



UNIVERSIDAD NACIONAL DE COLOMBIA

Aprendizaje profundo para la predicción de temperatura en las paredes refractarias de un horno de arco eléctrico

Diego Fernando Godoy Rojas

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia
2022

Aprendizaje profundo para la predicción de temperatura en las paredes refractarias de un horno de arco eléctrico

Diego Fernando Godoy Rojas

Trabajo de grado presentado como requisito parcial para optar al título de:
Magíster en Ingeniería - Automatización Industrial

Director: Ph.D Diego Alexander Tibaduiza Burgos
Co-Director: Ph.D Jersson Xavier Leon Medina

Línea de Investigación:
Automatización de Procesos y Máquinas

Grupo de Investigación:
Grupo de investigación en electrónica de alta frecuencia y telecomunicaciones - CMUN

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia

2022

Vive como si fueras a morir mañana y aprende
como si el mundo fuera a durar para siempre.

Mahatma Gandhi

Agradecimientos

En primer lugar quiero agradecer a mi director Diego Alexander Tibaduiza, profesor y mentor durante mi pregrado y maestría, por su acompañamiento, consejos y ayuda para poder llevar a cabo el desarrollo del presente proyecto. Agradezco a mi co-director Jersson Leon Medina, quien fue mi compañero de trabajo y guía a lo largo del proyecto y un pilar fundamental para la redacción del artículo publicado.

A mis padres Luis Godoy y Leydi Rojas por sus enseñanzas desde pequeño, con su guianza y consejos he logrado llegar a donde estoy hoy. A mis hermanos Daniela y David que han sido un apoyo de alegría y un aliento en los momentos mas difíciles. A ti Alejandra por todo tu amor y motivación, por tu forma de ver la vida y tu honestidad, que me motivan a siempre ser una mejor versión de mi.

Agradezco a la Universidad Nacional de Colombia (UNAL) por abrirme sus puertas hace 8 años y brindarme el honor de ser estudiante y profesor, en sus espacios he podido crecer como persona, amigo, hijo, hermano, académico, pero sobretodo como un ser integro. Recalco el gran apoyo que signifco para mi estancia en la maestría el programa “Beca - Asistente docente” en la sede UNAL-Bogotá.

Agradezco a todos los miembros del equipo de trabajo del proyecto en conjunto entre la Universidad Nacional y South 32 Cerromatoso S.A, por darme su apoyo, ideas y otros desarrollos que me brindaron una guía para la finalización del proyecto. En especial a South 32 Cerromatoso S.A por proveer el conjunto de datos con el cual se llevan a cabo las pruebas de funcionamiento del modelo construido.

Al Profesor John Alexander Cortés del Departamento de Ingeniería Eléctrica y Electrónica, por su calidad como docente y como persona, el cual me motivo a indagar en el ámbito de transmitir el conocimiento como profesor.

Resumen

Aprendizaje profundo para la predicción de temperatura en las paredes refractarias de un horno de arco eléctrico

En el presente documento se detalla el flujo de trabajo llevado a cabo para el desarrollo de modelos de aprendizaje profundo para la estimación de temperatura de pared media en dos hornos de arco eléctrico pertenecientes a la empresa Cerro Matoso S.A. El documento inicia con una introducción al contexto bajo el cual se desarrollo el trabajo final de maestría, dando paso a la descripción teórica de todos los aspectos relevantes y generalidades sobre el funcionamiento de la planta, las series de tiempo y el aprendizaje profundo requeridas durante el desarrollo del proyecto. El flujo de trabajo se divide en una metodología de 3 pasos empezando por el estudio y preparación del conjunto de datos brindado por CMSA, seguido por el desarrollo, entrenamiento y selección de diversos modelos de aprendizaje profundo usados en predicciones con datos de un conjunto de prueba obteniendo errores RMSE entre 1-2 °C y finalizando con una etapa de validación que estudia el desempeño de los diversos modelos obtenidos frente a diversas variaciones en las condiciones de los parámetros de entrenamiento.

Palabras clave: Aprendizaje profundo, GRU, Inteligencia artificial, LSTM, Mecanismos de atención, Redes Neuronales, Salud estructural, Series de tiempo. .

Abstract

Deep learning for temperature prediction in the refractory walls of an electric arc furnace

This document details the workflow followed for the development of deep learning models for the estimation of mean wall temperature in two electric arc furnaces belonging to the company Cerro Matoso S.A. The document begins by establishing the development context of the final master's degree project. Afterwards, the theoretical description of all the relevant aspects and generalities about the operation of the plant, the time series and the deep learning required during the development of the project is given. The workflow is divided into a 3-step methodology starting with the study and preparation of the data set provided by CMSA, followed by the development, training and selection of various deep learning models used in predictions with data from a test set. obtaining RMSE errors between 1-2 °C and ending with a validation stage that studies the performance of the various models obtained against various variations in the conditions of the training parameters.

Keywords: Attention Mechanisms, Deep Learning, GRU, LSTM, Neural Networks, Time Series forecasting.

Contenido

Agradecimientos	VII
Resumen	IX
Lista de figuras	XI
Lista de tablas	XIV
Lista de abreviaturas	XVII
1. Introducción	1
1.1. Introducción	1
1.2. Antecedentes y justificación	2
1.3. Objetivos	5
1.3.1. Objetivo general	5
1.3.2. Objetivos específicos	5
1.4. Resultados generales	6
2. Marco teórico	7
2.1. Horno de arco eléctrico	7
2.1.1. Termocuplas	9
2.1.2. Red de termocuplas hornos CMSA.	9
2.2. Series de tiempo	10
2.2.1. Componentes de las series de tiempo	10
2.3. Aprendizaje profundo	12
2.3.1. Redes neuronales	12
2.3.2. Arquitectura	12
2.3.3. Costo y funciones de costo	16
2.3.4. Propagación hacia delante, descenso por gradiente y retro propagación	18
2.3.5. Hiperparámetros en redes neuronales	20
2.4. Redes neuronales recurrentes - RNN	24
2.4.1. Long Short Term Memory - LSTM	25
2.4.2. Gated recurrent units - GRU	27
2.4.3. Arquitectura Encoder-Decoder	29
2.4.4. Mecanismos de atención	30

3. Metodología	32
4. Preparación del conjunto de datos	34
4.1. Análisis exploratorio y limpieza de datos	34
4.1.1. Conjunto de datos horno línea 1	36
4.1.2. Conjunto de datos horno línea 2	37
4.2. Variables de entrada y salida	38
4.2.1. Horno línea 1	39
4.2.2. Horno línea 2	39
4.3. Tiempo de predicción	40
4.4. Conjunto de entrenamiento y prueba	41
4.4.1. Horno línea 1	41
4.4.2. Horno línea 2	42
4.5. Escalamiento de datos	42
4.6. Generación de lotes	43
5. Desarrollo, entrenamiento y comparación de arquitecturas RNN	44
5.1. Función de costo y tiempo de calentamiento	44
5.2. Arquitecturas RNN, entrenamiento, hiperparámetros y ajuste de hiper- parámetros	46
5.2.1. Modelos Horno línea 1	47
5.2.2. Modelos horno línea 2	60
6. Validación de arquitecturas RNN	73
6.1. Tiempo de predicción	74
6.2. Cantidad de variables en la predicción	77
6.3. Validación cruzada	78
6.3.1. Ventana móvil de tiempo	78
6.3.2. Ventana móvil de tiempo - Conjunto de prueba estático	80
6.4. Distribución promedio de la función de costo	81
7. Conclusiones	83
Bibliografía	85
A. Anexo: Publicación revista indexada	91
B. Anexo: Acuerdo de trabajo conjunto South 32 Cerromatoso S.A	112

Lista de Figuras

2-1. Planta CMSA [21].	7
2-2. Ubicación de partes y componentes básicos en un horno de arco eléctrico [21].	8
2-3. Distribución de paneles por zonas en los hornos línea 1 y línea 2 de CMSA. .	10
2-4. Descomposición de los componentes de una serie de tiempo.	11
2-5. Arquitectura típica de un red neuronal.	13
2-6. Entradas, pesos y nodos de capas ocultas en una red neuronal.	14
2-7. Funciones de activación Sigmoide, Tanh y ReLU.	16
2-8. Propagación hacia adelante en las capas de una red neuronal.	19
2-9. Clasificación de hiperparámetros más comunes	21
2-10. Comportamiento del descenso por gradiente en la función de costo L a partir de la tasa de aprendizaje γ	23
2-11. Relación entre el número de épocas de entrenamiento y las funciones de pérdida de los conjunto de entrenamiento y validación.	23
2-12.(a) Estructura resumida de un RNN (b) Estructura detallada de un RNN. .	24
2-13. Arquitectura interna red LSTM.	26
2-14. Arquitectura interna red GRU.	27
3-1. Etapas generales de desarrollo y construcción de modelos de aprendizaje profundo para la predicción de temperatura en un horno de arco eléctrico. .	32
4-1. Etapas específicas para la preparación del conjunto de datos que alimentara los modelos de aprendizaje profundo.	34
4-2. Flujo de trabajo para la limpieza de datos en los hornos línea 1 y línea 2. . .	35
4-3. Paneles seleccionados en el horno para la predicción de temperatura en sus termocuplas.	38
4-4. Escalamiento aplicado a los datos de temperatura en la termocupla 16. . . .	42
5-1. Etapas específicas de desarrollo y entrenamiento de modelos de aprendizaje profundo para predicción de series de tiempo.	44
5-2. Tiempo de calentamiento RMSE.	46
5-3. Pérdida de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.	49
5-4. RMSE de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.	50

