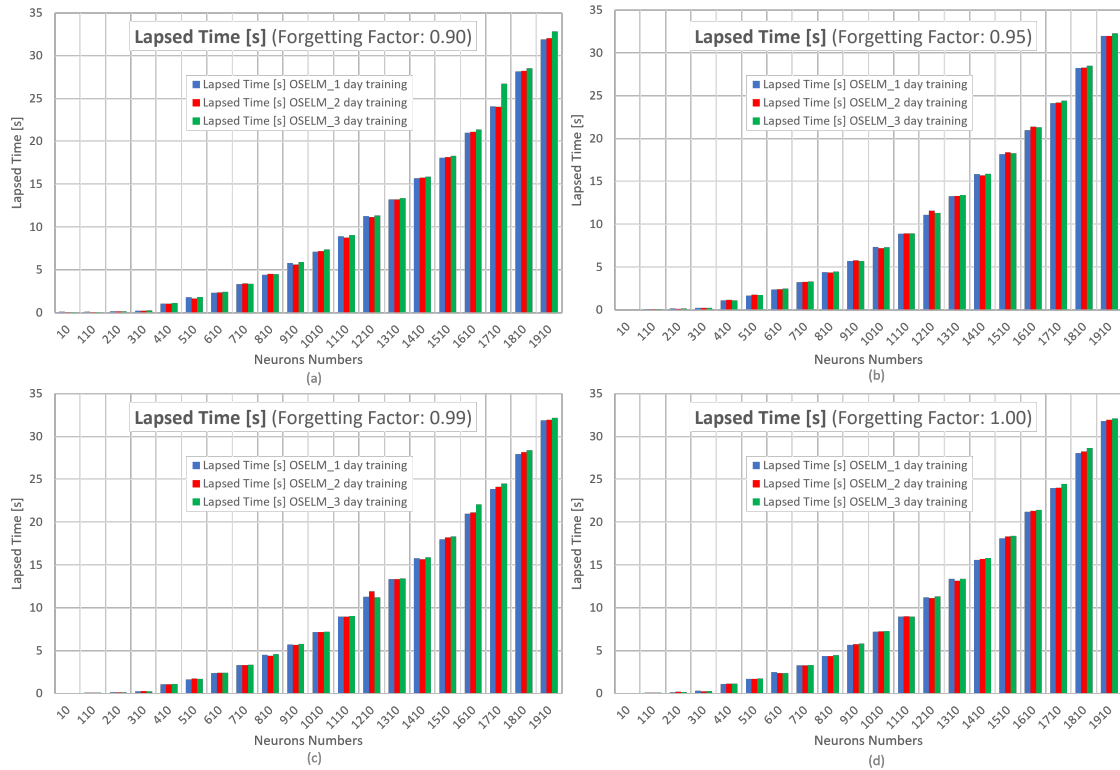


Figura 4.12. Métrica tiempo computacional o *lapsed time* para red OS-ELM con valor de olvido en (a) 0.90, (b) 0.95, (c) 0.99 y (d) 1.00

Se observa de forma muy clara que los tiempos computacionales en el factor de olvido 0.90, 0.95, 0.99 y 1.00 aumentan de forma considerable con mayor número de neuronas. Es por ello, que el criterio de elección de la mejor configuración de hiper-parámetros y días de entrenamiento para la red OS-ELM será en función del tiempo y contemplando un factor de olvido de 0.95 que permita tener posible variaciones de tráfico en la red, para no caer en un sobre-ajuste de acuerdo a lo visto en la Fig. 4.11.

Se presenta en Tabla 4.4 el resumen de los mejores resultados por cada día de entrenamiento de la red OS-ELM con un factor de olvido de 0.95 y 410 en número de neuronas por la métrica de tiempo computacional y convergencia en dicha cantidad de neuronas.

Tabla 4.4. Resumen de resultados para OS-ELM

Type	Group	Training	Forgetting Factor	Number Neurons	RMSE	MAE	MAPE	Lapsed Time [s]
OS-ELM	First	1 day	0.95	410	4336.068	3273.084	0.01761	1.128
OS-ELM	Second	2 days	0.95	410	4221.912	3037.816	0.01642	1.182
OS-ELM	Third	3 days	0.95	410	4384.103	3276.136	0.01778	1.113

Al comparar las métricas RMSE, MAE, MAPE se visualiza que las que presentan un mejor rendimiento es la que posee 2 días de entrenamiento. Con respecto a los tiempos computacionales están todos alrededor de los 1.2 segundos, por lo que la elección final es la red neuronal OS-ELM con 2 días de entrada.

4.2.3 Elección final de la red neuronal

En la Tabla 4.5 se muestra la comparación de las cuatro redes neuronales mejores calificadas por las métricas RMSE, MAE, MAPE en cada uno de los grupos de entrenamiento. Para efectos comparativos, se coloca como referencia a la peor de las redes correspondiente a RNN. Los resultados son determinantes, en términos de predicción comenzando con la métrica RMSE, las redes neuronales OS-ELM, LSTM y GRU superan a RNN en un 26%, 2% y un 1%, respectivamente. Para la métrica MAE, las redes OS-ELM, LSTM y GRU superan a RNN en un 25%, 3% y un 2%, respectivamente y para la medida MAPE, las redes OS-ELM, LSTM y GRU superan a RNN en un 23%, 6% y un 8%, respectivamente.

Finalmente en términos de tiempo computacional OS-ELM, LSTM y GRU superan a RNN en un factor de 217.6, 7.2 y 3.9 veces, respectivamente.

Tabla 4.5. Comparación de predicción de tráfico de las redes neuronales

Type	Training	RMSE	% RMSE	MAE	% MAE	MAPE	% MAPE	Lapsed Time [s]	times
OS-ELM	2 days	4221.912	26 %	3037.816	25 %	0.01642	23 %	1.182	217.6x
LSTM	1 day	5573.399	2 %	3961.516	3 %	0.01999	6 %	35.745	7.2x
GRU	2 days	5600.504	1 %	3972.234	2 %	0.01962	8 %	66.179	3.9x
RNN	3 days	5679.286	-	4070.981	-	0.02126	-	257.146	-

Para efectos de revisar la comparación de las dos mejores redes diferenciando su técnica de optimización. En la Tabla 4.6 se muestra a LSTM que se basa en la mejor red de optimización del descenso del gradiente y OS-ELM que utiliza la pseudo-inversa de Moore-Penrose. Se muestra los porcentajes de las métricas RMSE, MAE, MAPE tomando como referencia a la peor de ellas. En términos de predicción, OS-ELM supera a LSTM en un 24% en RMSE, 23% en MAE, y 16% en MAPE. En términos de tiempo computacional, OS-ELM tiene un factor de 30.2 veces más rápido que LSTM.

Tabla 4.6. Métricas de predicción de tráfico para las mejores redes de LSTM y OS-ELM

Type	Training	RMSE	% RMSE	MAE	% MAE	MAPE	% MAPE	Lapsed Time [s]	times
OS-ELM	2 days	4221.912	24 %	3037.816	23 %	0.01642	18 %	1.182	30.2x
LSTM	1 day	5573.399	-	3961.516	-	0.01999	-	35.745	-

En la Fig. 4.13 se muestra las dos redes neuronales OS-ELM y LSTM indicadas en la Tabla 4.6. Estas presentan la predicción de la carga de tráfico del *router* CDN cada cinco minutos en un día. En el eje x se representa el *Bit Rate* medida en Gbps, y en el eje y se representa el tiempo indicado en horas, el cual cada punto son 5 minutos (300 segundos).

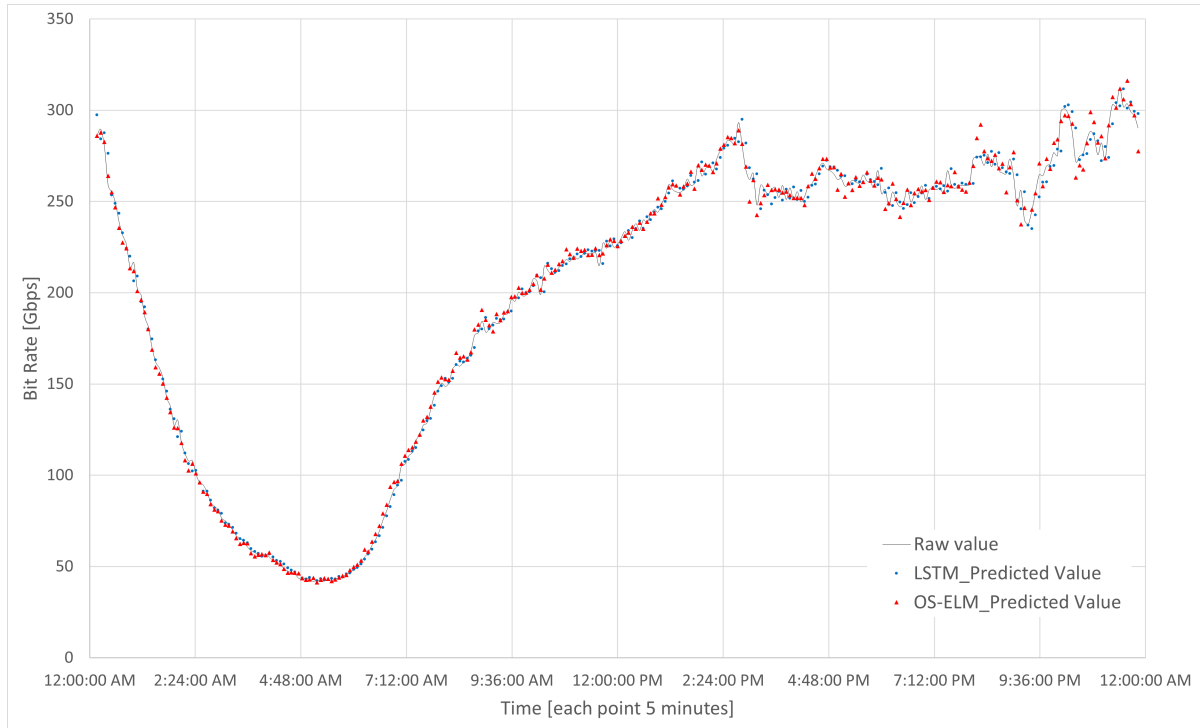


Figura 4.13. Comparación de predicción de tráfico de las mejores redes: LSTM y OS-ELM

La elección final de la red neuronal secuencial OS-ELM es clara, dado los resultados obtenidos.

4.3 RESULTADOS EN ALGORITMOS DE EFICIENCIA ENERGÉTICA

A continuación se mostrará los rendimientos de los algoritmos de optimización de la eficiencia energética propuestos e indicado en la Fig. 3.8. Estos algoritmos se compararán con el caso base, es decir, con el sistema operando en las condiciones actuales. Recordar que los algoritmos se encargan de encender o apagar los puertos correspondientes en un lado el *router* CDN y la conexión de enlace para reducir el consumo de energía de la red mientras se satisface la demanda de tráfico. El primer algoritmo está basado en el tráfico pasado con un valor de umbral del 100 %, llamado algoritmo basado en umbrales. El segundo algoritmo está basado en la predicción de tráfico realizada por la red neuronal seleccionada en el capítulo anterior, llamado algoritmo basado en predicción. Ambos ya explicados en el capítulo 3.2.1.

Los resultados de las simulaciones se presentan en las figuras 4.14, 4.15, 4.16 y 4.17, que fueron efectuadas para el día de testeo presentado en la Fig. 4.4.

En la Fig. 4.14 se muestra los resultados del ahorro acumulado en el día de testeo, para el caso base, algoritmo basado en umbral y de predicción. En el eje x se representa el tiempo en horas. En el eje y se representa el ahorro acumulado, medido W/h .



Figura 4.14. Ahorro acumulado en un día para caso base, umbral y predicción

Como se puede observar en la Fig. 4.14 el ahorro efectuado por el algoritmo basado en predicción es de 4829.58 W/h día, es decir 4.83 kW/h día. Por otro lado, considerando el algoritmo basado en umbrales, el ahorro es de 2.68 kW/h día. Obviamente para el caso base no existe

ahorro de energía. La potencia del *router* es de 1656 W calculado en la ecuación 4.2, por lo que la energía consumida es de 39.6 kW/h día. Por lo tanto, el ahorro por día es del 12.2%, considerando sólo un lado del equipamiento (correspondiente al de servicios), el consumo del chasis y el de las tarjetas.

Por otro lado, el máximo ahorro que se puede realizar, no utilizando ningún puerto del *router* es de 7 kW/h día, considerando 36.5 W de potencia por puerto, multiplicado por los 8 puertos que se tienen presente en el BE. Si se realiza la comparación del ahorro de predicción (4.83 kW/h día) frente al consumo total de los puertos (7 kW/h día), es casi un 70%.

En la Fig. 4.15 se muestra el número de puertos activos basado en encender (*turn ON*) o apagar (*turn OFF*) en el día de testeo, para el caso base, algoritmo basado en umbral y de predicción. En el eje *x* se representa el tiempo en horas. En el eje *y* se representa el número de puertos activos.

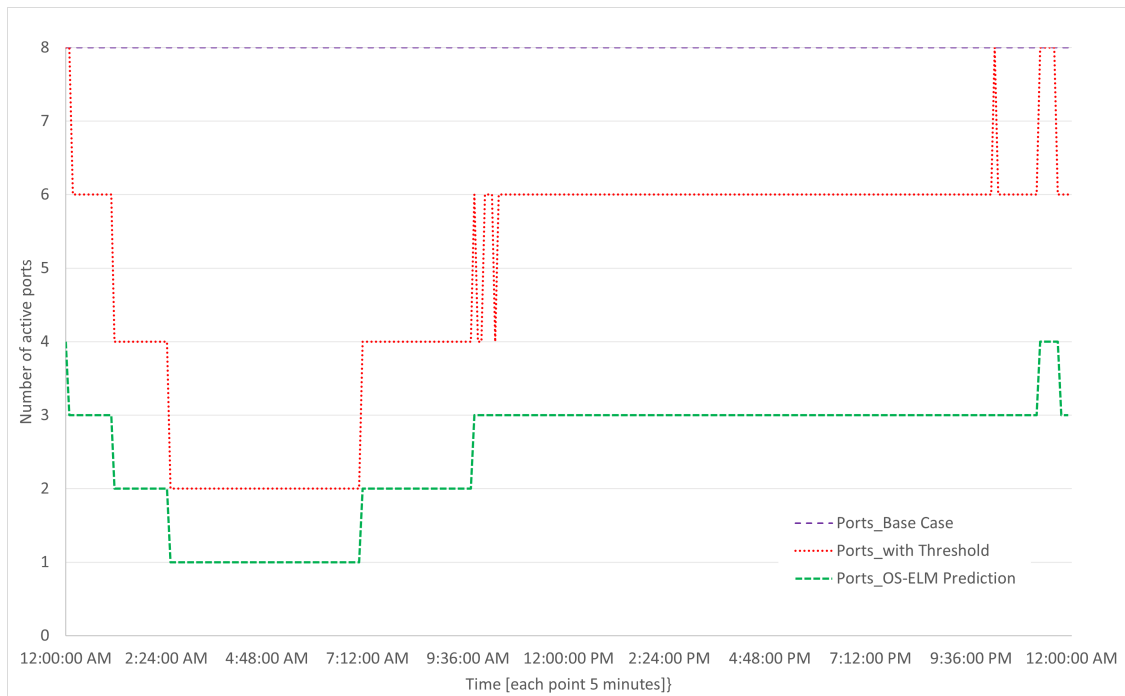


Figura 4.15. Número de puertos activos en un día para caso base, umbral y predicción

Se visualiza en la Fig. 4.15 que para el caso base en todo momento se encuentran todos los puertos activos, y que para ambos algoritmos comienza a descender de forma escalonada la utilización de puertos a contar de las 2:00 AM hasta llegando al punto más bajo dentro del rango de 3:00 AM a 7:00 AM, esto es debido por la baja de utilización del tráfico en las horas de la noche. Notar que en el caso del algoritmo basado en umbrales, en un punto de la mañana existe una alta variación, no presentada en el algoritmo de predicción, el cual se observa que es más estable.

Con respecto a la capacidad disponible del BE, la Fig. 4.16 muestra que en el caso base tiene una alta disponibilidad de capacidad o si se quiere ver de otra forma, una baja utilización, esto es debido a que la red está configurada de tal manera que pueda absorber un sitio de iguales características en caso de falla, por concepto de respaldo. En el caso de algoritmo basado en umbrales, aún queda capacidad disponible ante cualquier situación o evento. Ya en el algoritmo basado en predicción, de alguna manera se pierde esta capacidad de respaldo, frente al beneficio de ahorro de energía, presentado en este estudio.

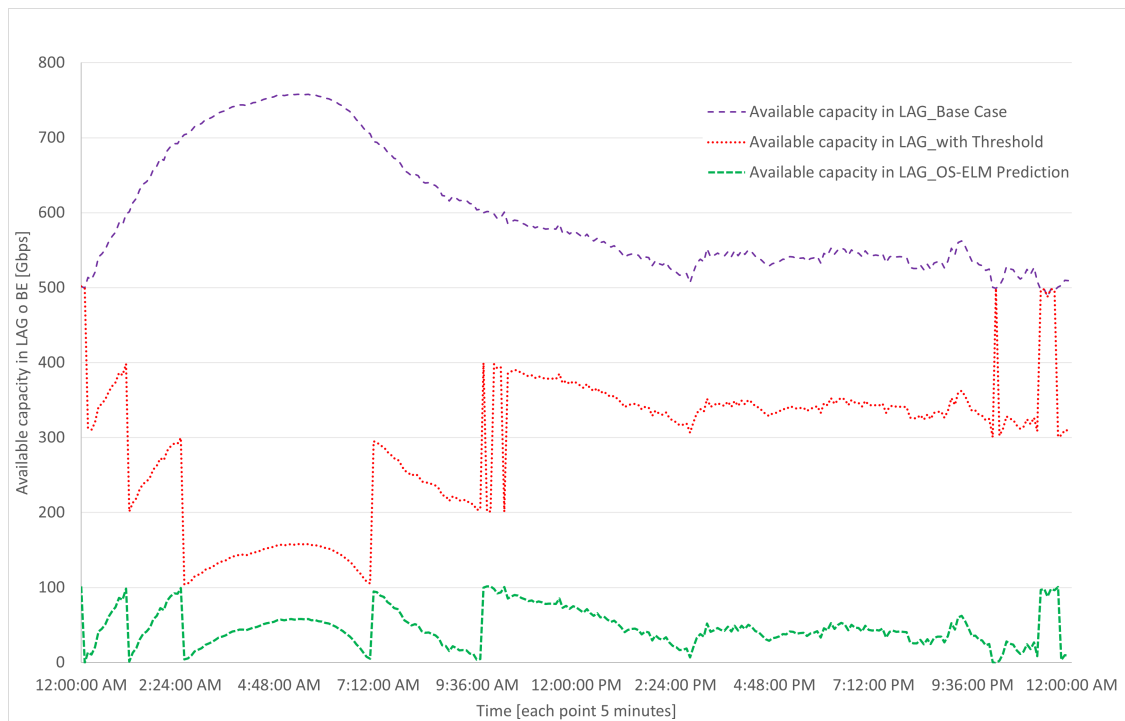


Figura 4.16. Capacidad disponible en el BE o LAG en un día para caso base, umbral y predicción

En esta simulación, en el algoritmo basado en predicción hay puntos que por efecto de una predicción no tan precisa, se pierde tráfico en dos puntos, 0.3 Gbps a las 12:05 AM y 0.79 Gbps a las 10:10 PM, como se muestra en la Fig. 4.17.

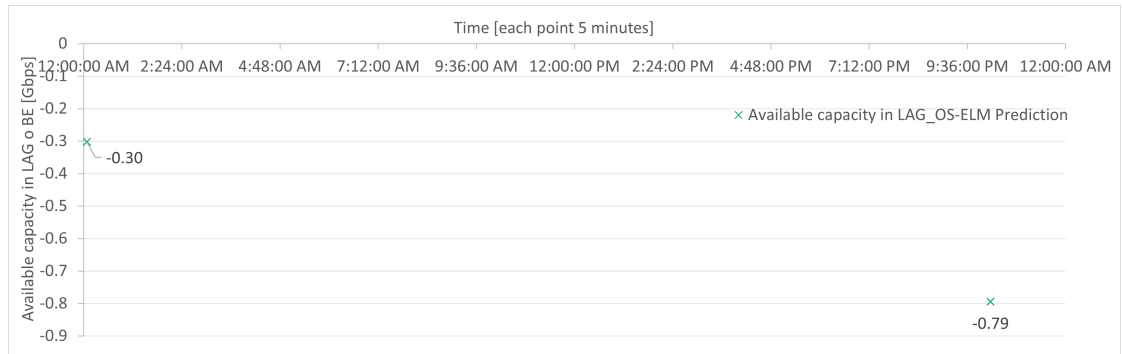


Figura 4.17. Perdida de capacidad en el BE o LAG en un día para caso base, umbral y predicción

Para efectos de proporcionalidad es bajo dado que se encuentran del orden del 0.2% del tráfico total (400 Gbps), pero para una TSO es crítico, dado que habría una pérdida de paquetes en un momento de la red, con intentos de re-conexión. Una solución a esto es contemplar una factor de seguridad a la predicción.

CAPÍTULO 5

CONCLUSIONES

En el presente capítulo se presentan las conclusiones generales del trabajo de tesis, las perspectivas futuras que se pueden seguir desarrollando en este tema y los productos desarrollados a partir de este estudio.

5.1 CONCLUSIÓN GENERAL

Este trabajo describe un método sistemático para resolver problemas complejos que requieren predicciones precisas. La utilización de redes neuronales como herramienta principal para la predicción permite una alta precisión y adaptabilidad a diferentes tipos de datos. Además, el énfasis en la eficiencia energética subraya la importancia de reducir el consumo de energía y descubrir formas de optimizar la utilización de recursos.

En el caso de estudio, se implementó una solución para abordar el problema de la eficiencia energética en los centros de datos de los Operadores de Servicios de Telecomunicaciones (TSO). Para lograr esto, se compararon cuatro redes neuronales recurrentes y secuenciales, permitiendo realizar predicciones cada 5 minutos utilizando ventanas deslizantes e hiper-parámetros. OS-ELM es la mejor red neuronal de alta precisión; en términos de predicción, las redes OS-ELM, LSTM y GRU superan a RNN en un 26 %, 2 % y 1 % para la métrica RMSE, respectivamente. Un 25 %, 3 % y 2 % para la métrica MAE, respectivamente. Un 23 %, 6 % y 8 % para la métrica MAPE, respectivamente.

En términos de tiempo computacional OS-ELM, LSTM y GRU superan a RNN en un factor de 217.6, 7.2 y 3.9 veces, respectivamente. Para todas ellas, se obtienen tiempos de ejecución

en predicción, menor al tiempo de que el sistema colecta los datos (menor a 300 segundos). Para este efecto, OS-ELM con aproximadamente 1.2 segundos de ejecución, debido a su simplicidad en su estructura y al no tener que ocupar minimización del gradiente para encontrar su óptimo.

Las simulaciones se han aplicado al tráfico real de un proveedor de servicios de telecomunicaciones, lo que ofrece una solución real para la eficiencia energética y el ahorro de energía que se puede aplicar no solo en la parte central (*Core*), sino también en las redes de agregación, donde hay una gran cantidad de BE o LAG y se pueden lograr importantes ahorros de energía.

Con respecto al caso base (condiciones actuales), el algoritmo basado en umbrales produjo un ahorro energético del 6.8% y el algoritmo basado en predicciones produjo un ahorro energético del 12.2% por día. Cabe señalar que en esta simulación solo se considera un lado del equipo para el ahorro de energía (lado del cliente o del servicio). Si se extrapola a una gran cantidad de equipos, representaría un ahorro de sustancial.

La metodología diseñada y aplicada a este trabajo de grado, tanto para la elección de red neuronal y de eficiencia energética se puede aplicar o extrapolar a diferentes ámbitos. No sólo en el de sector de Telecomunicaciones, si no que también como previsión de carga y desconexión de centrales de generación de energía en el sistema eléctrico en el sector de Energía, sistemas de logística por predicción de almacenamiento y ahorro de infraestructuras dentro del sector Retail, entre otros. La versatilidad y adaptabilidad del enfoque lo convierten en una solución prometedora para una amplia variedad de problemas de predicción. A medida que la tecnología y los datos continúan avanzando, esta metodología puede ser aún más desarrollada y mejorada para proporcionar predicciones aún más precisas y promover la innovación en numerosas industrias. El potencial de futuras aplicaciones y su impacto en una variedad de campos destacan la importancia de esta metodología y la necesidad de investigación adicional en este campo.

5.2 PERSPECTIVAS FUTURAS

A partir de los resultados obtenidos en el presente trabajo, se sugiere revisar el ahorro de energía no sólo de la utilización de puertos, si no que también el apagado de las tarjetas de servicio o de línea del equipamiento, dado que tienen un mayor consumo. También para aumentar el porcentaje de ahorro completa de la red, se podrían probar otros tipos de escenarios y equipamientos que estén asociados a esta condición de agregación de enlaces. Adicionalmente, simular e implementar la solución de conmutación de tráfico y que el BE pueda soportar la capacidad de la necesidad requerida.