5-5. Pérdida de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.	53
5-6. RMSE de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.	54
5-7. Pérdida del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.	56
5-8. RMSE del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.	56
5-9. Pérdida del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.	58
5-10. RMSE del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.	59
5-11. Pérdida de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.	62
5-12. RMSE de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.	63
5-13. Pérdida de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.	66
5-14. RMSE de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.	67
5-15. Pérdida del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.	69
5-16. RMSE del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.	69
5-17. Pérdida del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.	71
5-18. RMSE del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.	72
6-1. Etapas específicas de validación de modelos desarrollados para la predicción de temperatura en los hornos línea 1 y línea 2.	73
6-2. Influencia del tiempo de predicción en el RMSE para el conjunto de prueba en modelos de predicción del horno línea 1.	75
6-3. Influencia del tiempo de predicción en el RMSE para el conjunto de prueba en modelos de predicción del horno línea 2.	75
6-4. Comportamiento de las predicciones para el conjunto de prueba en modelos de predicción GRU y GRU con atención del horno línea 1 a partir del tiempo de predicción.	76
6-5. Patrón de selección de paneles para el aumento de termocuplas a predecir.	77

6-6. Selección de datos para validación cruzada con desplazamiento en conjuntos de entrenamiento y prueba.	79
6-7. Selección de datos para validación cruzada con desplazamiento del conjunto de entrenamiento manteniendo el conjunto de prueba estático.	80
6-8. Distribución RMSE en las predicciones para cada una de las 16 termocuplas con el modelo GRU en el horno línea 1.	82
6-9. Distribución RMSE en las predicciones para cada una de las 16 termocuplas con el modelo GRU con atención en el horno línea 2.	82

Lista de Tablas

2-1. Estándar ISA/ASTM para la clasificación de termocuplas [22].	9
4-1. Limpieza del conjunto de datos para el horno línea 1.	37
4-2. Limpieza del conjunto de datos para el horno línea 2.	37
4-3. Cantidad de variables seleccionadas por proceso para la entrada y salida del modelo de predicción de temperaturas en el horno línea 1.	39
4-4. Cantidad de variables seleccionadas por proceso para la entrada y salida del modelo de predicción de temperaturas en el horno línea 2.	40
4-5. Cantidad de datos por partición del conjunto de datos para prueba y entrenamiento en el horno línea 1.	41
4-6. Cantidad de datos por partición del conjunto de datos para prueba y entrenamiento en el horno línea 2.	42
5-1. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 32 celdas LSTM.	47
5-2. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM.	48
5-3. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 96 celdas LSTM.	49
5-4. Mejores resultados para cada tipo de arquitectura LSTM.	50
5-5. Arquitectura interna red neuronal LSTM seleccionada - Horno línea 1.	50
5-6. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 100 unidades GRU.	51
5-7. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 200 unidades GRU.	52
5-8. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU.	53
5-9. Mejores resultados para cada tipo de arquitectura GRU.	54
5-10. Arquitectura interna red neuronal GRU seleccionada - Horno línea 1.	54
5-11. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM con mecanismos de atención.	55
5-12. Arquitectura interna red neuronal LSTM con mecanismos de atención - Horno línea 1.	57

5-13. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU con mecanismos de atención.	58
5-14. Arquitectura interna red neuronal GRU con mecanismos de atención - Horno línea 1.	59
5-15. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 32 celdas LSTM.	60
5-16. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM.	61
5-17. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 96 celdas LSTM.	62
5-18. Mejores resultados para cada tipo de arquitectura LSTM.	63
5-19. Arquitectura interna red neuronal LSTM seleccionada - Horno línea 2.	63
5-20. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 100 unidades GRU.	64
5-21. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 200 unidades GRU.	65
5-22. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU.	66
5-23. Mejores resultados para cada tipo de arquitectura GRU.	67
5-24. Arquitectura interna red neuronal GRU seleccionada - Horno línea 2.	67
5-25. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM con mecanismos de atención.	68
5-26. Arquitectura interna red neuronal LSTM con mecanismos de atención - Horno línea 2.	70
5-27. Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU con mecanismos de atención.	71
5-28. Arquitectura interna red neuronal GRU con mecanismos de atención - Horno línea 2.	72
6-1. Comportamiento del RMSE para los modelos de predicción horno línea 1 y 2 ante la variación del tiempo de predicción.	74
6-2. Comportamiento del RMSE para los modelos de predicción horno línea 1 y 2 ante el aumento de termocuplas a predecir.	78
6-3. RMSE obtenido por iteración de ventana móvil para el conjunto de prueba.	80
6-4. RMSE obtenido por iteración de ventana móvil para el conjunto de prueba estático.	81

Lista de abreviaturas

Abreviatura	Término
HAE	Horno de arco eléctrico
CMSA	Cerromatoso S.A.
UNAL.	Universidad Nacional de Colombia
GRU.	Gate Recurrent Unit.
LSTM.	Long Short-Term Memory.
RNN.	Recurrent Neural Network.
ISA.	International Society of Automation.
RMSE.	Root Mean Squared Error.

1. Introducción

1.1. Introducción

El control y monitoreo de la salud estructural son dos procesos fundamentales en la operación de un sistema a nivel industrial. Estos surgen como una necesidad continua, pues el sistema siempre está sujeto a cambios extremos en las condiciones de operación, lo que puede resultar en posibles daños a nivel estructural. De esta manera, el desarrollo adecuado de estos dos procesos está asociado con el uso de datos confiables capturados por parte de la red de sensores usados en el sistema, que a su vez requiere el uso de técnicas avanzadas de procesamiento de señales, para de esta manera poder producir un sistema confiable y así evitar fallas en el proceso [1].

En la industria minera se utilizan diversas estructuras y sistemas de alta complejidad, entre ellas se encuentra el horno de arco eléctrico (HAE), el cual calienta los materiales mediante el proceso de fundición de arco cubierto, en donde su eficiencia depende del control y monitoreo de algunas variables como la potencia, la temperatura del horno, la alimentación entregada, la composición química del material a fundir, entre otras. Esto se da gracias a un conjunto de sensores ubicados en los diferentes sistemas que monitorean las variables mencionadas. Algunos HAE funcionan en el orden de Mega Volt-Amperes (MVA), lo que significa que cualquier mejora en la eficiencia representa un ahorro de energía para la industria [2]. En este campo tradicionalmente se han empleado técnicas analíticas para la predicción de variables debido a su baja carga computacional, sin embargo, estas técnicas presentan fallas cuando se incluyen muchas variables de entrada, que con el crecimiento de la industria cada vez tienden a aumentar más [3]. Otras técnicas empleadas corresponden a modelos autorregresivos, sin embargo, los procesos de estimación de parámetros generalmente no son adaptativos, y estos hornos al ser un sistema complejo y no lineal, requieren de un modelo que se adapte continuamente.

En la última década, algunas técnicas de aprendizaje automático, como las redes neuronales y la lógica difusa, se han utilizado para modelar y estimar algunas de estas variables. Estas técnicas presentan ciertas ventajas sobre los métodos tradicionales al tener un comportamiento adaptativo, poder trabajar con múltiples variables de entrada y salida, tener la capacidad de aprender de patrones ocultos, entre otras ventajas [4][5]. Bajo este contexto se desarrolló el trabajo final de maestría, el cual hizo parte del proyecto “Incremento de la capacidad operativa y eficiencia de producción de Cerro Matoso S.A” surgido a partir del

convenio entre la Universidad Nacional De Colombia (UNAL) y la Empresa Cerro Matoso SA (CMSA), en el marco de proyectos del Ministerio de Ciencia, Tecnología e Innovación (Minciencias). Este proyecto tiene completadas cuatro fases de desarrollo e implementación.

CMSA se encuentra ubicado en el municipio de Montelíbano (Colombia) siendo uno de los mayores productores de ferroníquel a nivel mundial [6], actualmente en su línea de producción cuenta con dos hornos de arco eléctrico de 75MW (Línea 1 y Línea 2), los cuales están equipados con un conjunto de sensores para el monitoreo constante de variables físicas, eléctricas, mecánicas, entre otras. Estas variables se utilizan para llevar a cabo el control operativo, además del monitoreo continuo al funcionamiento y la salud estructural [7].

1.2. Antecedentes y justificación

Diversos han sido los autores que han centrado sus investigaciones en estudios de caso para la aplicación de modelos de inteligencia artificial en hornos de arco eléctrico, los cuales abarcan desde el modelado de características hasta la generación de predicciones sobre algunas variables de interés. Este panorama se puede ver retratado en [8] donde los autores brindan una introducción bastante amplia en su estado del arte, sobre lo que se ha venido trabajando en la utilización de redes neuronales, de igual manera, también se puede abstraer como parte útil de esta investigación, el manejo dado a los datos obtenidos, dividiéndolos en secuencias y ciclos (Batch) para lograr el modelado del comportamiento del arco eléctrico junto con la forma en cómo se llevó a cabo la optimización y selección final de los hiperparámetros para la construcción de la red neuronal. En esta selección final se realizó una optimización a partir del error cuadrático medio estudiando diferentes configuraciones donde se variaba la estructura de la red, el número de capas ocultas y así mismo el número de neuronas por capa.

En la misma línea de investigación, el trabajo descrito en [9] sobre el modelado de consumo de energía en estos hornos, se realiza a partir de modelos de machine learning y deep learning con el objetivo de poder establecer cuáles de ellos tienen mejor comportamiento para lograr los requerimientos planteados y en general para trabajar con sistemas tan complejos, como lo son, en este caso, donde se concluye que los modelos de deep learning tendrán un rendimiento mayor a los modelos convencionales de machine learning. Por último, en este trabajo se resalta la importancia de la recolección de datos, selección de características, el preprocesamiento y la calidad de los datos, que como se mencionaba en el planteamiento del problema, es algo importante a trabajar para la resolución del mismo.

Continuando con la revisión de literatura se encuentra el desarrollo realizado en [10], el cual empieza a acercarse a temas más relacionados al marco de esta investigación, desarrollando un trabajo sobre la predicción de series de tiempo de las características de corriente y voltaje, en donde se tiene una rigurosidad matemática para el diseño y explicación del modelo

Extreme Learning Machine (ELM) empleado, de esta manera, se brinda un paso a paso del proceso y las consideraciones realizadas por los autores para llegar a la obtención de un modelo con rendimientos de alta velocidad y bajos errores respecto a otras técnicas de aprendizaje automático.

Entrando en el campo de la predicción de temperatura en HAE se encuentra la referencia [11], el cual empieza dando una introducción al funcionamiento de este tipo de hornos en la industria del acero con apuntes importantes sobre las diferentes etapas del proceso con un enfoque hacia la productividad y como mejorarla. Posteriormente y siendo de especial interés para el proyecto, los autores centran sus explicaciones sobre la importancia de la adquisición y el preprocesamiento que deben tener los datos para mejorar el rendimiento de los modelos en aplicaciones de la vida real, como lo es la predicción de temperatura, pues dedican una sección especial sobre el tratamiento que deben tener los datos de esta variable de interés.

Por último, el trabajo realizado en [12] presenta una perspectiva de construcción de un modelo de redes neuronales específico enfocado también en la predicción de temperatura en HAE, el autor sin ahondar mucho en el trasfondo matemático centra sus explicaciones en el paso a paso y las decisiones que se tomaron para el diseño de la red, comparando las ventajas y desventajas que presenta respecto a otros modelos.

En general todos los trabajos analizados sobre el uso de aprendizaje profundo en hornos de arco eléctrico concluyen sobre la superioridad de los modelos construidos a partir de redes neuronales, pues el tener una alta complejidad en los procesos del HAE hace que los métodos clásicos sean difíciles de optimizar, lo que conlleva a que no tengan buenos resultados en la práctica. Al ser cada trabajo tan específico en los modelos y datos empleados, no se presentan controversias ni conflictos de interés, cuestión que también se presenta al ser trabajos puramente cuantitativos. De esta misma manera se puede establecer que aún quedan muchas investigaciones por surgir en este campo, pues la gran cantidad de modelos y sus derivaciones existentes pueden ser aplicadas a diversos casos de estudio para cada HAE en específico.

Ahora bien, a nivel de la Universidad Nacional de Colombia, la facultad de ingeniería viene desarrollando actividades de extensión con la empresa CMSA, donde específicamente en la línea de analítica de datos, se han desarrollado tres fases a cargo de varios profesores y bajo la coordinación del director de esta de tesis de maestría. A continuación, se describen algunos de los resultados obtenidos del trabajo conjunto con CMSA en las fases anteriores junto con literatura específica, dando soporte a las conclusiones obtenidas en los artículos publicados sobre trabajos futuros y de esta manera poder dar contexto a la continuidad del trabajo realizado en la fase 4.

El trabajo descrito en la referencia [13] nos presenta el enfoque realizado para llevar a cabo el preprocesamiento correspondiente a un conjunto de datos compuesto por cuatro años de

registros de todas las variables correspondientes al funcionamiento del horno línea 1, en donde se pudieron identificar problemas típicos con la calidad de los datos (Fallos en los procesos de adquisición y presencia de valores faltantes y valores atípicos). De esta manera, en un proceso de limpieza de varias etapas junto con encargados de CMSA, se logró generar un conjunto de reglas que conducen a la detección y limpieza de datos de mala calidad que no representan el comportamiento real del horno.

En los trabajos realizados en [14][15] se llevó a cabo el desarrollo, validación e implementación de diversos algoritmos para la predicción de temperaturas en el horno línea 1 a partir de modelos basados en redes neuronales recurrentes. Con los resultados allí obtenidos se comparó la precisión de las predicciones entre los modelos implementados, destacando las características y el comportamiento que posee cada uno ante diversos casos de estudio. Estas dos investigaciones permitieron determinar la relevancia o importancia de algunas variables para la predicción de temperatura en el horno, lo cual puede ser demostrado a partir de las correlaciones existentes entre las variables de funcionamiento del horno y la temperatura de pared media del mismo.

En [15] plantean la posibilidad de realizar como trabajo futuro la aplicación de los modelos de aprendizaje profundos allí trabajados con la integración de mecanismos de atención, basados en el enfoque codificador-decodificador, con el objetivo de predecir la temperatura del horno considerando las relaciones entre las variables en periodos de tiempo largos y cortos. La implementación de estos mecanismos de atención para la predicción de series de tiempo es relativamente reciente, pues en su mayoría la literatura corresponde a publicaciones realizadas en los tres últimos años, como es el caso del trabajo descrito en [16], realizado en el año 2019, en este se describen los beneficios y características de combinar redes neuronales recurrentes junto con mecanismos de atención aplicados a varios casos del mundo real como la energía solar, pues a partir de la implementación de estos mecanismos se logró modelar la dependencia temporal entre variables a largo plazo, extrayendo de esta manera los patrones temporales invariantes en el tiempo más relevantes para la predicción de las series de tiempo.

Por otro lado, siguiendo con el campo de los mecanismos de atención se encuentra el enfoque codificador-decodificador descrito en [17] en donde se integra el vector de contexto representativo de las estructuras codificadoras con un vector de atención que permitirá al modelo aprender de forma adaptativa las dependencias temporales a largo plazo, extrayendo patrones y correlaciones lineales y no lineales ocultas en los datos. Para este desarrollo las estructuras de codificación y decodificación se basaron en estructuras de memoria a corto plazo “LSTM” empleando como prueba cinco conjuntos de series de tiempo multivariantes, para los cuales se obtuvieron resultados con mejor rendimiento sobre los modelos tradicionales de aprendizaje profundo. En el desarrollo realizado en [18] los autores buscan obtener un equilibrio entre la complejidad que conlleva la implementación de estos modelos con la precisión que se pueda obtener en las predicciones, para esto emplean un enfoque secuencial

(SEQ-to-SEQ) que logra minimizar la dependencia de la predicción de los datos no periódicos en la serie de tiempo. Un conjunto de datos sobre la polución de aire en una ciudad fue empleado para comparar los resultados obtenidos frente a diversas estructuras de redes neuronales basadas en LSTM.

Por último, el trabajo descrito por los autores S. Huang, D. Wang, X. Wu, y A. Tang en [19] plantea un nuevo enfoque para el manejo de los mecanismos de atención, logrando resolver así los problemas que presentaban estos mecanismos ante patrones altamente dinámicos, pues propone el uso de una red dual de atención, es decir, compone los modelos ya establecidos en una estructura paralela que permite trabajar con patrones globales y locales por igual entre múltiples series de tiempo. Este trabajo se centra en la robustez del modelo diseñado, por lo que integra un tercer componente paralelo a la red dual, en esta ocasión será un modelo lineal autorregresivo.

De los trabajos anteriores se puede destacar como cada uno de ellos presenta un nuevo enfoque a partir de los modelos con mecanismos de atención sobre los modelos tradicionales, aunque hayan sido construidos a partir de diferentes sistemas y estructuras neuronales, por lo que al ser empleados en diferentes situaciones de la vida real se comportan de manera robusta y precisa, lo cual abre un mundo de posibilidades para seguir realizando investigaciones y desarrollos en este campo novedoso de las series de tiempo.

1.3. Objetivos

1.3.1. Objetivo general

Realizar un estudio comparativo de modelos de aprendizaje profundo basados en redes neuronales recurrentes (RNN) para la predicción de series de tiempo multivariantes de temperatura de pared media en los hornos de arco eléctrico línea 1 y 2 en Cerro Matoso S.A.

1.3.2. Objetivos específicos

1. Caracterizar las variables asociadas al funcionamiento de los hornos línea 1 y línea 2 de Cerro Matoso S.A mediante un análisis exploratorio de datos (EDA).
2. Desarrollar la estimación de las variables de temperatura de pared media en los hornos de Cerro Matoso S.A usando algoritmos de aprendizaje profundo para obtener al menos dos modelos basados en datos.
3. Estudiar el desempeño y comportamiento de al menos dos modelos de estimación obtenidos en diversos escenarios a partir de su validación en un conjunto de prueba empleando el error cuadrático medio (RMSE) como métrica de evaluación.

1.4. Resultados generales

Objetivo específico 1: Caracterizar las variables asociadas al funcionamiento de los hornos línea 1 y línea 2 de Cerro Matoso S.A mediante un análisis exploratorio de datos (EDA).

En la sección 4 mediante un flujo de trabajo estructurado se llevó a cabo el análisis exploratorio y la limpieza de datos para los conjuntos de datos brindados por CMSA de los hornos línea 1 y línea 2, donde se logró la caracterización y selección de las variables más relacionadas con la temperatura en las paredes medias de los hornos, las cuales permitieron la construcción del conjunto de datos a ser usado por los modelos de aprendizaje profundo desarrollados más adelante. Por último, se definieron parámetros para el entrenamiento de modelos que serán usados más adelante pero que están relacionados directamente con el conjunto de datos, como lo son el tiempo de predicción y el porcentaje de división entre conjuntos de entrenamiento y prueba.

Objetivo específico 2: Desarrollar la estimación de las variables de temperatura de pared media en los hornos de Cerro Matoso S.A usando algoritmos de aprendizaje profundo para obtener al menos dos modelos basados en datos.

A lo largo de la sección 5 se llevó a cabo la construcción, compilación y entrenamiento de ocho modelos de estimación basados en datos correspondientes a los hornos línea 1 y línea 2, donde se continuo con la definición de parámetros importantes para el entrenamiento de los modelos, estableciendo la métrica RMSE como función de costo para estudiar los resultados obtenidos en cada uno de los modelos y de esta manera poder realizar un ajuste a los hiperparámetros de cada uno de los modelos para mejorar su desempeño. Por último, se observaron y compararon las predicciones obtenidas para cada uno de los modelos tanto del horno línea 1 como del horno línea 2.