Un tema a contemplar son las posibles pérdidas de datos en algunos momentos de la red, dada a una posible predicción errónea o ráfagas de tráfico (en este caso menos probable que

ocurra, debido a la agregación de tráficos dentro del *Core*), pero una solución sería dar un mayor factor de seguridad o disminuyendo el muestreo de datos. Además de esto, en caso de corte de un sitio o necesidad de redundancia de la red, el sistema tendría que tener un mecanismo de parar el algoritmo de ahorro de energía para poder sostener el tráfico conmutado.

5.3 PRODUCTOS DE INVESTIGACIÓN

A continuación, se muestran los productos de investigación que se han generado a lo largo del desarrollo de este trabajo de tesis:

5.3.1 Publicados (SCOPUS)

- Francisco Rau, Ismael Soto, Pablo Adasme, David Zabala-Blanco, Cesar A. Azurdia–Meza, “Network Traffic Prediction Using Online-Sequential Extreme Learning Machine”, 3rd South American Colloquium on Visible Light Communications, (SACVLC 2021), Toledo, Brazil, 11-12 November, 2021. <https://doi.org/10.1109/SACVLC53127.2021.9652247>
- Francisco Rau, Ismael Soto, David Zabala-Blanco, “Forecasting Mobile Network Traffic based on Deep Learning Networks”, IEEE Latin-American Conference on Communications, (LATINCOM 2021), Santo Domingo, Dominican Republic, 17-19 November, 2021. <https://doi.org/10.1109/LATINCOM53176.2021.9647788>

5.3.2 Bajo revisión (WOS)

- Francisco Rau, Ismael Soto, David Zabala-Blanco, Cesar A. Azurdia-Meza, Muhammad Ijaz, Sunday Ekpo, Sebastian Gutierrez “A Novel Traffic Prediction Method using Machine Learning for Energy Efficiency in Service Provider Networks”, Energy-Efficient Communication Networks and Systems, Sensors 2023.

ACRÓNIMOS

ARIMA Autoregressive Integrated Moving Average.

BE Bundle Ethernet.

BLSTM Bidirectional Long Short-Term Memory.

CDN Content Delivery Network.

CNN Convolutional Neural Networks.

DLHT Dynamic Local Heuristic Threshold.

DNN Deep Neural Network.

DRCN Design of Reliable Communication Networks.

DT Decision Tree.

EEE Energy Efficient Ethernet.

ELM Extreme Learning Machine.

EPC Evolved Packet Core.

FLHT Fixed Local Heuristic Threshold.

GPU Graphics Processing Unit.

GRU Gated Recurrent Units.

IEEE Institute of Electrical and Electronics Engineers.

IGR Internet Gateway Router.

ILP Integer Linear Programming.

IOT Internet of Things.

IP Internet Protocol.

LACP Link Aggregation Control Protocol.

LAG Link Aggregation Groups.

LP Linear Programming.

LPI Low Power Idle.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

MILP Mixed Integer Linear Problem.

OS-ELM Online Sequential Extreme Learning Machine.

OTN Optical Transport Network.

PHY Physical Layer.

RF Random Forest.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

SARIMA Seasonal Autoregressive Integrated Moving Average.

SDN Software Defined Network.

SMA Simple Moving Average.

SNMP Simple Network Management Protocol.

SPA Standby Port Algorithm.

SVM Support Vector Machine.

TQA Two Queuing Algorithm.

TSO Telecom Service Operator.

WDM Wavelength Division Multiplexing.

REFERENCIAS

- [1] K. M. U. Ahmed, M. H. J. Bollen, and M. Alvarez, "A Review of Data Centers Energy Consumption and Reliability Modeling," *IEEE Access*, vol. 9, pp. 152 536–152 563, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9599719/>
- [2] A. Andrae and T. Edler, "On Global Electricity Usage of Communication Technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, apr 2015. [Online]. Available: <http://www.mdpi.com/2078-1547/6/1/117>
- [3] R. R. R. Junior, M. A. M. Vieira, L. F. M. Vieira, and A. A. F. Loureiro, "Intra and inter-flow link aggregation in SDN," *Telecommunication Systems*, vol. 79, no. 1, pp. 95–107, jan 2022. [Online]. Available: <https://doi.org/10.1007/s11235-021-00841-7><https://link.springer.com/10.1007/s11235-021-00841-7>
- [4] B. Addis, A. Capone, G. Carello, L. G. Gianoli, and B. Sanso, "Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 313–325, feb 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6490068/>
- [5] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5550 LNCS, pp. 795–808, 2009.
- [6] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*. New York, NY, USA: ACM, aug 2010, pp. 29–34. [Online]. Available: <https://dl.acm.org/doi/10.1145/1851290.1851297>

- [7] IEEE Standards Association, "IEEE Std 802.1AX™-2020, IEEE Standard for Local and Metropolitan Area Networks—Link Aggregation," *IEEE Std 802.1AX-2020 (Revision of IEEE Std 802.1AS-2014)*, vol. 2020, pp. 1–421, 2020.
- [8] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A Survey of Green Networking Research," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 3–20, 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/5677351/>
- [9] IEEE 802.3az-2010, *Energy Efficient Ethernet*. IEEE, 2010.
- [10] P. Reviriego, K. Christensen, M. Bennett, B. Nordman, and J. A. Maestro, "Energy Efficiency in Ethernet," in *Green Communications*. Chichester, UK: John Wiley & Sons, Ltd, jul 2015, pp. 277–290. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/9781118759257.ch14>
- [11] L. Liu and B. Ramamurthy, "A dynamic local method for bandwidth adaptation in bundle links to conserve energy in core networks," *Optical Switching and Networking*, vol. 10, no. 4, pp. 481–490, nov 2013. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S157342771300009X>
- [12] P. Fondo-Ferreiro, M. Rodríguez-Pérez, M. Fernández-Veiga, and S. Herrería-Alonso, "Matching SDN and Legacy Networking Hardware for Energy Efficiency and Bounded Delay," *Sensors*, vol. 18, no. 11, p. 3915, nov 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/11/3915>
- [13] H. Imaizumi, T. Nagata, G. Kunito, K. Yamazaki, and H. Morikawa, "Power Saving Mechanism Based on Simple Moving Average for 802.3ad Link Aggregation," in *2009 IEEE Globecom Workshops*. IEEE, nov 2009, pp. 1–6. [Online]. Available: <http://ieeexplore.ieee.org/document/5360735/>
- [14] S. Nihale, S. Sharma, L. Parashar, and U. Singh, "Network Traffic Prediction Using Long Short-Term Memory," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, jul 2020, pp. 338–343. [Online]. Available: <https://ieeexplore.ieee.org/document/9156045/>
- [15] F. Rau, I. Soto, and D. Zabala-Blanco, "Forecasting Mobile Network Traffic based on Deep Learning Networks," in *2021 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, nov 2021, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9647788/>
- [16] D. Andreoletti, S. Troia, F. Musumeci, S. Giordano, G. Maier, and M. Tornatore, "Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural

- Networks,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, apr 2019, pp. 246–251. [Online]. Available: <https://ieeexplore.ieee.org/document/8845132/>
- [17] W. Wang, C. Zhou, H. He, W. Wu, W. Zhuang, and X. Shen, “Cellular Traffic Load Prediction with LSTM and Gaussian Process Regression,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE, jun 2020, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9148738/>
- [18] Nan-Ying Liang, Guang-Bin Huang, P. Saratchandran, and N. Sundararajan, “A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks,” *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, nov 2006. [Online]. Available: <http://ieeexplore.ieee.org/document/4012031/>
- [19] F. Rau, I. Soto, P. Adasme, D. Zabala-Blanco, and C. A. Azurdia-Meza, “Network Traffic Prediction Using Online-Sequential Extreme Learning Machine,” in *2021 Third South American Colloquium on Visible Light Communications (SACVLC)*. IEEE, nov 2021, pp. 01–06. [Online]. Available: <https://ieeexplore.ieee.org/document/9652247/>
- [20] R. Singh, H. Kumar, and R. Singla, “An intrusion detection system using network traffic profiling and online sequential extreme learning machine,” *Expert Systems with Applications*, vol. 42, no. 22, pp. 8609–8624, dec 2015. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0957417415004753>
- [21] M. Joshi and T. H. Hadi, “A Review of Network Traffic Analysis and Prediction Techniques,” jul 2015. [Online]. Available: <http://arxiv.org/abs/1507.05722>
- [22] Z. Liu, Z. Zhu, J. Gao, and C. Xu, “Forecast Methods for Time Series Data: A Survey,” *IEEE Access*, vol. 9, pp. 91 896–91 912, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9461796/>
- [23] J. Wu and Y. He, “Prediction of GDP in Time Series Data Based on Neural Network Model,” in *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*. IEEE, may 2021, pp. 20–23. [Online]. Available: <https://ieeexplore.ieee.org/document/9456509/>
- [24] Q. H. Do, T. T. H. Doan, T. V. A. Nguyen, N. T. Duong, and V. V. Linh, “Prediction of Data Traffic in Telecom Networks based on Deep Neural Networks,” *Journal of Computer Science*, vol. 16, pp. 1268–1277, 2020.

- [25] Q. Mao, F. Hu, and Q. Hao, "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2595–2621, 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8382166/>
- [26] Y. Hou, X. Zheng, C. Han, W. Wei, R. Scherer, and D. Polap, "Deep Learning Methods in Short-Term Traffic Prediction: A Survey," *Information Technology and Control*, vol. 51, no. 1, pp. 139–157, mar 2022. [Online]. Available: <https://itc.ktu.lt/index.php/ITC/article/view/29947>
- [27] S.-C. Huang and C.-F. Wu, "Energy Commodity Price Forecasting with Deep Multiple Kernel Learning," *Energies*, vol. 11, no. 11, p. 3029, nov 2018. [Online]. Available: <http://www.mdpi.com/1996-1073/11/11/3029>
- [28] C. Xiao, E. Choi, and J. Sun, "Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review," *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1419–1428, oct 2018. [Online]. Available: <https://doi.org/10.1093/jamia/ocy068https://academic.oup.com/jamia/article/25/10/1419/5035024>
- [29] M. Lepot, J.-B. Aubin, and F. Clemens, "Interpolation in Time Series: An Introductory Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment," *Water*, vol. 9, no. 10, p. 796, oct 2017. [Online]. Available: <http://www.mdpi.com/2073-4441/9/10/796>
- [30] X. Zhang, H. Kuehnelt, and W. De Roeck, "Traffic Noise Prediction Applying Multivariate Bi-Directional Recurrent Neural Network," *Applied Sciences*, vol. 11, no. 6, p. 2714, mar 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/6/2714>
- [31] J. Shin, K. Yeon, S. Kim, M. Sunwoo, and M. Han, "Comparative Study of Markov Chain With Recurrent Neural Network for Short Term Velocity Prediction Implemented on an Embedded System," *IEEE Access*, vol. 9, pp. 24 755–24 767, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9345689/>
- [32] D. Impedovo, V. Dentamaro, G. Pirlo, and L. Sarcinella, "TrafficWave: Generative Deep Learning Architecture for Vehicular Traffic Flow Prediction," *Applied Sciences*, vol. 9, no. 24, p. 5504, dec 2019. [Online]. Available: <https://www.mdpi.com/2076-3417/9/24/5504>
- [33] S. Sha, J. Li, K. Zhang, Z. Yang, Z. Wei, X. Li, and X. Zhu, "RNN-Based Subway Passenger Flow Rolling Prediction," *IEEE Access*, vol. 8, pp. 15 232–15 240, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8951085/>
- [34] C. Zeng, C. Ma, K. Wang, and Z. Cui, "Parking Occupancy Prediction Method Based on Multi Factors and Stacked GRU-LSTM," *IEEE Access*, vol. 10, pp. 47 361–47 370, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9765513/>

- [35] Z. Khan, S. M. Khan, K. Dey, and M. Chowdhury, "Development and Evaluation of Recurrent Neural Network-Based Models for Hourly Traffic Volume and Annual Average Daily Traffic Prediction," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 7, pp. 489–503, jul 2019. [Online]. Available: <https://doi.org/10.1177/0361198119849059><http://journals.sagepub.com/doi/10.1177/0361198119849059>
- [36] K. T. Chui, B. B. Gupta, R. W. Liu, X. Zhang, P. Vasant, and J. J. Thomas, "Extended-Range Prediction Model Using NSGA-III Optimized RNN-GRU-LSTM for Driver Stress and Drowsiness," *Sensors*, vol. 21, no. 19, p. 6412, sep 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/19/6412>
- [37] M. Nguyen, N. Sun, D. C. Alexander, J. Feng, and B. T. Yeo, "Modeling Alzheimer's disease progression using deep recurrent neural networks," in *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. IEEE, jun 2018, pp. 1–4. [Online]. Available: <https://ieeexplore.ieee.org/document/8423955/>
- [38] P. Li, Y. Shi, Y. Xing, C. Liao, M. Yu, C. Guo, and L. Feng, "Intra-Cluster Federated Learning-Based Model Transfer Framework for Traffic Prediction in Core Network," *Electronics*, vol. 11, no. 22, p. 3793, nov 2022. [Online]. Available: <https://www.mdpi.com/2079-9292/11/22/3793>
- [39] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide Cellular Traffic Prediction Based on Densely Connected Convolutional Neural Networks," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1656–1659, aug 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8368274/>
- [40] Y. Fu, S. Wang, C.-X. Wang, X. Hong, and S. McLaughlin, "Artificial Intelligence to Manage Network Traffic of 5G Wireless Networks," *IEEE Network*, vol. 32, no. 6, pp. 58–64, nov 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8553655/>
- [41] D. Zhang, L. Liu, C. Xie, B. Yang, and Q. Liu, "Citywide Cellular Traffic Prediction Based on a Hybrid Spatiotemporal Network," *Algorithms*, vol. 13, no. 1, p. 20, jan 2020. [Online]. Available: <https://www.mdpi.com/1999-4893/13/1/20>
- [42] C.-C. Kao, C.-W. Chang, C.-P. Cho, and J.-Y. Shun, "Deep Learning and Ensemble Learning for Traffic Load Prediction in Real Network," in *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*. IEEE, oct 2020, pp. 36–39. [Online]. Available: <https://ieeexplore.ieee.org/document/9302005/>
- [43] G. L. Santos, P. Rosati, T. Lynn, J. Kelner, D. Sadok, and P. T. Endo, "Predicting short-term mobile Internet traffic from Internet activity using recurrent neural networks," *International Journal of Network Management*, vol. 32, no. 3, p. e2191, may

2022. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2191>
<https://onlinelibrary.wiley.com/doi/10.1002/nem.2191>
- [44] A. Nejadettehad, H. Mahini, and B. Bahrak, "Short-term Demand Forecasting for Online Car-hailing Services Using Recurrent Neural Networks," *Applied Artificial Intelligence*, vol. 34, no. 9, pp. 674–689, jul 2020. [Online]. Available: <https://doi.org/10.1080/08839514.2020.1771522>
<https://www.tandfonline.com/doi/full/10.1080/08839514.2020.1771522>
- [45] B. P. Kumar, K. Hariharan, R. Shanmugam, S. Shriram, and J. Sridhar, "Enabling internet of things in road traffic forecasting with deep learning models," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 5, pp. 6265–6276, sep 2022. [Online]. Available: <https://doi.org/10.3233/JIFS-220230>
<https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/JIFS-220230>
- [46] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, dec 2006. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231206000385>
- [47] J.-M. Park and J.-H. Kim, "Online recurrent extreme learning machine and its application to time-series prediction," in *2017 International Joint Conference on Neural Networks (IJCNN)*, vol. 2017-May. IEEE, may 2017, pp. 1983–1990. [Online]. Available: <http://ieeexplore.ieee.org/document/7966094/>
- [48] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, nov 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231218310294>
- [49] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient Subtyping via Time-Aware LSTM Networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: ACM, aug 2017, pp. 65–74. [Online]. Available: <https://doi.org/10.1145/3097983.3097997>
<https://dl.acm.org/doi/10.1145/3097983.3097997>
- [50] Y. Rubanova, R. T. Chen, and D. Duvenaud, "Latent odes for irregularly-sampled time series (2019)," *arXiv preprint arXiv:1907.03907*, 1907.
- [51] N. Vecoven, D. Ernst, and G. Drion, "A bio-inspired bistable recurrent cell allows for long-lasting memory," *PLOS ONE*, vol. 16, no. 6, p. e0252676, jun 2021. [Online]. Available: <https://doi.org/10.1371/journal.pone.0252676>
<https://dx.plos.org/10.1371/journal.pone.0252676>
- [52] J. Zhou and Z. Huang, "Recover missing sensor data with iterative imputing network," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

- [53] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, jun 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231221003003>
- [54] A. Mahmood, M. L. Mat Kiah, M. Reza Z'Abu, A. N. Qureshi, M. S. S. Kassim, Z. H. Azizul Hasan, J. Kakarla, I. Sadegh Amiri, and S. R. Azzuhri, "Capacity and Frequency Optimization of Wireless Backhaul Network Using Traffic Forecasting," *IEEE Access*, vol. 8, pp. 23 264–23 276, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8974293/>
- [55] S. Ba, I. A. Ouédraogo, and E. Oki, "A power consumption reduction scheme in hose-model networks with bundled links," *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCoM 2013*, pp. 40–45, 2013.
- [56] J. Galán-Jiménez and A. Gazo-Cervero, "Designing energy-efficient link aggregation groups," *Ad Hoc Networks*, vol. 25, no. PB, pp. 595–605, feb 2015. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S157087051400239X>
- [57] M. Rodriguez-Perez, M. Fernandez-Veiga, S. Herreria-Alonso, M. Hmila, and C. Lopez-Garcia, "Optimum Traffic Allocation in Bundled Energy-Efficient Ethernet Links," *IEEE Systems Journal*, vol. 12, no. 1, pp. 593–603, mar 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/7243300/>
- [58] P. Fondo-Ferreiro, M. Rodriguez-Perez, and M. Fernandez-Veiga, "Implementing energy saving algorithms for ethernet link aggregates with ONOS," *2018 5th International Conference on Software Defined Systems, SDS 2018*, pp. 118–125, 2018.
- [59] N. Ramakrishnan and T. Soni, "Network Traffic Prediction Using Recurrent Neural Networks," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, dec 2018, pp. 187–193. [Online]. Available: <https://ieeexplore.ieee.org/document/8614060/>
- [60] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *Computer and information sciences*, pp. 1–9, dec 2014. [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [61] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression," *Neurocomputing*, vol. 311, pp. 126–136, oct 2018. [Online]. Available: <https://doi.org/10.1016/j.neucom.2018.05.057><https://linkinghub.elsevier.com/retrieve/pii/S0925231218306428>

- [62] J. seok Lim, S. Lee, and H. S. Pang, "Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations," *Neural Computing and Applications*, vol. 22, pp. 569–576, 2013.
- [63] L. Jian, F. Gao, P. Ren, Y. Song, and S. Luo, "A Noise-Resilient Online Learning Algorithm for Scene Classification," *Remote Sensing*, vol. 10, no. 11, p. 1836, nov 2018. [Online]. Available: <http://dx.doi.org/10.3390/rs10111836><http://www.mdpi.com/2072-4292/10/11/1836>
- [64] S. Shrivastava, "Cross Validation in Time Series," 2020. [Online]. Available: <https://medium.com/@soumyachess1496>
- [65] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, may 2012. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0020025511006773>
- [66] L. J. Tashman, "Out of Sample Tests of Forecasting Accuracy: An Analysis and Review," *International Journal of Forecasting*, vol. 16, pp. 437–450, 2000.
- [67] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, no. 1, p. 91, dec 2006. [Online]. Available: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-91>
- [68] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Computer and information sciences; Machine Learning*, 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [69] T. Schaul, I. Antonoglou, and D. Silver, "Unit Tests for Stochastic Optimization," *Computer and information sciences*, 2013. [Online]. Available: <https://arxiv.org/abs/1312.6055>
- [70] S. Bock, J. Goppold, and M. Weiß, "An improvement of the convergence proof of the ADAM-Optimizer," *Computer and information sciences*, 2018. [Online]. Available: <https://arxiv.org/abs/1804.10587>
- [71] K. Carling and X. Meng, "Confidence in Heuristic Solutions," *Journal of Global Optimization*, vol. 63, pp. 381–399, 2015.
- [72] Y. Yue, Q. Wang, J. Yao, J. O'Neil, D. Pudvay, and J. Anderson, "400GbE Technology Demonstration Using CFP8 Pluggable Modules," *Applied Sciences*, vol. 8, no. 11, p. 2055, oct 2018. [Online]. Available: <http://www.mdpi.com/2076-3417/8/11/2055>
- [73] W. Zhang, B. G. Bathula, R. K. Sinha, R. Doverspike, P. Magill, A. Raghuram, and G. Choudhury, "Cost Comparison of Alternative Architectures for IP-over-Optical Core

Networks,” *Journal of Network and Systems Management*, vol. 24, no. 3, pp. 607–628, jul 2016. [Online]. Available: <http://link.springer.com/10.1007/s10922-016-9374-z>

- [74] IEEE Standards Association, “IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications-Aggregation of Multiple Link Segments,” *IEEE Std 802.3ad-2000*, pp. 1–184, 2000.
- [75] R.-P. Braun, “100Gbit/s IP-Router and DWDM Transmission Interoperability Field Tests,” in *Photonic Networks, 12 . ITG Symposium*, 2011, pp. 1–3.
- [76] IEEE, “IEEE Std 802.3-2015,” *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)*, pp. 1–4017, 2016.
- [77] IEEE Standards Association, *IEEE Std 802.3bm-2015, IEEE Standard for Ethernet Amendment 3: Physical Layer Specifications and Management Parameters for 40 Gb/s and 100 Gb/s Operation over Fiber Optic Cables*. IEEE, 2015.
- [78] P. Reviriego, J.-A. Hernandez, D. Larrabeiti, and J. A. Maestro, “Burst Transmission in Energy Efficient Ethernet,” *IEEE Internet Computing*, vol. 14, no. 4, pp. 50–57, 2010. [Online]. Available: <http://ieeexplore.ieee.org/document/5432134/>

ANEXO A

EQUIPAMIENTO NE40E-X8A

Basic Configuration Information [\[hide\]](#)

No.	Type	Board Info	Typical Power At 25°C(W)	Power Used At 40°C(W)	Heat Dissipation At 40°C(BTU/Hr)	Power Used At 55°C(W)	Heat Dissipation At 55°C(BTU/Hr)
1	SRU	SRU-480(8GB)-03056097	200.00	220.00	713.77	220.00	713.77
2	SRU	SRU-480(8GB)-03056097	200.00	220.00	713.77	220.00	713.77
3	SFU	SFUI-480-C-03056095	120.00	140.00	454.22	150.00	486.66
4	SFU	SFUI-480-C-03056095	120.00	140.00	454.22	150.00	486.66
5	FAN	FAN-500W-02120866	48.00	500.00	341.52	500.00	341.52
6	FAN	FAN-500W-02120866	48.00	500.00	341.52	500.00	341.52
7	FAN	FAN-500W-02120866	48.00	500.00	341.52	500.00	341.52
Total			784.00	2220.00	3360.54	2240.00	3425.42

Business Card Information [\[hide\]](#)

No.	Slot	Board Info	Application Capacity(G)	Typical Power At 25°C(W)	Power Used At 40°C(W)	Heat Dissipation At 40°C(BTU/Hr)	Power Used At 55°C(W)	Heat Dissipation At 55°C(BTU/Hr)
1	Slot1	LPUF-480-03057844		290.00	385.00	1249.10	400.00	1297.77
2	Slot1-PIC0	PIC-2*100GBase-QSFP28-H-03032YMW		73.00	76.00	246.58	76.00	246.58
3	Slot1-PIC1	P480-24x10G LAN/WAN-SFP+-030305WL		96.40	106.00	343.91	106.00	343.91
4	Slot2	LPUF-480-03057844		290.00	385.00	1249.10	400.00	1297.77
5	Slot2-PIC0	P480-24x10G LAN/WAN-SFP+-H-03032KGV		93.40	103.00	334.18	103.00	334.18
6	Slot2-PIC1	-- Empty Slot --		0.00	0.00	0.00	0.00	0.00
7	Slot3	LPUF-120-12*10GE-SFP+-03054681		201.70	230.20	746.87	230.20	766.33
8	Slot4	LPUF-480-03057844		290.00	385.00	1249.10	400.00	1297.77
9	Slot4-PIC0	PIC-2*100GBase-QSFP28-H-03032YMW		73.00	76.00	246.58	76.00	246.58
10	Slot4-PIC1	PIC-2*100GBase-QSFP28-H-03032YMW		73.00	76.00	246.58	76.00	246.58
11	Slot5	LPUF-400-E-03050EQY		210.00	242.00	785.15	254.00	824.08
12	Slot5-PIC0	2x100GBase-QSFP28-A-03034HHY		33.70	37.50	121.67	38.20	123.94
13	Slot5-PIC1	2x100GBase-QSFP28-A-03034HHY		33.70	37.50	121.67	38.20	123.94
14	Slot6	LPUF-240-03054690		244.00	310.00	1005.77	332.00	1077.15
15	Slot6-PIC0	PIC-1*100G ETH-OTN-CFP2-03032PCD		56.00	56.00	181.69	56.00	181.69
16	Slot6-PIC1	-- Empty Slot --		0.00	0.00	0.00	0.00	0.00
17	Slot7	LPUF-400-E-03050EQY		210.00	242.00	785.15	254.00	824.08
18	Slot7-PIC0	2x100GBase-QSFP28-A-03034HHY		33.70	37.50	121.67	38.20	123.94
19	Slot7-PIC1	2x100GBase-QSFP28-A-03034HHY		33.70	37.50	121.67	38.20	123.94
20	Slot8	LPUF-240-03054690		244.00	310.00	1005.77	332.00	1077.15
21	Slot8-PIC0	PIC-1*100G ETH-OTN-CFP2-03032PCD		56.00	56.00	181.69	56.00	181.69
22	Slot8-PIC1	-- Empty Slot --		0.00	0.00	0.00	0.00	0.00
Total				2623.30	3188.20	10343.90	3310.00	10739.07

Figura A.1. Consumo de *router* CDN NE40E-X8A.