Objetivo específico 3: Estudiar el desempeño y comportamiento de al menos dos modelos de estimación obtenidos en diversos escenarios a partir de su validación en un conjunto de prueba empleando el error cuadrático medio (RMSE) como métrica de evaluación.

A partir de los ocho modelos obtenidos cuatro respectivamente para cada uno de los hornos de CMSA, se estudió el desempeño y comportamiento de los mismos ante variaciones en las condiciones para validar y comparar su funcionamiento con un mismo conjunto de prueba. Como primer caso de estudio se analizó la influencia de tiempos de predicción cortos a tiempos de predicción largos, siguiendo con el aumento de variables de temperatura a predecir. Posteriormente se realizó una validación cruzada para estudiar el efecto entre la separación de tiempo que existirá entre el conjunto de datos con el que se entrenó el modelo y la información actual con la cual está prediciendo el modelos y por último, se obtuvo la distribución promedio de la función de costo de manera aislada para cada una de las termocuplas en un modelo de predicción tanto para el horno línea 1 como para el horno línea 2.

2. Marco teórico

2.1. Horno de arco eléctrico

El proceso para la producción de ferróníquel en CMSA se lleva a cabo en diferentes pasos, se inicia con la extracción del material de la mina el cual es triturado en partes mas pequeñas, a este proceso se le conoce como homogeneización, esto se realiza para facilitar el secado y almacenamiento del mismo. Este material una vez listo se traslada a un calcinador de horno rotatorio, arrojando calcina como material final a la salida del horno, posteriormente este material ingresara a fundición en los hornos de arco eléctrico a través de varios conductos aéreos, los cuales se dividen en tres zonas, central, semicentral y lateral. Una vez el material se encuentre fundido se retira del horno utilizando dos salidas a diferentes niveles del horno, una para ferróníquel y otra para escoria. El siguiente paso en el proceso es el refinamiento y granulación del material. La figura **2-1** muestra una imagen del edificio donde se encuentran los dos hornos. Las dimensiones de cada horno son de 22 m de diámetro y 7 m de altura [20][21].

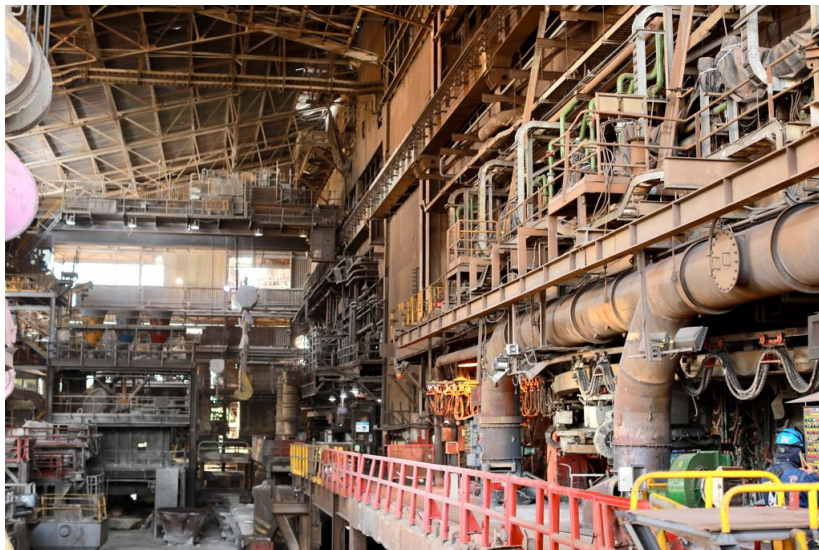


Figura 2-1.: Planta CMSA [21].

La etapa principal de la producción de ferróníquel es la fundición, esta se lleva a cabo en el EAF. La figura **2-2** muestra una vista interior del horno de fundición, detallando sus partes:

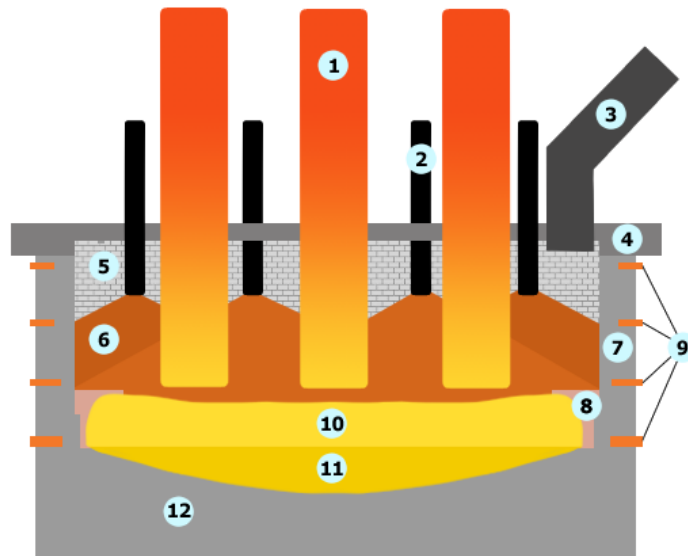


Figura 2-2.: Ubicación de partes y componentes básicos en un horno de arco eléctrico [21].

1. Electrodo
2. Tubos de alimentación
3. Chimenea de escape
4. Techo superior
5. Pared trasera
6. Entrada de calcina
7. Pared lateral
8. Waffle coolers
9. Plate coolers (Termocuplas)
10. Escoria
11. Ferróniquel fundido
12. Revestimiento (Base) del horno

2.1.1. Termocuplas

En los hornos a nivel industrial, el dispositivo de medición de temperatura más empleado debido a su alta precisión y tolerancia a altas temperaturas es la termocupla o termopar, el cual dependiendo de su construcción puede llegar a medir temperaturas desde los $-270\text{ }^{\circ}\text{C}$ hasta los $1815\text{ }^{\circ}\text{C}$, como se observa en la tabla **2-1**.

Tipo	Color	Rangos de funcionamiento recomendados
B	Gris/Rojo	870 - 1700 °C
E	Morado/Rojo	-200 - 870 °C
J	Blanco/Rojo	0 - 760 °C
K	Amarillo/Rojo	-200 - 1260 °C
N	Naranja/Rojo	0 - 1260 °C
R	Verde/Rojo	0 - 1480 °C
S	Verde/Rojo	0 - 1480 °C
T	Azul/Rojo	-200 - 370 °C
C	Verde/Rojo	0 - 2315 °C

Tabla 2-1.: Estándar ISA/ASTM para la clasificación de termocuplas [22].

Las termocuplas listadas en la tabla **2-1** no se diferencian únicamente en el rango de medición de temperatura, pues se pueden analizar ciertas características debidas a su diseño que las hacen mejores para ciertas aplicaciones, por ejemplo, las termocuplas tipo E son más resistentes al ruido, mientras que termocuplas tipo J cada vez son menos empleadas debido a ser propensas de oxidación dado por su construcción en hierro. En los hornos de arco eléctrico los tipos de termocuplas que se emplean en su mayoría son tipo K y N, debido a su rango extendido de medición [23].

2.1.2. Red de termocuplas hornos CMSA.

El desarrollo de este trabajo se centra en la monitorización y previsión de la temperatura de paredes laterales, la cual se mide mediante una red de termocuplas distribuidas radialmente en cuatro niveles (A, B, C y D) a lo largo de los enfriadores de placas de la pared lateral en cuatro zonas (NO, NE, SO y SE) en los hornos línea 1 y línea 2 como se observa en la figura **2-3**, las termocuplas (Zonas verdes) se encuentran distribuidas de manera secuencial alternando entre paneles con 2 termocuplas en los niveles B y D y paneles con 3 termocuplas en los niveles A, C y D [20].

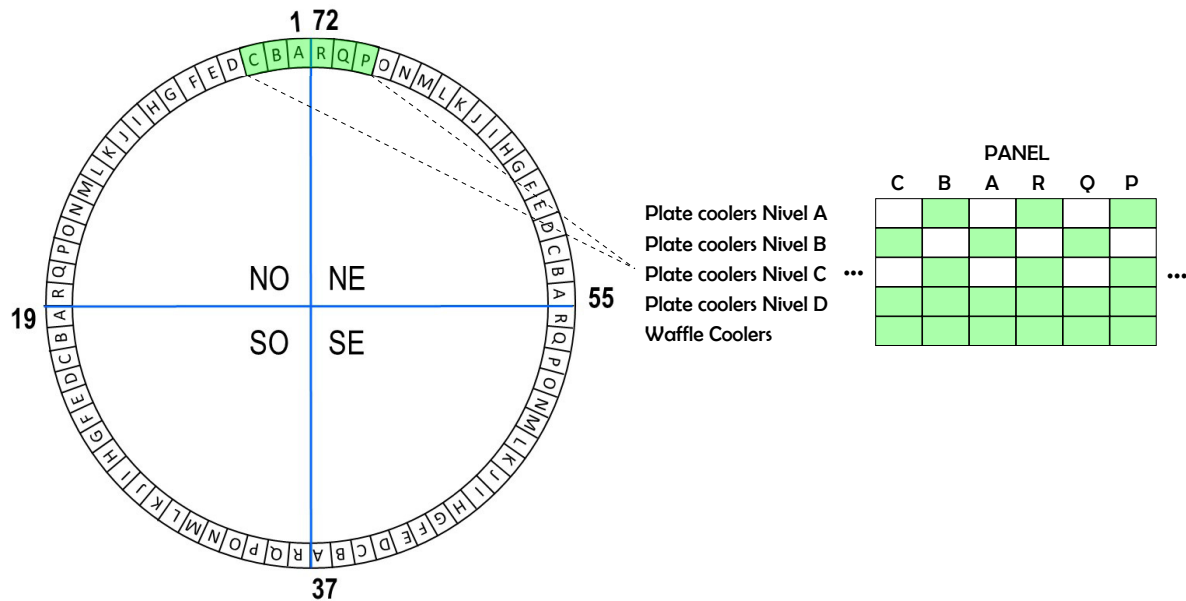


Figura 2-3.: Distribución de paneles por zonas en los hornos línea 1 y línea 2 de CMSA.

2.2. Series de tiempo

En los problemas que se pueden presentar en la vida real y que pueden ser trabajados con inteligencia artificial, se encuentran las series de tiempo, las cuales son un conjunto de datos ordenado y estructurado a lo largo del tiempo, es decir, el tiempo será una variable importante en el análisis siendo una variable totalmente independiente, el objetivo principal de trabajar con este tipo de datos es realizar predicciones a futuro y de esta manera poder simular posibles escenarios o comportamientos del problema. El tiempo no es el único factor que tiene impacto en este tipo de problemas, pues existen ciertas características propias de las series temporales como se observa en la figura 2-4, que son importantes conocer para poder trabajar con ellas [24], tal como se muestra en las siguientes subsecciones.

2.2.1. Componentes de las series de tiempo

Autocorrelación

La primera característica presente en las series de tiempo y que brinda las bases para algunas otras características, es la autocorrelación, la cual expresa la dependencia que puede llegar a existir entre los datos de la serie, es decir, la correlación que tiene la serie en dos puntos diferentes de tiempo, lo que puede ser visto como la similitud existente entre la serie y una versión retrasada de la misma, lo cual además se emplea para identificar la estacionalidad y la tendencia en los datos en la serie de tiempo [24][25].

Estacionalidad

La estacionalidad es una característica presente en algunas series de tiempo relacionada con la periodicidad o frecuencia de aparición que pueden llegar a tener algunos comportamientos en la serie a lo largo del tiempo, es decir, el comportamiento de la serie se repite de manera aproximada en periodos de tiempo iguales (Días, semanas, meses, entre otros.). Esta característica se puede inferir a partir de la autocorrelación, pues gracias a ella se sabe que, si varios valores de la serie están correlacionados en diferentes periodos de tiempo de igual duración, existe algún tipo de componente estacional en los datos [24][25].

Tendencia

La estacionalidad evalúa efectos entre ventanas cortas de tiempo a lo largo de la serie, sin embargo, a largo plazo puede existir una tendencia a un comportamiento general de los datos ignorando su comportamiento en el corto plazo, por ejemplo, los datos de una serie que tienden a aumentar de manera gradual sin importar las pequeñas variaciones que puedan existir entre pasos de tiempo cercanos [24][25].

Estacionariedad

Una característica muy única en las series de tiempo ocurre cuando a lo largo del tiempo se conservan las propiedades estadísticas (Media, varianza y covarianza), es decir, son independientes del paso del tiempo [24].

Ruido

Por último, como todo sistema o problema en la vida real, tiene un factor de variación aleatorio debido a factores externos del mismo, las series temporales no son la excepción pues el conjunto de datos a lo largo del tiempo puede presentar estas variaciones aleatorias conocidas como ruido.

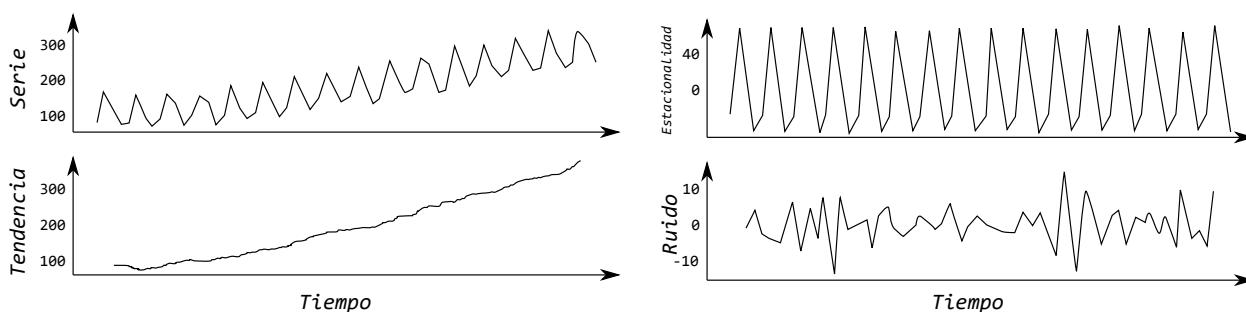


Figura 2-4.: Descomposición de los componentes de una serie de tiempo.

Un campo de estudio de especial interés hoy en día con el despliegue de la inteligencia artificial y el aprendizaje profundo en aplicaciones de series de tiempo a nivel industrial es

la predicción del comportamiento de las mismas, lo cual conlleva muchas ventajas como el conocimiento a futuro de un proceso para la evaluación de diferentes escenarios y la toma de decisiones [25].

2.3. Aprendizaje profundo

El aprendizaje profundo es un rama o subconjunto de estudio en el campo de la inteligencia artificial, similar en algunos aspectos a otros algoritmos de aprendizaje automático, sin embargo, difieren en cuanto a sus capacidades, pues en los algoritmos tradicionales de aprendizaje automático, la mayoría de las características en los datos deben ser identificadas por un experto en la materia para reducir la complejidad y hacer más visibles los patrones para que funcionen los algoritmos, lo cual puede llegar a ser un proceso lento debido a que se realiza de manera manual y no siempre todas las características identificadas son de gran utilidad. Aquí es donde aparece la mayor ventaja de los algoritmos de aprendizaje profundo, pues estos son capaces de aprender patrones ocultos de los datos sin supervisión humana y a partir de esto intentar construir características de los datos de forma incremental para establecer reglas de decisión mucho más rápidas y eficientes. Estos algoritmos basan su funcionamiento en algo que se conoce como redes neuronales inspiradas en el funcionamiento de las neuronas del cerebro humano [26].

2.3.1. Redes neuronales

Las redes neuronales son el núcleo o eje central para la construcción de los algoritmos de aprendizaje profundo, estas redes son un conjunto de elementos interconectados (Red) que basan su estructura y funcionamiento en las formas de comunicación entre las neuronas del cerebro humano (Neuronal) [27]. La función principal de una red neuronal es producir un patrón de salida cuando se le presenta un patrón de entrada, a través de una red que se basa en los datos para realizar distintas tareas como aprender y extraer patrones, clasificar conjuntos de datos, predecir eventos futuros en series de tiempo y tareas más complejas como el reconocimiento de voz y la asistencia para vehículos autónomos, teniendo la característica de realizar estos procesos de una manera más rápida y eficaz que otros algoritmos de aprendizaje automático [28][29].

2.3.2. Arquitectura

La estructura interna de las redes neuronales se basa en la conexión entre elementos, a cada uno de estos elementos o “neuronas” de manera individual se le conoce como nodo, si se tiene una agrupación de estos de manera estructurada se le conoce como capas, las cuales a su vez conectadas entre sí formarían la red neuronal, de manera que cada capa utiliza la

salida de la capa anterior como entrada [30]. Estas relaciones se pueden observar en la figura 2-5, donde se listan los tres tipos básicos de capas:

- **Capa de entrada:** Esta capa es la primera de todas, la capa donde entrará la información proveniente de las características del conjunto de datos con el que se esté trabajando, así mismo este conjunto de características será el que define el tamaño de nodos que compondrá la capa.
- **Capas ocultas:** Son todas las capas que se ubican entre la capa de entrada y de salida, se puede establecer que son capas diseñadas para producir una salida específica para un resultado previsto (Transformación de datos), mediante la asignación de pesos y funciones de activación específicas. El número y profundidad de estas capas es asignado de manera manual dependiendo de la aplicación hasta lograr un resultado óptimo a través del ajuste de los hiperparámetros (Tasa de aprendizaje, cantidad de épocas de entrenamiento, tamaño de Batch, entre otros.).
- **Capa de salida:** Esta última capa se encuentra al otro extremo de la capa de entrada, esta será la encargada de obtener un resultado final a partir de la información brindada por las capas anteriores, donde realizará los cálculos a través de sus neuronas y luego calculará la salida.

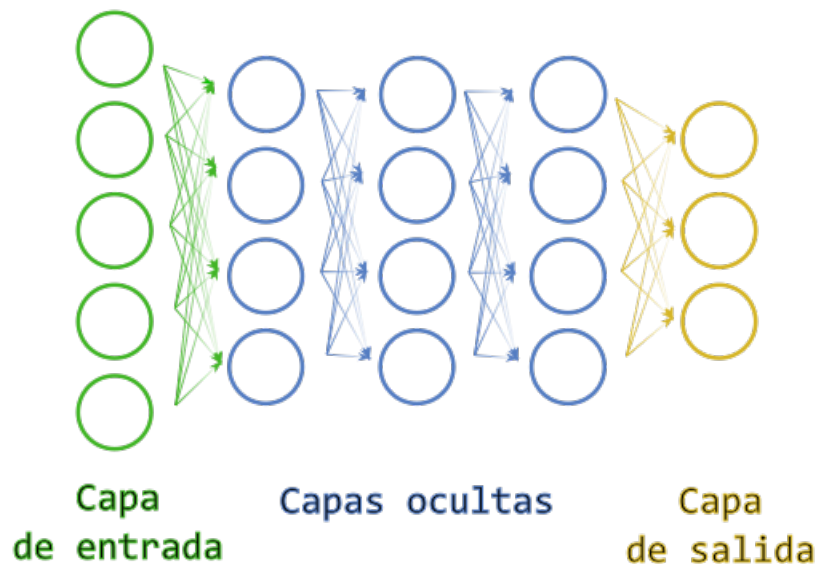


Figura 2-5.: Arquitectura típica de un red neuronal.

Los nodos no son lo único para tener en cuenta al momento de construir una red neuronal, pues la forma en como las “neuronas” manejan y toman decisiones de la información proveniente de otros nodos también es importante, lo cual se realiza a partir de dos términos fundamentales: Peso y función de activación.