Link: [Huawei support NE40E-X8A](#)

ANEXO B

TABULACIÓN DE VALORES DE SIMULACIÓN PARA RNN

Tabla B.1. First group - RNN results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	8041.010	6184.416	0.03766	4.294
4	1	1	4,1,1	7745.429	5885.487	0.03556	4.243
8	1	1	8,1,1	7698.151	5879.583	0.03587	4.413
16	1	1	16,1,1	7498.964	5757.861	0.03570	4.816
32	1	1	32,1,1	8502.158	6697.261	0.04575	5.555
1	10	1	1,10,1	6523.799	4838.783	0.02687	4.377
4	10	1	4,10,1	6245.870	4561.643	0.02516	4.251
8	10	1	8,10,1	6354.512	4697.402	0.02608	4.455
16	10	1	16,10,1	6421.354	4781.975	0.02701	4.899
32	10	1	32,10,1	6582.429	4972.353	0.02892	5.613
1	50	1	1,50,1	6021.628	4441.591	0.02390	4.095
4	50	1	4,50,1	6283.493	4688.289	0.02623	4.402
8	50	1	8,50,1	6111.695	4547.374	0.02524	4.406
16	50	1	16,50,1	6586.458	5194.036	0.03156	4.840
32	50	1	32,50,1	7269.405	5916.334	0.03710	5.588
1	1	10	1,1,10	6092.491	4490.291	0.02506	5.778
4	1	10	4,1,10	6000.176	4367.495	0.02343	7.369
8	1	10	8,1,10	5998.643	4386.663	0.02386	10.179
16	1	10	16,1,10	6051.919	4443.976	0.02454	13.049
32	1	10	32,1,10	6142.847	4526.946	0.02553	17.963
1	10	10	1,10,10	5835.602	4241.175	0.02260	5.793
4	10	10	4,10,10	5711.161	4054.292	0.02137	7.177
8	10	10	8,10,10	5686.112	4152.902	0.02245	8.916
16	10	10	16,10,10	5930.762	4382.871	0.02412	12.436
32	10	10	32,10,10	6197.371	4600.675	0.02617	18.303
1	50	10	1,50,10	6086.476	4516.026	0.02532	6.472
4	50	10	4,50,10	5911.804	4246.990	0.02273	7.930
8	50	10	8,50,10	5955.428	4412.038	0.02449	9.737
16	50	10	16,50,10	6069.447	4614.068	0.02698	13.011
32	50	10	32,50,10	6583.370	5094.305	0.02989	18.922
1	1	100	1,1,100	6002.425	4432.743	0.02472	24.350
4	1	100	4,1,100	5847.244	4245.517	0.02287	36.942
8	1	100	8,1,100	5803.888	4220.450	0.02292	53.517
16	1	100	16,1,100	5792.359	4184.402	0.02232	85.802
32	1	100	32,1,100	6061.722	4414.835	0.02414	145.852
1	10	100	1,10,100	5771.389	4203.303	0.02283	24.362
4	10	100	4,10,100	5902.660	4208.705	0.02266	37.626
8	10	100	8,10,100	5900.848	4314.305	0.02350	55.971
16	10	100	16,10,100	6089.943	4505.597	0.02484	89.621
32	10	100	32,10,100	6423.240	4710.539	0.02543	158.550
1	50	100	1,50,100	5777.194	4255.769	0.02339	25.056
4	50	100	4,50,100	5969.873	4340.069	0.02428	39.825
8	50	100	8,50,100	6346.355	4729.042	0.02667	55.836
16	50	100	16,50,100	6344.973	4818.684	0.02761	89.855
32	50	100	32,50,100	6333.144	4787.140	0.02726	153.253

Tabla B.2. Second group - RNN results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	8264.423	6198.696	0.03628	4.444
4	1	1	4,1,1	7466.769	5548.018	0.03223	4.488
8	1	1	8,1,1	7817.215	5897.032	0.03493	5.066
16	1	1	16,1,1	8330.440	6344.814	0.04002	5.806
32	1	1	32,1,1	7735.334	5943.193	0.03821	7.533
1	10	1	1,10,1	6397.451	4687.231	0.02511	4.543
4	10	1	4,10,1	6315.344	4637.096	0.02564	4.561
8	10	1	8,10,1	6305.934	4672.947	0.02599	5.101
16	10	1	16,10,1	6445.454	4842.255	0.02741	5.780
32	10	1	32,10,1	6438.671	4888.691	0.02858	7.291
1	50	1	1,50,1	7310.982	5823.423	0.03626	4.300
4	50	1	4,50,1	6819.065	5294.410	0.03207	4.566
8	50	1	8,50,1	7044.288	5546.835	0.03449	4.934
16	50	1	16,50,1	7748.800	6430.560	0.04272	5.751
32	50	1	32,50,1	7480.378	6125.640	0.03914	7.167
1	1	10	1,1,10	6126.284	4524.729	0.02560	7.650
4	1	10	4,1,10	5981.849	4349.155	0.02331	10.520
8	1	10	8,1,10	5973.884	4371.984	0.02382	13.910
16	1	10	16,1,10	5975.165	4371.917	0.02398	21.010
32	1	10	32,1,10	6102.142	4486.834	0.02504	33.442
1	10	10	1,10,10	6144.888	4450.322	0.02431	8.193
4	10	10	4,10,10	5970.575	4177.205	0.02179	11.366
8	10	10	8,10,10	5866.308	4212.578	0.02237	14.408
16	10	10	16,10,10	6211.495	4650.978	0.02623	21.456
32	10	10	32,10,10	6359.569	4882.864	0.02941	35.186
1	50	10	1,50,10	6272.191	4632.629	0.02563	7.876
4	50	10	4,50,10	5980.826	4243.003	0.02244	10.940
8	50	10	8,50,10	6048.965	4375.544	0.02398	14.886
16	50	10	16,50,10	7049.940	5532.115	0.03337	21.298
32	50	10	32,50,10	6899.477	5497.488	0.03322	35.358
1	1	100	1,1,100	6061.380	4498.280	0.02566	43.547
4	1	100	4,1,100	5818.858	4195.087	0.02193	71.291
8	1	100	8,1,100	5797.595	4195.585	0.02223	104.282
16	1	100	16,1,100	5728.889	4126.839	0.02184	172.208
32	1	100	32,1,100	6135.245	4500.233	0.02490	295.226
1	10	100	1,10,100	6116.857	4523.705	0.02571	43.426
4	10	100	4,10,100	6040.680	4155.594	0.02132	70.466
8	10	100	8,10,100	5953.377	4153.981	0.02154	105.968
16	10	100	16,10,100	5992.731	4422.035	0.02428	178.091
32	10	100	32,10,100	6270.852	4643.521	0.02637	307.073
1	50	100	1,50,100	6129.236	4581.464	0.02645	44.117
4	50	100	4,50,100	5938.367	4207.977	0.02211	72.367
8	50	100	8,50,100	6537.277	4763.506	0.02620	106.444
16	50	100	16,50,100	6196.209	4692.807	0.02710	179.049
32	50	100	32,50,100	6644.541	5165.368	0.03050	312.755

Tabla B.3. Third group - RNN results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	7194.319	5331.976	0.03055	4.668
4	1	1	4,1,1	6963.530	5128.606	0.02875	4.873
8	1	1	8,1,1	7026.252	5221.889	0.02969	5.475
16	1	1	16,1,1	7214.087	5433.772	0.03194	6.811
32	1	1	32,1,1	7253.621	5567.343	0.03588	8.976
1	10	1	1,10,1	6308.709	4605.106	0.02508	4.725
4	10	1	4,10,1	6088.770	4402.207	0.02397	5.048
8	10	1	8,10,1	6169.332	4523.705	0.02481	5.452
16	10	1	16,10,1	6381.347	4804.183	0.02733	6.855
32	10	1	32,10,1	6401.480	4881.550	0.02824	9.162
1	50	1	1,50,1	7885.471	6371.299	0.04007	4.471
4	50	1	4,50,1	6206.752	4596.026	0.02591	5.057
8	50	1	8,50,1	6556.461	4970.793	0.02923	5.494
16	50	1	16,50,1	7564.609	6176.705	0.04009	6.593
32	50	1	32,50,1	7481.168	6140.377	0.03910	8.709
1	1	10	1,1,10	6100.248	4505.734	0.02556	9.576
4	1	10	4,1,10	5918.521	4288.998	0.02277	13.833
8	1	10	8,1,10	5886.132	4275.692	0.02293	18.791
16	1	10	16,1,10	5916.945	4310.722	0.02341	29.035
32	1	10	32,1,10	6089.722	4485.436	0.02503	48.671
1	10	10	1,10,10	6110.475	4405.764	0.02390	9.709
4	10	10	4,10,10	5971.927	4138.098	0.02134	13.888
8	10	10	8,10,10	5810.695	4137.337	0.02183	19.777
16	10	10	16,10,10	6088.359	4531.125	0.02542	31.360
32	10	10	32,10,10	6392.863	4930.141	0.03027	51.752
1	50	10	1,50,10	6131.799	4504.953	0.02482	9.734
4	50	10	4,50,10	6062.011	4340.094	0.02322	15.465
8	50	10	8,50,10	5991.916	4345.938	0.02394	22.341
16	50	10	16,50,10	6569.367	5059.891	0.02976	31.518
32	50	10	32,50,10	7055.334	5672.917	0.03461	51.265
1	1	100	1,1,100	6032.104	4477.264	0.02544	63.459
4	1	100	4,1,100	5813.156	4185.606	0.02179	103.505
8	1	100	8,1,100	5731.945	4122.351	0.02151	152.779
16	1	100	16,1,100	5679.286	4070.981	0.02126	257.146
32	1	100	32,1,100	6021.056	4398.973	0.02413	452.392
1	10	100	1,10,100	6131.135	4530.797	0.02569	64.706
4	10	100	4,10,100	6030.218	4150.092	0.02120	107.461
8	10	100	8,10,100	5846.307	4100.779	0.02130	159.554
16	10	100	16,10,100	5837.383	4290.233	0.02371	269.197
32	10	100	32,10,100	6163.707	4538.004	0.02554	473.384
1	50	100	1,50,100	6161.513	4616.692	0.02664	62.573
4	50	100	4,50,100	5899.451	4122.769	0.02136	104.873
8	50	100	8,50,100	6273.308	4588.548	0.02508	157.796
16	50	100	16,50,100	6037.334	4536.079	0.02564	277.025
32	50	100	32,50,100	6775.861	5332.925	0.03192	472.039

ANEXO C

TABULACIÓN DE VALORES DE SIMULACIÓN PARA LSTM

Tabla C.1. First group - LSTM results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	6278.625	4616.917	0.02600	4.852
4	1	1	4,1,1	6246.828	4582.252	0.02600	4.935
8	1	1	8,1,1	6457.482	4780.564	0.02800	4.832
16	1	1	16,1,1	6393.788	4741.358	0.02800	4.816
32	1	1	32,1,1	6277.409	4640.037	0.02719	4.962
1	10	1	1,10,1	5915.114	4282.088	0.02301	4.907
4	10	1	4,10,1	5815.973	4202.451	0.02212	4.865
8	10	1	8,10,1	5809.749	4209.799	0.02226	4.883
16	10	1	16,10,1	5839.847	4223.117	0.02250	4.905
32	10	1	32,10,1	6021.453	4389.524	0.02390	4.884
1	50	1	1,50,1	5738.843	4118.099	0.02163	4.914
4	50	1	4,50,1	5798.812	4206.891	0.02219	4.970
8	50	1	8,50,1	5696.249	4101.376	0.02121	5.104
16	50	1	16,50,1	5756.111	4132.511	0.02157	4.948
32	50	1	32,50,1	6055.832	4477.512	0.02500	4.944
1	1	10	1,1,10	6030.275	4436.444	0.02458	7.478
4	1	10	4,1,10	5816.714	4187.694	0.02180	7.562
8	1	10	8,1,10	5768.011	4161.317	0.02180	7.429
16	1	10	16,1,10	5884.758	4267.433	0.02294	7.745
32	1	10	32,1,10	6059.172	4418.967	0.02431	8.007
1	10	10	1,10,10	5597.255	3957.924	0.02024	7.737
4	10	10	4,10,10	5690.536	4058.700	0.02081	7.769
8	10	10	8,10,10	5604.793	3991.098	0.02023	7.828
16	10	10	16,10,10	5701.377	4051.952	0.02074	7.878
32	10	10	32,10,10	5937.894	4302.242	0.02291	8.109
1	50	10	1,50,10	5720.684	4006.395	0.02048	7.571
4	50	10	4,50,10	5767.160	4167.126	0.02173	7.714
8	50	10	8,50,10	5792.853	4123.933	0.02082	7.727
16	50	10	16,50,10	5763.336	4125.696	0.02147	8.213
32	50	10	32,50,10	6008.060	4386.517	0.02384	8.988
1	1	100	1,1,100	5989.132	4447.379	0.02493	38.309
4	1	100	4,1,100	5622.655	3979.365	0.02022	35.318
8	1	100	8,1,100	5573.399	3961.516	0.01959	35.745
16	1	100	16,1,100	5601.727	3967.866	0.02032	36.849
32	1	100	32,1,100	5946.213	4228.376	0.02196	38.831
1	10	100	1,10,100	5673.150	3989.207	0.02020	36.235
4	10	100	4,10,100	5713.276	4160.416	0.02200	36.890
8	10	100	8,10,100	5927.279	4178.253	0.02120	37.349
16	10	100	16,10,100	5922.645	4207.654	0.02150	38.728
32	10	100	32,10,100	6101.376	4347.116	0.02220	40.124
1	50	100	1,50,100	6599.346	4767.674	0.02700	35.798
4	50	100	4,50,100	6249.300	4521.593	0.02370	36.638
8	50	100	8,50,100	6909.840	4820.470	0.02430	37.081
16	50	100	16,50,100	6576.371	4766.463	0.02420	38.136
32	50	100	32,50,100	6366.802	4555.314	0.02300	40.388