Peso:

Entre cada par de nodos existe una conexión única, en la cual el nodo de la capa anterior transmitirá su información contenida al nodo conectado de la capa siguiente, sin embargo, existe información más valiosa que otra a lo largo del trabajo realizado por la red neuronal, es por esto que a cada conexión se le asigna un parámetro conocido como peso, que no es más que un valor escalar constante por el cual se multiplicara el valor actual del nodo y esta información será la que hará de entrada para la siguiente neurona. En un principio estos valores se asignan de manera aleatoria para observar la salida obtenida al final de la red neuronal, a partir de esto, cada uno de ellos se actualiza dependiendo la importancia (Peso) que tenga cada uno de ellos para obtener la predicción final, esto se realiza de manera automática a través de un mecanismo conocido como retro propagación. Es decir, que cada neurona en la capa de entrada tendrá un valor asignado por el conjunto de datos, el cual se multiplicara por su respectivo peso asignado y la suma de estos mismos producirán el valor de la neurona conectada en la siguiente capa [31].

En la figura 2-6 se desglosa la relación entre los nodos de la capa de entrada con el primer nodo de la capa oculta, donde las x_n representan cada una de las características del conjunto de datos, los w_n corresponden a los pesos de cada uno de los nodos de la capa de entrada al primer nodo de la capa oculta, de esta manera, el valor del primer nodo oculto n_1 será $x_1w_1 + x_2w_2 + x_3w_3$ y se extrapola este procedimiento para calcular el valor de cada uno de los demás nodos de cada capa oculta existente, así como de los nodos de la capa de salida, sin embargo, falta la inclusión de la función de activación para realizar un análisis completo de cómo se transmite de forma básica la información en una red neuronal.

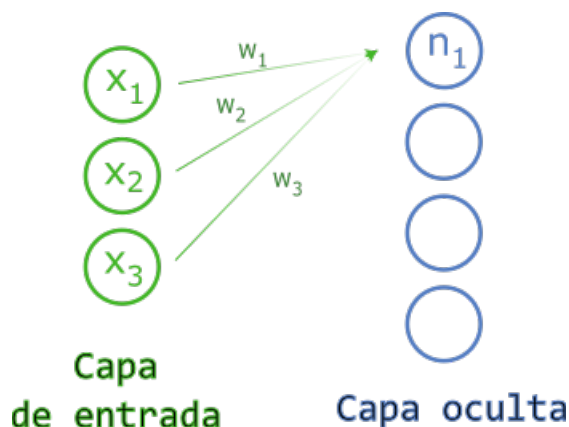


Figura 2-6.: Entradas, pesos y nodos de capas ocultas en una red neuronal.

Función de activación:

Hasta el momento lo que se ha descrito para las redes neuronales obedece a comportamientos y procedimientos puramente lineales, pues operaciones solo entre pesos y nodos con valores

escalares harán que la salida responda a una función lineal, es decir, sería un modelo más de regresión lineal limitado por su propia estructura, que si bien es fácil de resolver, presenta muchas limitantes al momento de aprender y extraer información y patrones de los datos para resolver problemas complejos y no lineales del mundo real.

Para resolver este problema en las redes neuronales se introdujeron las funciones de activación, las cuales son un hiperparámetro que tiene como objetivo principal introducir propiedades de análisis no lineal en el cómputo de los valores que tendrán los nodos dentro de las capas ocultas y de salida, es decir, buscan ayudar a la red neuronal a encontrar y dar sentido a patrones dentro de conjuntos de datos con comportamientos complejos y probablemente no lineales. Todo empieza desde la capa de entrada a la primera capa oculta donde se convierten las entradas lineales en salidas no lineales a las demás capas ocultas, lo que permite la aparición de polinomios de mayor orden para que la red neuronal pueda realizar un aprendizaje más profundo. Las funciones de activación más empleadas además de ser no lineales son diferenciables, lo cual es fundamental para el funcionamiento del proceso de retro propagación, pues a partir de los diferenciables es que surgen los gradientes y los errores necesarios para poder modificar los pesos entre nodos [32]. Las funciones de activación más empleadas son:

- **Función de activación logística (Sigmoide):** Esta función de activación entra en la categoría de funciones de activación no lineales y diferenciables, tiene un rango de valores posibles entre 0 y 1, lo cual hace que sea bastante útil para predecir salidas basadas en probabilidad, pues la probabilidad de cualquier suceso en la vida real sólo existe entre el 0 y 1. Tiene una forma característica de ‘S’ como se puede observar en la figura 2-7 lo cual simboliza el fenómeno de muchos procesos naturales a tener un periodo de progresión baja al inicio y en algún momento tener un gran cambio o crecimiento acelerado para crecer y nuevamente llegar a un punto estable en el tiempo. Generalmente este tipo de función aparece en la capa de salida de los modelos de aprendizaje profundo, sin embargo, posee algunos inconvenientes relacionados con el gradiente y la velocidad de convergencia y tener una salida centrada en un valor distinto de cero causa que actualizaciones del gradiente se propaguen en direcciones diferentes [33].
- **Función de activación de tangente hiperbólica (Tanh):** A diferencia de la función de activación anterior esta función se encuentra centrada en cero, tomando valores entre un rango entre -1 y 1 como se observa en la figura 2-7, lo que hace que tenga un comportamiento más suave en la región de cambio entre el mínimo y máximo, se emplea con mayor frecuencia que la función sigmoide debido a su mayor rendimiento en el entrenamiento de redes neuronales multicapa, pues su principal ventaja se debe a que produce una salida centrada en cero para valores de entrada cercanos a cero, mientras castiga los valores negativos alejados del cero permitiendo un proceso de retro propagación más óptimo [33].

La función tanh también presenta ciertos inconvenientes y limitaciones dados por el gradiente al igual que la función sigmoide, adicionalmente, su gradiente únicamente puede alcanzar valores de 1 cuando su entrada es cero, lo que causa que algunos nodos queden inutilizados durante el proceso de cálculo [34].

- Función de activación de unidad lineal rectificada (ReLU):** Por último, una de las funciones de activación más conocidas y empleadas en redes neuronales de aprendizaje profundo, que a diferencia de las anteriores no tiene un rango acotado a la salida, pues para valores menores que cero genera una salida negativa y para valores iguales o mayores a cero la entrada se mantiene igual como se observa en la figura 2-7, al trabajar de esta manera con los valores negativos esta función elimina el problema del gradiente observado en los anteriores tipos de función de activación (Sigmoide y tanh), por lo tanto ofrece un rendimiento mayor en la etapa de aprendizaje pues su comportamiento casi lineal hace que sea fácil de optimizar con métodos de ascenso de gradiente [35][36].

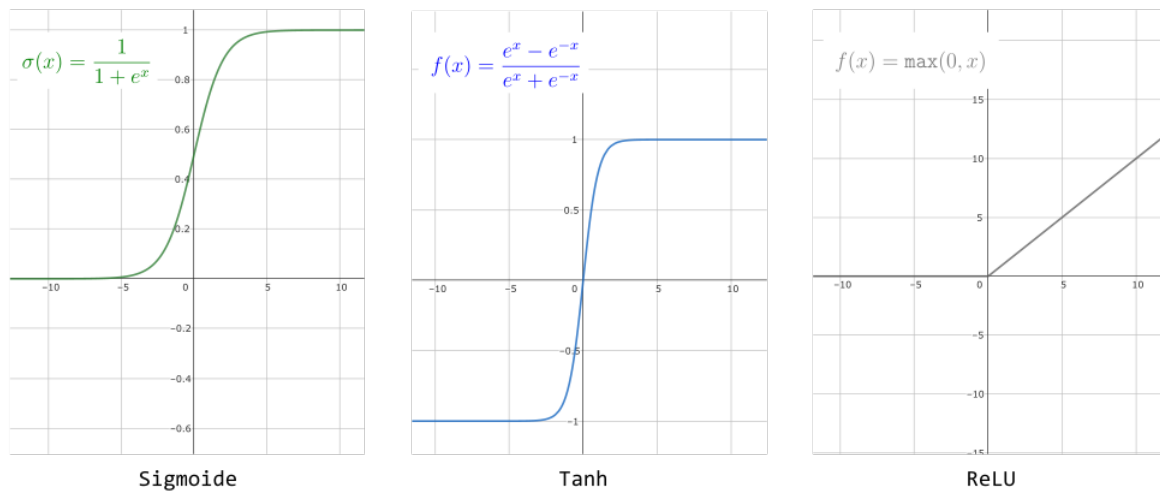


Figura 2-7.: Funciones de activación Sigmoide, Tanh y ReLU.

2.3.3. Costo y funciones de costo

Se ha profundizado en la arquitectura interna de una red neuronal, desde cómo se construye hasta cómo funciona, donde su principal objetivo es realizar buenas predicciones sobre un conjunto de datos de entrada nuevo, la medida para cuantificar que tan bien predicen dos modelos diferentes sobre un mismo problema, es decir, que tanto difieren los valores predichos de los valores reales en cada uno de los modelos, se realiza a partir de las funciones de pérdida y costo, que si bien parecen cosas iguales tienen ciertas diferencia. La función de pérdida se

emplea cuando se va a evaluar una única entrada del conjunto de entrenamiento, mientras que la función de costo evalúa la pérdida para todo el conjunto de datos de entrenamiento, por lo tanto, aquí es donde se centra el estudio de todas las estrategias de optimización, buscando minimizar cada vez más el costo de cada modelo a través de un método conocido como descenso por gradiente [37].

En la práctica cuando se trabajan problemas complejos no existe una función de coste ideal para trabajar dependiendo de factores como la entrada de datos, la red neuronal seleccionada e incluso el mismo objetivo que se quiere alcanzar, sin embargo, si se pueden clasificar en dos grandes grupos: las empleadas en problemas de clasificación y las empleadas en problemas de regresión, siendo estas últimas bastante utilizadas en problemas de predicción de series de tiempo.