Tabla C.2. Second group - LSTM results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	6242.708	4580.582	0.0258	5.149
4	1	1	4,1,1	6124.999	4466.070	0.0246	5.116
8	1	1	8,1,1	6132.678	4491.354	0.0250	5.157
16	1	1	16,1,1	6088.017	4468.417	0.0252	5.232
32	1	1	32,1,1	6189.000	4594.940	0.0268	5.327
1	10	1	1,10,1	5908.768	4246.514	0.0224	5.073
4	10	1	4,10,1	5788.395	4169.099	0.0217	5.136
8	10	1	8,10,1	5738.395	4145.350	0.0216	5.195
16	10	1	16,10,1	5777.050	4161.851	0.0219	5.305
32	10	1	32,10,1	5995.695	4383.680	0.0238	5.377
1	50	1	1,50,1	5723.640	4072.236	0.0210	5.155
4	50	1	4,50,1	5827.845	4244.855	0.0223	5.211
8	50	1	8,50,1	5723.338	4143.657	0.0215	5.176
16	50	1	16,50,1	5776.457	4168.793	0.0219	5.440
32	50	1	32,50,1	5972.842	4353.326	0.0234	5.942
1	1	10	1,1,10	6140.694	4537.688	0.0258	11.417
4	1	10	4,1,10	5849.655	4220.092	0.0221	11.717
8	1	10	8,1,10	5757.818	4156.135	0.0218	10.545
16	1	10	16,1,10	5815.280	4200.822	0.0223	10.956
32	1	10	32,1,10	6030.587	4404.127	0.0241	11.575
1	10	10	1,10,10	5813.022	4128.869	0.0215	10.684
4	10	10	4,10,10	5758.686	4134.566	0.0213	10.916
8	10	10	8,10,10	5627.983	4025.243	0.0205	11.118
16	10	10	16,10,10	5767.064	4126.484	0.0215	11.515
32	10	10	32,10,10	5826.665	4313.961	0.0243	12.103
1	50	10	1,50,10	5995.720	4174.210	0.0217	10.763
4	50	10	4,50,10	5926.611	4189.583	0.0217	10.923
8	50	10	8,50,10	5734.521	4071.088	0.0207	11.165
16	50	10	16,50,10	5843.860	4242.921	0.0225	11.251
32	50	10	32,50,10	6037.831	4546.040	0.0263	11.966
1	1	100	1,1,100	6073.303	4521.338	0.0258	62.194
4	1	100	4,1,100	5666.535	4016.597	0.0203	64.102
8	1	100	8,1,100	5581.399	3947.681	0.0199	66.765
16	1	100	16,1,100	5590.626	3958.830	0.0201	69.435
32	1	100	32,1,100	5952.715	4293.704	0.0230	76.126
1	10	100	1,10,100	6016.432	4256.494	0.0219	66.919
4	10	100	4,10,100	6000.570	4092.247	0.0205	72.178
8	10	100	8,10,100	6070.580	4197.389	0.0208	70.396
16	10	100	16,10,100	5829.366	4184.102	0.0214	74.220
32	10	100	32,10,100	5705.844	4039.341	0.0207	79.664
1	50	100	1,50,100	6284.524	4439.835	0.0233	66.763
4	50	100	4,50,100	6357.275	4349.593	0.0217	68.606
8	50	100	8,50,100	7075.479	4854.955	0.0239	69.805
16	50	100	16,50,100	6460.692	4627.694	0.0237	73.318
32	50	100	32,50,100	6104.232	4377.700	0.0228	79.177

Tabla C.3. Third group - LSTM results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	6161.156	4508.858	0.0249	5.291
4	1	1	4,1,1	6109.359	4472.690	0.0247	5.417
8	1	1	8,1,1	6124.661	4499.495	0.0250	5.471
16	1	1	16,1,1	6040.620	4431.244	0.0247	5.635
32	1	1	32,1,1	6145.139	4562.047	0.0262	5.712
1	10	1	1,10,1	5945.197	4300.941	0.0231	5.366
4	10	1	4,10,1	5807.818	4198.970	0.0220	5.406
8	10	1	8,10,1	5730.962	4140.903	0.0216	5.472
16	10	1	16,10,1	5799.240	4189.905	0.0221	5.483
32	10	1	32,10,1	6085.750	4539.235	0.0258	5.648
1	50	1	1,50,1	5774.096	4128.657	0.0215	5.379
4	50	1	4,50,1	5869.819	4300.171	0.0230	5.378
8	50	1	8,50,1	5741.143	4170.777	0.0219	5.504
16	50	1	16,50,1	5832.070	4235.896	0.0225	5.591
32	50	1	32,50,1	6030.012	4468.232	0.0248	5.786
1	1	10	1,1,10	6112.032	4520.958	0.0257	13.301
4	1	10	4,1,10	5775.796	4125.667	0.0210	13.422
8	1	10	8,1,10	5661.056	4047.893	0.0206	13.643
16	1	10	16,1,10	5751.102	4130.992	0.0217	14.472
32	1	10	32,1,10	6031.618	4416.048	0.0242	15.356
1	10	10	1,10,10	5806.067	4109.838	0.0212	13.757
4	10	10	4,10,10	5764.339	4122.760	0.0211	13.959
8	10	10	8,10,10	5686.531	4042.560	0.0205	14.392
16	10	10	16,10,10	5690.439	4050.731	0.0207	14.956
32	10	10	32,10,10	5615.942	4088.563	0.0222	16.079
1	50	10	1,50,10	6079.735	4174.435	0.0216	14.411
4	50	10	4,50,10	5751.876	4016.660	0.0208	15.333
8	50	10	8,50,10	5735.563	4043.623	0.0205	14.448
16	50	10	16,50,10	5764.435	4171.048	0.0219	14.931
32	50	10	32,50,10	5924.575	4438.517	0.0255	16.498
1	1	100	1,1,100	6073.468	4511.421	0.0256	93.922
4	1	100	4,1,100	5667.515	4012.328	0.0203	96.341
8	1	100	8,1,100	5585.884	3950.845	0.0196	103.195
16	1	100	16,1,100	5591.259	3960.110	0.0200	106.520
32	1	100	32,1,100	5869.387	4198.889	0.0221	116.640
1	10	100	1,10,100	5804.391	4168.364	0.0224	96.799
4	10	100	4,10,100	6022.752	4090.603	0.0204	99.622
8	10	100	8,10,100	6208.955	4289.021	0.0213	104.550
16	10	100	16,10,100	5744.344	4109.157	0.0210	112.981
32	10	100	32,10,100	5753.072	4090.430	0.0211	123.130
1	50	100	1,50,100	5980.236	4260.675	0.0228	97.283
4	50	100	4,50,100	6398.844	4377.032	0.0218	101.800
8	50	100	8,50,100	7001.159	4859.779	0.0243	104.162
16	50	100	16,50,100	6542.472	4737.485	0.0248	106.592
32	50	100	32,50,100	6057.119	4378.193	0.0236	119.411

ANEXO D

TABULACIÓN DE VALORES DE SIMULACIÓN PARA GRU

Tabla D.1. First group - GRU results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	6569.771	4855.293	0.02812	4.743
4	1	1	4,1,1	6615.609	4879.668	0.02839	4.755
8	1	1	8,1,1	6468.121	4774.461	0.02753	4.770
16	1	1	16,1,1	6381.878	4732.238	0.02775	4.928
32	1	1	32,1,1	6524.250	4895.939	0.03058	5.021
1	10	1	1,10,1	5969.190	4369.032	0.02413	4.730
4	10	1	4,10,1	5918.853	4303.231	0.02300	4.767
8	10	1	8,10,1	5941.580	4343.006	0.02351	4.819
16	10	1	16,10,1	5948.027	4334.812	0.02359	4.864
32	10	1	32,10,1	6052.370	4442.710	0.02455	4.915
1	50	1	1,50,1	5726.762	4159.749	0.02233	4.669
4	50	1	4,50,1	5835.018	4252.360	0.02256	4.714
8	50	1	8,50,1	5741.617	4171.770	0.02190	4.778
16	50	1	16,50,1	5900.426	4308.588	0.02321	4.876
32	50	1	32,50,1	6040.121	4468.919	0.02497	4.910
1	1	10	1,1,10	6007.400	4405.882	0.02433	7.174
4	1	10	4,1,10	5890.667	4264.852	0.02244	7.276
8	1	10	8,1,10	5836.200	4239.890	0.02254	7.353
16	1	10	16,1,10	5922.002	4306.030	0.02335	7.547
32	1	10	32,1,10	6086.061	4460.794	0.02489	7.746
1	10	10	1,10,10	5616.065	3983.146	0.02040	7.273
4	10	10	4,10,10	5715.431	4077.131	0.02059	7.351
8	10	10	8,10,10	5629.239	4018.160	0.02031	7.465
16	10	10	16,10,10	5760.788	4118.081	0.02136	7.624
32	10	10	32,10,10	6016.407	4400.741	0.02433	7.847
1	50	10	1,50,10	5693.598	4066.125	0.02132	7.272
4	50	10	4,50,10	6002.166	4280.342	0.02339	7.374
8	50	10	8,50,10	5831.738	4168.453	0.02109	7.474
16	50	10	16,50,10	5827.276	4187.464	0.02195	7.623
32	50	10	32,50,10	6052.259	4443.811	0.02485	7.930
1	1	100	1,1,100	6009.450	4449.406	0.02482	32.672
4	1	100	4,1,100	5612.749	3953.799	0.01970	34.093
8	1	100	8,1,100	5634.153	3970.874	0.01970	34.641
16	1	100	16,1,100	5637.084	3988.363	0.01994	35.598
32	1	100	32,1,100	5929.482	4224.268	0.02204	36.835
1	10	100	1,10,100	5761.358	4019.238	0.02043	34.079
4	10	100	4,10,100	5761.818	4085.790	0.02178	34.734
8	10	100	8,10,100	5923.270	4203.806	0.02125	35.865
16	10	100	16,10,100	5965.551	4189.438	0.02143	37.764
32	10	100	32,10,100	5896.241	4186.021	0.02129	37.481
1	50	100	1,50,100	6080.313	4421.152	0.02436	32.873
4	50	100	4,50,100	6326.006	4519.644	0.02345	34.110
8	50	100	8,50,100	6839.716	4897.951	0.02472	34.035
16	50	100	16,50,100	7199.972	5273.600	0.02659	35.464
32	50	100	32,50,100	6320.111	4615.995	0.02449	39.277