Error Medio - ME

Es la función de costo más básica que existe, la cual sirve como punto de partida para construir funciones más complejas y recomendadas de usar. Esta calcula el error para cada dato de entrenamiento y posteriormente halla la media entre todos estos valores de error:

$$ME = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)$$

Al emplear esta función de costo surge un problema, pues el error puede ser tanto positivo como negativo, lo que puede generar cancelaciones entre estos valores disminuyendo el valor que se obtenga para el error medio, lo cual no significa que las predicciones sean más adecuadas [37].

Error Cuadrático Medio - MSE

Para solucionar el problema del error negativo existente anteriormente se eleva al cuadrado la diferencia entre las predicciones y los valores reales para nuevamente ser promediado:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Ahora bien, esta función de costo conlleva consigo un problema frente a los valores de error muy grandes o atípicos, pues el error al estar dentro de un término cuadrado y presentar un valor grande causara que el error crezca mucho más en comparación a los valores pequeños que si son menores que uno serán penalizados, lo cual llevara a que el modelo tenga un MSE mayor, por lo tanto, no es una función robusta frente a valores atípicos [37].

Error Absoluto Medio - MAE

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

Para solucionar los problemas que poseen las dos funciones de coste anteriores, se reemplazó el termino cuadrático por un valor absoluto, el cual nos dará la magnitud media de todos los errores sin importar su signo, solucionando así el problema del signo en ME y siendo robusto frente a valores atípicos, pues ya no aumentará en gran manera los errores grandes ni penalizará los errores pequeños como la función MSE [38].

Raíz del Error Cuadrático Medio - RMSE

Por último, tenemos la función de coste que se forma a partir de la raíz cuadrada de la función MSE, que como ya se había establecido da más importancia a los errores grandes, sin embargo, los errores al cuadrado se promedian dentro de la raíz, lo que supone una gran penalización en los errores grandes. Esto implica que RMSE es útil cuando no se desean errores grandes [38].

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

2.3.4. Propagación hacia delante, descenso por gradiente y retro propagación

Propagación hacia delante

El cálculo de información dentro de una red neuronal, así como de la salida que esta genera, se lleva a cabo por un algoritmo con sus pasos bien definidos, el primero de ellos es la propagación hacia adelante, el más sencillo de todos, donde a partir de la entrada de datos obtenemos un resultado, yendo desde la capa de entrada, pasando por las capas ocultas y terminando en la capa de salida como se observa en la figura **2-8**.

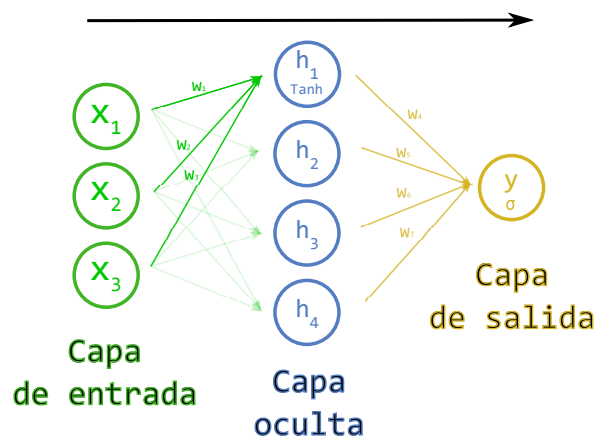


Figura 2-8.: Propagación hacia adelante en las capas de una red neuronal.

En la figura 2-8 se tiene una red neuronal simple con una única capa oculta, para la cual se calcula un nodo de la capa de salida a partir de los tres nodos de la capa de entrada, se calcula el primer nodo de la capa oculta h_1 que tiene una función de activación **Tanh**:

$$h_1 = \text{Tanh}(x_1w_1 + x_2w_2 + x_3w_3)$$

Este proceso se repite para cada uno de los nodos de la capa oculta a partir de las entradas y sus respectivos pesos, una vez obtenidos todos los valores se procede a calcular la salida y :

$$y = \sigma(h_1w_4 + h_2w_5 + h_3w_6 + h_4w_7)$$

Y por último se aplica una función de pérdida entre la salida predicha y el valor real para calcular el error, el cual es importante para la red neuronal, ya que obtiene información de que tan precisas son las predicciones realizadas y si es el caso realizar una actualización de los pesos a través de técnicas como el descenso por gradiente y la retro propagación[40].

Descenso por gradiente y retro propagación

El descenso por gradiente es uno de los algoritmos más empleados para entrenar y optimizar redes neuronales, el cual nos permite calcular los ajustes necesario a realizar para los parámetros de la red neuronal con el objetivo de minimizar la función de costo. Una vez calculada la salida y la función de costo a partir de la propagación hacia adelante, se calcula el gradiente de la misma a partir de la derivada multivariable, el cual matemáticamente brinda información sobre la dirección en el que la función se maximiza más rápido, por lo tanto, se trabaja con la dirección inversa del gradiente, lo cual ayudara al algoritmo a encontrar un mínimo [41]. Los pesos se actualizan de la siguiente manera:

$$W'_i = W_i - \gamma \frac{\partial L}{\partial W_i}$$

Donde \mathbf{W}'_i corresponde al peso actualizado, \mathbf{W}_i es el peso actual y el termino restante es el gradiente de la función de pérdida \mathbf{L} respecto a los pesos multiplicado por la tasa de aprendizaje γ , la cual brinda una medida de cuanto debe ser el impacto del gradiente en la actualización de los pesos. En problemas de la vida real se tienen funciones de costos complejas debido a los múltiples parámetros y cantidad de capas que componen la red, por lo que el cálculo del gradiente deja de ser algo trivial que además puede llevar a la red a un mínimo local y no al mínimo global, el cual es el objetivo de minimizar la función de costo, para esto se sigue con el siguiente paso del algoritmo que realiza un red neuronal, la retro propagación que basa sus principios en el descenso por gradiente, esta es la encargada de hacer que la red neuronal aprenda a mejorar sus predicciones tras cada actualización de parámetros [41].

La retro propagación empieza calculando las derivadas parciales de la función de costo respecto a los pesos de la capa oculta anterior a la capa de salida, una vez obtenidas se realiza el mismo calculo pero ahora con los pesos de la capa inmediatamente anterior y de esta manera progresivamente propagando el cálculo hacia atrás hasta llegar a la capa de entrada de la red neuronal, a partir de esto se construye un vector de gradiente, para aplicar la fórmula de descenso de gradiente establecida anteriormente, multiplicando por la tasa de aprendizaje y restándolo del valor actual de los pesos, este procedimiento se repite determinadas veces hasta que el algoritmo detecte que la función de costo y los pesos están empezando a aumentar de forma sostenida en vez de disminuir [39][42].

2.3.5. Hiperparámetros en redes neuronales

Las redes neuronales son una estructura que se compone y construye a partir de múltiples variables (Parámetros e hiperparámetros) establecidas a lo largo de sus nodos y capas, los parámetros son aquellas variables que son aprendidas por la red durante el entrenamiento a partir de los datos, mientras que los hiperparámetros son un conjunto de variables encargadas de establecer la estructura y el entrenamiento de la red para obtener los mejores resultados posibles, por esto su optimización es una tarea importante dentro del trabajo con redes neuronales que se debe realizar de manera manual, ya que definen cosas como el costo computacional, el tiempo que se tarda el modelo en entrenar así como la precisión y la capacidad de abstracción y generalización que este tendrá [43]. Existen diversos tipos y clasificaciones para los hiperparámetros de una red neuronal como se enuncia en la figura 2-9.

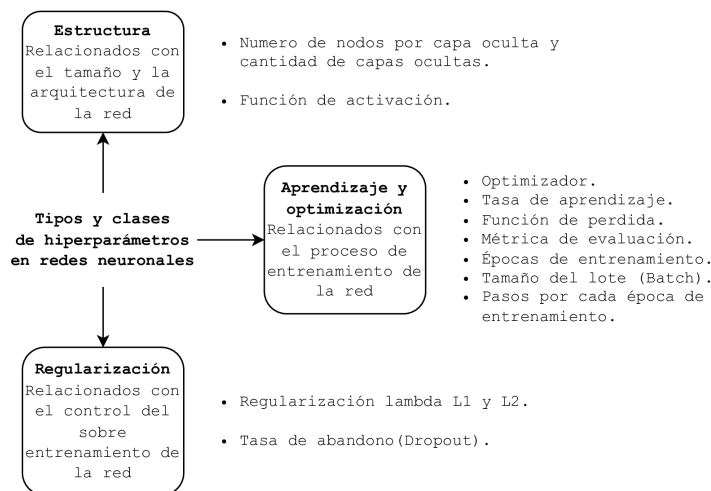


Figura 2-9.: Clasificación de hiperparámetros más comunes

Anteriormente se había descrito la función de activación respecto a la estructura de una red neuronal, sin embargo, no es el único hiperparámetro que cumple esta función, pues se tiene el siguiente hiperparámetro:

Cantidad de capas ocultas

Cuando se habla de redes de inteligencia artificial profundas, este hiperparámetro es el motivo principal, pues a partir de este se define la “Profundidad” de la red, que no es más que el número de capas ocultas existentes entre la entrada y la salida de la red, el cual es un factor importante al momento de trabajar con problemas de la vida real debido a la complejidad de los mismos, pues la red debe contar con un número suficiente de capas oculta para poder aprender y extraer los patrones complejos y no lineales provenientes de los datos, pues de lo contrario la red no se ajustara lo suficiente a las dinámicas de los datos y será ineficiente para llevar acabo las predicciones con datos nuevos, sin embargo, muchas capas tampoco es la solución, pues aumentar de manera arbitraria el número de capas conlleva a que la red neuronal se sobreajuste a los datos con los que está siendo entrenada y más allá de aprender los patrones y las dinámicas que existen entre estos, la red tendera a solo memorizarlos y al momento de realizar predicciones a partir de datos nuevos falla por su falta de generalización frente a los datos. Siempre se evita llegar a alguno de los dos extremos, tanto el desajuste como el sobreajuste son no deseados, sin embargo, el sobreajuste puede tratarse o en algunas situaciones eliminarse a partir de métodos de regularización adecuados, aunque tener una red más profunda también conlleva a que el procesamiento de los datos sea más lento y computacionalmente más pesado.