Tabla D.2. Second group - GRU results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	6360.410	4684.523	0.02635	5.091
4	1	1	4,1,1	6303.501	4608.186	0.02561	4.996
8	1	1	8,1,1	6189.367	4558.193	0.02556	4.901
16	1	1	16,1,1	6332.833	4689.390	0.02723	5.051
32	1	1	32,1,1	6304.758	4721.360	0.02835	5.490
1	10	1	1,10,1	5867.339	4267.535	0.02309	5.115
4	10	1	4,10,1	5852.714	4244.746	0.02237	4.862
8	10	1	8,10,1	5765.053	4184.822	0.02206	4.909
16	10	1	16,10,1	5908.953	4306.048	0.02328	4.946
32	10	1	32,10,1	6049.816	4467.086	0.02490	5.202
1	50	1	1,50,1	5866.334	4271.599	0.02304	4.915
4	50	1	4,50,1	5838.039	4267.848	0.02256	4.966
8	50	1	8,50,1	5765.330	4204.988	0.02215	5.232
16	50	1	16,50,1	5935.866	4368.570	0.02359	5.269
32	50	1	32,50,1	5999.229	4426.770	0.02420	5.329
1	1	10	1,1,10	6148.208	4535.183	0.02581	9.784
4	1	10	4,1,10	5866.668	4236.843	0.02217	10.357
8	1	10	8,1,10	5789.346	4193.518	0.02211	10.663
16	1	10	16,1,10	5858.335	4248.431	0.02282	11.038
32	1	10	32,1,10	6049.794	4432.880	0.02453	11.551
1	10	10	1,10,10	5818.687	4195.365	0.02242	10.326
4	10	10	4,10,10	5906.268	4146.437	0.02116	10.589
8	10	10	8,10,10	5640.153	4025.283	0.02046	10.571
16	10	10	16,10,10	5813.864	4186.414	0.02194	10.856
32	10	10	32,10,10	6184.696	4690.328	0.02857	11.720
1	50	10	1,50,10	5737.576	4175.129	0.02260	10.102
4	50	10	4,50,10	5929.785	4086.182	0.02103	10.253
8	50	10	8,50,10	5780.716	4124.516	0.02157	10.369
16	50	10	16,50,10	5904.450	4336.681	0.02326	10.687
32	50	10	32,50,10	6108.307	4608.849	0.02714	11.321
1	1	100	1,1,100	6073.434	4553.358	0.02636	59.597
4	1	100	4,1,100	5663.787	4003.391	0.02007	60.554
8	1	100	8,1,100	5600.504	3972.234	0.01962	66.179
16	1	100	16,1,100	5613.535	3977.132	0.01991	65.991
32	1	100	32,1,100	5932.439	4291.394	0.02325	71.698
1	10	100	1,10,100	6016.253	4346.663	0.02439	62.912
4	10	100	4,10,100	5987.468	4091.271	0.02039	64.926
8	10	100	8,10,100	6207.939	4278.759	0.02146	68.696
16	10	100	16,10,100	5860.762	4236.582	0.02151	72.864
32	10	100	32,10,100	5744.926	4090.100	0.02088	77.632
1	50	100	1,50,100	6210.329	4623.366	0.02722	61.279
4	50	100	4,50,100	6281.798	4326.437	0.02155	62.613
8	50	100	8,50,100	7114.088	5058.448	0.02531	65.951
16	50	100	16,50,100	6889.954	4996.702	0.02617	69.129
32	50	100	32,50,100	6352.796	4601.297	0.02446	73.971

Tabla D.3. Third group - GRU results

TimeStep (Lags)	Number Neurons	Epochs	Hyperparameters Setting	RMSE	MAE	MAPE	Lapsed Time [s]
1	1	1	1,1,1	6292.901	4625.978	0.02596	4.983
4	1	1	4,1,1	6361.407	4661.800	0.02604	4.962
8	1	1	8,1,1	6192.800	4557.082	0.02544	5.303
16	1	1	16,1,1	6202.509	4582.264	0.02627	5.346
32	1	1	32,1,1	6248.128	4663.599	0.02767	5.431
1	10	1	1,10,1	5900.884	4303.794	0.02343	5.142
4	10	1	4,10,1	5858.279	4256.796	0.02255	5.199
8	10	1	8,10,1	5753.003	4174.992	0.02198	5.316
16	10	1	16,10,1	5917.516	4322.414	0.02340	5.392
32	10	1	32,10,1	6120.331	4596.175	0.02654	5.605
1	50	1	1,50,1	6097.480	4467.955	0.02468	5.174
4	50	1	4,50,1	5885.619	4330.158	0.02334	5.252
8	50	1	8,50,1	5803.171	4244.146	0.02257	5.319
16	50	1	16,50,1	6044.986	4480.390	0.02471	5.238
32	50	1	32,50,1	6060.390	4545.008	0.02564	5.213
1	1	10	1,1,10	6164.286	4548.490	0.02592	11.887
4	1	10	4,1,10	5799.550	4158.502	0.02126	12.151
8	1	10	8,1,10	5692.001	4088.355	0.02100	12.682
16	1	10	16,1,10	5786.517	4170.085	0.02203	13.399
32	1	10	32,1,10	6030.681	4415.796	0.02435	14.205
1	10	10	1,10,10	5764.643	4136.643	0.02185	12.266
4	10	10	4,10,10	5843.914	4066.168	0.02090	12.443
8	10	10	8,10,10	5680.398	4008.220	0.02015	12.828
16	10	10	16,10,10	5789.711	4151.167	0.02144	13.526
32	10	10	32,10,10	6040.930	4522.583	0.02660	14.543
1	50	10	1,50,10	5739.662	4190.454	0.02291	12.495
4	50	10	4,50,10	5897.331	4074.475	0.02089	12.642
8	50	10	8,50,10	5789.226	4104.813	0.02134	12.979
16	50	10	16,50,10	5862.431	4296.989	0.02294	13.639
32	50	10	32,50,10	5891.884	4379.923	0.02481	14.654
1	1	100	1,1,100	6045.292	4510.236	0.02595	85.225
4	1	100	4,1,100	5648.052	3989.837	0.01997	88.006
8	1	100	8,1,100	5600.595	3971.039	0.01962	89.822
16	1	100	16,1,100	5616.484	3980.177	0.01994	95.403
32	1	100	32,1,100	5839.373	4195.324	0.02210	105.138
1	10	100	1,10,100	5927.587	4297.485	0.02408	94.793
4	10	100	4,10,100	6011.212	4098.466	0.02028	93.348
8	10	100	8,10,100	6294.265	4345.685	0.02167	99.651
16	10	100	16,10,100	5804.764	4182.763	0.02142	98.484
32	10	100	32,10,100	5747.472	4098.183	0.02100	108.213
1	50	100	1,50,100	6312.766	4671.989	0.02839	86.705
4	50	100	4,50,100	6268.669	4301.007	0.02143	90.225
8	50	100	8,50,100	7011.844	5015.805	0.02519	92.308
16	50	100	16,50,100	6797.988	4904.121	0.02541	98.080
32	50	100	32,50,100	6364.698	4669.818	0.02650	108.948

ANEXO E

TABULACIÓN DE VALORES DE SIMULACIÓN PARA OS-ELM

Tabla E.1. First group - OS-ELM results

Number Neurons	Forgetting Factor	RMSE	MAE	MAPE	Lapsed Time [s]
10	0.9	9518.178	6344.956	0.03877	0.142
10	0.95	10447.018	7477.483	0.04325	0.067
10	0.99	20116.983	16061.938	0.08413	0.069
10	0.995	22149.009	17693.711	0.09213	0.069
10	1	24030.764	19198.389	0.09998	0.070
110	0.9	24773.798	15367.068	0.06576	0.113
110	0.95	11183.264	7400.749	0.03484	0.100
110	0.99	3815.085	2782.685	0.01810	0.099
110	0.995	4038.103	2938.963	0.01918	0.089
110	1	4694.241	3364.208	0.02149	0.093
210	0.9	13126.480	7256.976	0.03271	0.146
210	0.95	8287.700	5181.770	0.02434	0.156
210	0.99	3255.397	2380.911	0.01279	0.138
210	0.995	3027.727	2225.221	0.01215	0.143
210	1	3275.512	2326.732	0.01256	0.140
310	0.9	6814.641	4687.769	0.02342	0.243
310	0.95	5829.936	4149.298	0.02077	0.249
310	0.99	3220.672	2412.050	0.01318	0.250
310	0.995	2946.759	2210.344	0.01231	0.264
310	1	3071.738	2242.558	0.01230	0.322
410	0.9	4563.982	3457.625	0.01882	1.070
410	0.95	4336.068	3273.084	0.01761	1.128
410	0.99	2986.602	2245.612	0.01250	1.071
410	0.995	2782.776	2093.925	0.01168	1.147
410	1	2886.329	2122.076	0.01154	1.111
510	0.9	4565.021	3447.846	0.01880	1.816
510	0.95	4366.424	3308.021	0.01766	1.668
510	0.99	3065.389	2336.221	0.01301	1.659
510	0.995	2825.950	2146.006	0.01211	1.734
510	1	2856.735	2138.646	0.01190	1.714
610	0.9	4193.854	3156.977	0.01741	2.338
610	0.95	4095.412	3062.757	0.01663	2.370
610	0.99	3149.027	2367.706	0.01324	2.388
610	0.995	2894.098	2151.517	0.01229	2.490
610	1	2876.412	2090.328	0.01195	2.486
710	0.9	3826.541	2909.227	0.01638	3.358
710	0.95	3738.864	2818.195	0.01564	3.259
710	0.99	2997.537	2283.044	0.01285	3.342
710	0.995	2775.530	2119.636	0.01208	3.366
710	1	2751.764	2045.987	0.01165	3.279
810	0.9	3715.441	2937.893	0.01663	4.433
810	0.95	3579.276	2818.415	0.01579	4.392
810	0.99	2957.600	2318.680	0.01306	4.526
810	0.995	2741.812	2143.297	0.01224	4.370
810	1	2683.983	2036.018	0.01169	4.374
910	0.9	3592.576	2844.872	0.01615	5.813
910	0.95	3489.045	2761.651	0.01547	5.696
910	0.99	2946.158	2308.912	0.01294	5.703
910	0.995	2740.974	2136.257	0.01213	5.759
910	1	2661.935	2028.125	0.01155	5.700

Tabla E.2. First group - OS-ELM results

Number Neurons	Forgetting Factor	RMSE	MAE	MAPE	Lapsed Time [s]
1010	0.9	3538.666	2778.493	0.01600	7.140
1010	0.95	3440.850	2701.221	0.01532	7.370
1010	0.99	2957.347	2315.113	0.01321	7.166
1010	0.995	2774.945	2161.927	0.01249	7.232
1010	1	2717.057	2082.360	0.01206	7.208
1110	0.9	3643.603	2829.937	0.01646	8.928
1110	0.95	3545.353	2742.221	0.01571	8.878
1110	0.99	3074.664	2355.257	0.01348	8.986
1110	0.995	2874.876	2201.640	0.01274	8.813
1110	1	2763.342	2088.075	0.01214	8.982
1210	0.9	3389.862	2663.211	0.01559	11.262
1210	0.95	3306.455	2585.616	0.01495	11.118
1210	0.99	2925.682	2265.692	0.01311	11.278
1210	0.995	2752.683	2145.790	0.01255	11.226
1210	1	2646.672	2049.718	0.01207	11.219
1310	0.9	3286.706	2561.644	0.01536	13.217
1310	0.95	3216.566	2507.593	0.01482	13.280
1310	0.99	2865.721	2247.599	0.01325	13.339
1310	0.995	2723.324	2143.885	0.01273	13.245
1310	1	2643.741	2048.163	0.01221	13.377
1410	0.9	3393.612	2625.812	0.01562	15.697
1410	0.95	3305.670	2558.466	0.01502	15.854
1410	0.99	2981.871	2309.858	0.01345	15.768
1410	0.995	2823.314	2194.610	0.01290	15.722
1410	1	2693.521	2090.383	0.01238	15.601
1510	0.9	3370.988	2645.770	0.01563	18.081
1510	0.95	3313.622	2580.004	0.01505	18.189
1510	0.99	3001.985	2336.505	0.01354	18.001
1510	0.995	2845.344	2229.373	0.01301	17.981
1510	1	2709.221	2129.918	0.01250	18.109
1610	0.9	3215.939	2534.384	0.01496	21.017
1610	0.95	3147.107	2474.343	0.01444	20.991
1610	0.99	2878.874	2251.807	0.01303	21.004
1610	0.995	2742.549	2146.220	0.01252	21.051
1610	1	2623.975	2047.366	0.01203	21.196
1710	0.9	3267.404	2573.922	0.01527	24.072
1710	0.95	3207.005	2528.353	0.01483	24.121
1710	0.99	2960.628	2329.307	0.01355	23.885
1710	0.995	2835.284	2228.045	0.01306	24.053
1710	1	2724.294	2134.799	0.01258	23.980
1810	0.9	3132.301	2504.491	0.01491	28.149
1810	0.95	3051.633	2435.082	0.01439	28.236
1810	0.99	2849.639	2268.773	0.01325	27.939
1810	0.995	2748.642	2181.278	0.01280	27.946
1810	1	2659.198	2093.724	0.01236	28.066
1910	0.9	3151.609	2489.225	0.01488	31.911
1910	0.95	3081.088	2428.907	0.01443	31.976
1910	0.99	2888.963	2269.098	0.01329	31.878
1910	0.995	2793.084	2189.906	0.01287	31.734
1910	1	2694.890	2107.548	0.01244	31.804

Tabla E.3. Second group - OS-ELM results

Number Neurons	Forgetting Factor	RMSE	MAE	MAPE	Lapsed Time [s]
10	0.9	20117.256	9109.940	0.05642	0.091
10	0.95	13058.925	9341.422	0.06228	0.069
10	0.99	29173.025	21481.911	0.11620	0.067
10	0.995	34078.132	24857.005	0.13027	0.071
10	1	38499.311	27737.583	0.14283	0.070
110	0.9	37684.421	21316.569	0.08850	0.092
110	0.95	11077.969	7290.247	0.03247	0.091
110	0.99	3646.505	2719.539	0.01546	0.091
110	0.995	3802.878	2805.557	0.01617	0.094
110	1	4482.937	3248.657	0.01838	0.092
210	0.9	13475.129	7508.044	0.03411	0.158
210	0.95	8670.672	5624.273	0.02641	0.143
210	0.99	3314.602	2380.069	0.01342	0.153
210	0.995	3109.600	2274.129	0.01308	0.142
210	1	3506.822	2529.973	0.01414	0.204
310	0.9	5715.490	3728.667	0.01968	0.253
310	0.95	5148.772	3512.700	0.01828	0.252
310	0.99	3004.841	2265.511	0.01289	0.277
310	0.995	2796.760	2171.049	0.01245	0.266
310	1	3011.770	2309.079	0.01291	0.268
410	0.9	4425.352	3213.695	0.01778	1.058
410	0.95	4221.912	3037.816	0.01642	1.182
410	0.99	3108.086	2285.874	0.01273	1.094
410	0.995	2849.514	2140.723	0.01211	1.142
410	1	2858.322	2176.607	0.01217	1.142
510	0.9	3982.600	2939.566	0.01685	1.674
510	0.95	3887.684	2842.497	0.01593	1.779
510	0.99	3157.159	2347.349	0.01332	1.737
510	0.995	2918.107	2230.114	0.01274	1.912
510	1	2885.479	2196.272	0.01250	1.734
610	0.9	3730.220	2859.060	0.01641	2.374
610	0.95	3635.407	2754.626	0.01557	2.422
610	0.99	3044.922	2291.230	0.01327	2.429
610	0.995	2834.409	2177.701	0.01274	2.396
610	1	2804.704	2170.089	0.01262	2.405
710	0.9	3524.341	2735.240	0.01600	3.435
710	0.95	3450.040	2652.238	0.01525	3.268
710	0.99	3021.454	2298.983	0.01331	3.329
710	0.995	2820.213	2150.124	0.01269	3.292
710	1	2747.953	2139.596	0.01259	3.298
810	0.9	3512.667	2735.985	0.01595	4.537
810	0.95	3441.602	2666.664	0.01535	4.375
810	0.99	3048.632	2340.783	0.01360	4.392
810	0.995	2853.813	2226.214	0.01307	4.498
810	1	2750.845	2186.928	0.01281	4.385
910	0.9	3369.028	2645.914	0.01584	5.618
910	0.95	3304.476	2582.256	0.01519	5.789
910	0.99	2986.013	2326.393	0.01371	5.690
910	0.995	2838.166	2224.149	0.01323	5.714
910	1	2771.505	2196.174	0.01302	5.751

Tabla E.4. Second group - OS-ELM results

Number Neurons	Forgetting Factor	RMSE	MAE	MAPE	Lapsed Time [s]
1010	0.9	3213.902	2485.934	0.01495	7.202
1010	0.95	3174.153	2446.635	0.01450	7.212
1010	0.99	2903.160	2247.321	0.01325	7.179
1010	0.995	2764.915	2159.697	0.01283	7.146
1010	1	2687.220	2134.519	0.01263	7.264
1110	0.9	3157.495	2486.606	0.01490	8.782
1110	0.95	3096.048	2441.669	0.01449	8.930
1110	0.99	2881.358	2285.601	0.01356	8.974
1110	0.995	2757.975	2195.704	0.01316	8.833
1110	1	2681.799	2166.867	0.01297	9.012
1210	0.9	3122.518	2468.755	0.01496	11.159
1210	0.95	3074.744	2431.464	0.01458	11.596
1210	0.99	2882.478	2281.962	0.01361	11.918
1210	0.995	2780.049	2202.703	0.01323	11.259
1210	1	2710.093	2177.818	0.01305	11.162
1310	0.9	3073.274	2413.782	0.01451	13.208
1310	0.95	3031.795	2382.368	0.01421	13.311
1310	0.99	2855.443	2241.310	0.01332	13.360
1310	0.995	2738.975	2148.520	0.01288	13.218
1310	1	2633.021	2080.787	0.01253	13.162
1410	0.9	3053.697	2441.372	0.01451	15.754
1410	0.95	3038.414	2422.547	0.01429	15.708
1410	0.99	2885.336	2287.570	0.01349	15.680
1410	0.995	2788.005	2214.275	0.01314	15.779
1410	1	2702.085	2161.732	0.01285	15.712
1510	0.9	2995.507	2302.195	0.01391	18.154
1510	0.95	2963.620	2288.220	0.01372	18.381
1510	0.99	2840.262	2187.672	0.01307	18.233
1510	0.995	2755.514	2128.816	0.01279	18.114
1510	1	2669.063	2092.205	0.01259	18.342
1610	0.9	2950.376	2328.141	0.01378	21.120
1610	0.95	2910.679	2300.299	0.01357	21.381
1610	0.99	2784.678	2191.986	0.01293	21.150
1610	0.995	2711.350	2130.812	0.01264	21.145
1610	1	2646.293	2078.112	0.01237	21.320
1710	0.9	2983.426	2348.267	0.01413	24.047
1710	0.95	2941.477	2319.911	0.01391	24.210
1710	0.99	2829.665	2230.983	0.01332	24.126
1710	0.995	2761.738	2179.401	0.01306	24.145
1710	1	2695.948	2131.668	0.01280	24.014
1810	0.9	2860.484	2302.973	0.01391	28.242
1810	0.95	2838.065	2286.009	0.01373	28.284
1810	0.99	2732.408	2188.892	0.01312	28.185
1810	0.995	2672.533	2141.838	0.01289	28.268
1810	1	2613.818	2106.573	0.01268	28.265
1910	0.9	2846.211	2261.756	0.01367	32.041
1910	0.95	2831.315	2244.945	0.01349	31.965
1910	0.99	2738.269	2165.189	0.01295	31.942
1910	0.995	2685.148	2123.049	0.01274	31.883
1910	1	2636.693	2097.742	0.01258	31.957

Tabla E.5. Third group - OS-ELM results

Number Neurons	Forgetting Factor	RMSE	MAE	MAPE	Lapsed Time [s]
10	0.9	16339.629	9384.567	0.05783	0.092
10	0.95	14484.560	9956.305	0.06466	0.069
10	0.99	21753.823	16303.731	0.10193	0.070
10	0.995	24178.829	17857.327	0.11171	0.070
10	1	26369.795	19430.043	0.12224	0.072
110	0.9	47798.870	24185.199	0.10071	0.097
110	0.95	14008.613	8674.388	0.03883	0.093
110	0.99	4100.081	3062.899	0.01784	0.092
110	0.995	4307.946	3241.140	0.01904	0.098
110	1	5035.674	3741.690	0.02159	0.095
210	0.9	19954.726	8494.369	0.03693	0.174
210	0.95	9738.465	5566.033	0.02570	0.179
210	0.99	3572.369	2707.083	0.01480	0.155
210	0.995	3474.253	2632.742	0.01466	0.158
210	1	3835.781	2807.140	0.01559	0.167
310	0.9	6352.361	4294.393	0.02174	0.275
310	0.95	5633.419	3859.218	0.01961	0.265
310	0.99	3533.834	2589.914	0.01426	0.274
310	0.995	3260.704	2416.496	0.01362	0.257
310	1	3386.356	2472.499	0.01387	0.296
410	0.9	4587.048	3431.223	0.01882	1.140
410	0.95	4384.103	3276.136	0.01778	1.113
410	0.99	3278.790	2553.202	0.01451	1.127
410	0.995	3068.922	2361.303	0.01374	1.133
410	1	3156.913	2377.180	0.01375	1.142
510	0.9	4058.494	3150.569	0.01766	1.861
510	0.95	3931.416	3046.943	0.01691	1.760
510	0.99	3159.923	2502.659	0.01423	1.712
510	0.995	2953.041	2368.254	0.01362	1.732
510	1	2944.116	2333.071	0.01340	1.752
610	0.9	3596.982	2863.029	0.01619	2.428
610	0.95	3521.108	2790.580	0.01566	2.479
610	0.99	3065.820	2446.975	0.01389	2.437
610	0.995	2836.113	2268.224	0.01313	2.564
610	1	2762.903	2201.870	0.01277	2.407
710	0.9	3368.045	2651.728	0.01519	3.389
710	0.95	3316.643	2591.473	0.01472	3.330
710	0.99	2972.525	2340.077	0.01335	3.360
710	0.995	2789.908	2220.665	0.01283	3.497
710	1	2712.284	2139.846	0.01244	3.321
810	0.9	3520.609	2731.606	0.01536	4.518
810	0.95	3475.003	2686.539	0.01500	4.489
810	0.99	3122.810	2409.146	0.01363	4.583
810	0.995	2929.472	2276.847	0.01307	4.543
810	1	2799.360	2183.526	0.01266	4.472
910	0.9	3339.669	2640.449	0.01516	5.911
910	0.95	3288.807	2580.457	0.01468	5.711
910	0.99	3009.332	2338.746	0.01339	5.784
910	0.995	2839.323	2219.157	0.01286	5.874
910	1	2702.602	2113.756	0.01239	5.814

Tabla E.6. Third group - OS-ELM results

Number Neurons	Forgetting Factor	RMSE	MAE	MAPE	Lapsed Time [s]
1010	0.9	3268.363	2638.530	0.01525	7.373
1010	0.95	3227.611	2599.060	0.01490	7.316
1010	0.99	2977.599	2392.777	0.01377	7.220
1010	0.995	2842.762	2292.936	0.01335	7.394
1010	1	2728.279	2197.906	0.01294	7.277
1110	0.9	3257.415	2566.175	0.01492	9.051
1110	0.95	3210.294	2521.505	0.01457	8.916
1110	0.99	2991.257	2332.108	0.01348	9.066
1110	0.995	2869.447	2238.199	0.01306	9.128
1110	1	2752.108	2155.685	0.01267	8.964
1210	0.9	3225.348	2492.428	0.01468	11.363
1210	0.95	3187.115	2465.834	0.01441	11.335
1210	0.99	3006.546	2325.932	0.01355	11.209
1210	0.995	2890.572	2242.303	0.01317	11.415
1210	1	2772.724	2164.397	0.01280	11.330
1310	0.9	3049.582	2401.745	0.01417	13.357
1310	0.95	3037.566	2392.662	0.01397	13.429
1310	0.99	2901.856	2270.661	0.01321	13.435
1310	0.995	2809.879	2207.526	0.01289	13.317
1310	1	2713.225	2141.247	0.01256	13.391
1410	0.9	3145.673	2494.660	0.01466	15.894
1410	0.95	3100.956	2455.043	0.01435	15.885
1410	0.99	2959.222	2334.995	0.01361	15.909
1410	0.995	2862.634	2271.171	0.01331	15.813
1410	1	2747.048	2195.263	0.01294	15.818
1510	0.9	2914.779	2304.487	0.01387	18.320
1510	0.95	2906.483	2293.301	0.01370	18.292
1510	0.99	2800.541	2199.187	0.01307	18.349
1510	0.995	2734.120	2151.809	0.01283	18.341
1510	1	2665.251	2106.224	0.01260	18.417
1610	0.9	3020.125	2430.698	0.01427	21.368
1610	0.95	2990.673	2390.516	0.01399	21.300
1610	0.99	2885.666	2293.988	0.01343	22.073
1610	0.995	2811.450	2237.254	0.01317	21.442
1610	1	2722.825	2174.215	0.01288	21.452
1710	0.9	2942.098	2343.208	0.01376	26.714
1710	0.95	2915.188	2307.354	0.01352	24.412
1710	0.99	2816.109	2215.167	0.01299	24.507
1710	0.995	2755.990	2170.564	0.01277	24.513
1710	1	2692.903	2118.451	0.01252	24.462
1810	0.9	2904.413	2297.851	0.01358	28.525
1810	0.95	2849.189	2241.468	0.01330	28.491
1810	0.99	2765.894	2173.994	0.01287	28.413
1810	0.995	2712.407	2143.923	0.01272	28.587
1810	1	2649.944	2103.028	0.01250	28.647
1910	0.9	2886.515	2269.565	0.01352	32.845
1910	0.95	2850.386	2240.994	0.01332	32.266
1910	0.99	2769.063	2168.933	0.01285	32.178
1910	0.995	2718.117	2133.532	0.01267	32.079
1910	1	2658.902	2091.701	0.01245	32.099