Numero de nodos:

Una vez definida la profundidad que tendrá una red neuronal, se define el ancho que tendrá la misma, es decir la cantidad de nodos que realizaran el procesamiento en cada capa oculta de la red, lo que también está directamente relacionado con la capacidad que tendrá la red para aprender, similarmente a como sucede con el número de capas ocultas, tener demasiados nodos por capa llevara a la red a un sobreajuste de los datos de entrenamiento. Se dice que este hiperparámetro esta referenciado a las capas ocultas pues el número de nodos en la capa de entrada depende de la dimensionalidad en los datos de entrada mientras que en la capa de salida depende del problema con el que se esté trabajando, pues en problemas de clasificación binaria y regresión, se emplean un único nodo en la capa de salida, mientras que para problemas de clasificación multiclase o multietiqueta se utilizan n nodos dependiendo de las n clases que se tengan en el problema [44].

Los principales hiperparámetros dados al entrenamiento y aprendizaje de la red neuronal son:

Tasa de aprendizaje:

La tasa de aprendizaje es uno de los hiperparámetros más importantes, pues está directamente relacionado con la optimización de los parámetros de la red, define la rapidez con la que se realiza esta acción, es decir, una tasa de aprendizaje baja hace que el proceso de actualización sea muy lento, lo que en caso de ser posible asegura una convergencia del sistema, mientras que una tasa de aprendizaje alta hace que el proceso de actualización de los parámetros sea muy rápido, pero con el riesgo muy alto de que la red no converja nunca. Se sabe que el optimizador de una red neuronal realiza pequeños pasos en el descenso por la curva de error buscando la convergencia, el tamaño de estos pasos es lo que se conoce como la tasa de aprendizaje donde su dirección está determinada por el gradiente. En más detalle al aumentar el valor de la tasa de aprendizaje se aumenta la velocidad de entrenamiento a la par del riesgo que la función de perdida que se desea optimizar oscile alrededor del mínimo y nunca descienda a él o si es muy grande esta oscilación aparezca el problema del gradiente de explosión, como se observa en la figura **2-10**, lo que significa que el modelo nunca lograra ser entrenado, por el contrario, al disminuir el valor de la tasa de aprendizaje el tiempo de entrenamiento del modelo aumenta, ya que el descenso entre paso y paso en la función de perdida será muy poco como se observa en la figura **2-10**, lo cual logra asegurar la convergencia del sistema en caso que exista, pero será un proceso ineficiente que corre el riesgo alcanzar el problema del gradiente de fuga [45][46].

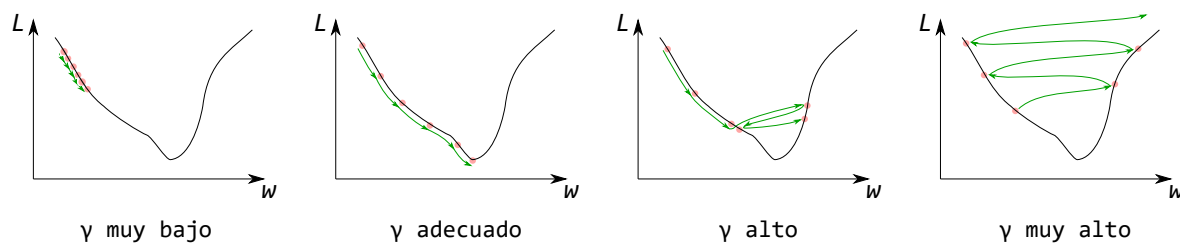


Figura 2-10.: Comportamiento del descenso por gradiente en la función de costo L a partir de la tasa de aprendizaje γ .

Numero de épocas de entrenamiento:

Una forma de controlar el entrenamiento de una red neuronal recae sobre el acceso que tiene esta a los datos durante el entrenamiento, es decir, cuantas veces permitimos que el algoritmo se entrene con todo el conjunto de datos, esto se logra a partir de este hiperparámetro, el cual va de la mano de alguna métrica de evaluación o función de pérdida para obtener el número de épocas que las optimiza, debido a que un gran número de épocas sobre ajusta el modelo disminuyendo el error en las predicciones del conjunto de entrenamiento mientras que el error con datos nuevos durante la validación ira aumentando como se observa en la figura 2-11, por lo tanto, un numero óptimo de épocas de entrenamiento es aquel que minimice la función de pérdida de la red [47].

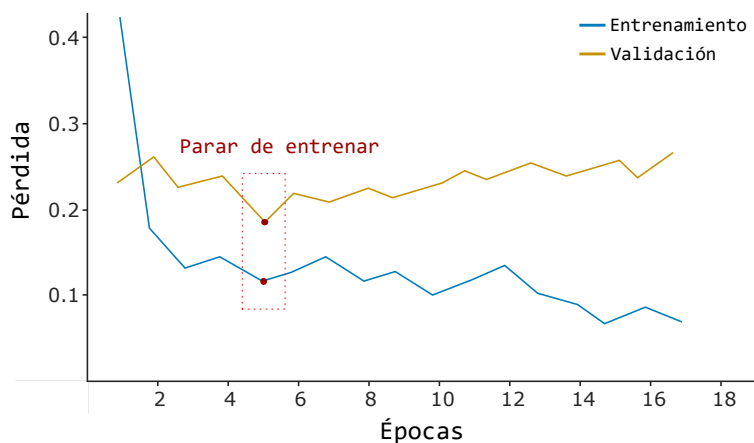


Figura 2-11.: Relación entre el número de épocas de entrenamiento y las funciones de pérdida de los conjunto de entrenamiento y validación.

Tamaño del Lote (Batch):

En la vida real existen problemas muy complejos que abarcan cantidades grandes de datos, por lo que entrenar una red neuronal con todos ellos puede resultar algo ineficiente y computacionalmente pesado, es por esto, que se divide el conjunto total de datos en muestras más pequeñas conocidas como lotes, esto permite que en cada entrenamiento, la red se entrene

con una muestra representativa del total del conjunto de datos. Este hiperparámetro se relaciona con el número de épocas de entrenamientos, pues el tamaño del lote define el número de instancias utilizadas por cada iteración de cada una de las época de entrenamiento [47].

2.4. Redes neuronales recurrentes - RNN

Las redes neuronales recurrentes (RNN) son una arquitectura de red neuronal ampliamente usada en datos que son dependientes entre si como las series de tiempo, al igual que las redes neuronales típicas se apoyan en datos de entrenamiento para aprender y entrenarse, con la diferencia que ahora la red neuronal tendrá memoria, es decir, dependerá de entradas pasadas dentro de la secuencia para calcular salidas nuevas lo que se le conoce con el concepto de recurrencia, como se observa en la figura 2-12 para cada paso de tiempo (t) la red tendrá una entrada x_t y una salida y_t correspondiente, ahora con la existencia de un estado oculto h_t como entrada adicional para la secuencia del siguiente paso de tiempo ($t+1$), de esta manera se vinculan las salidas actuales con valores pasados, por lo que ahora dos entradas iguales pueden generar dos salidas diferentes que dependerán de las entradas anteriores de la misma [48].

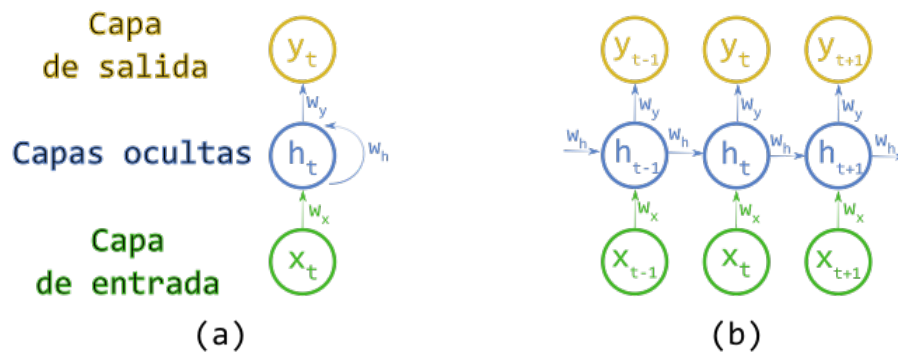


Figura 2-12.: (a) Estructura resumida de un RNN (b) Estructura detallada de un RNN.

Otra diferencia existente entre las RNN y las redes neuronales tradicionales corresponde a los pesos existentes entre cada una de las capas de la red, pues para este tipo de redes se comparte o estandariza el mismo peso entre los nodos de dos capas conectadas, aun así, estos valores se siguen ajustando mediante los procesos de retro propagación y descenso de gradiente, con algunas ligeras variaciones para poder trabajar con los pasos en el tiempo, a esta variación se le conoce como algoritmo de retro propagación en el tiempo (BPTT) y a diferencia del algoritmo tradicional los errores en cada paso de tiempo son tenidos en cuenta sumándose de manera progresiva para obtener un error total que será tenido en cuenta al momento de recalculer los pesos [49]. A medida que se siguen avanzando y profundizando en estructuras

más complejas o especializadas de redes neuronales, surgen a la par nuevos problemas con su funcionamiento, a partir del algoritmo BPTT surgen dos problemas nuevamente relacionados con el gradiente “Gradientes de explosión” y “Gradientes de fuga” [50], que como se vio anteriormente se deben al tamaño o valor del mismo, donde se tiene que:

- **Gradiente pequeño (Gradiente de fuga):** Cuando el gradiente se encuentra en valores mínimos, tiende a hacerse aún más pequeño, lo que hará que los pesos de la red neuronal en cada una de las actualizaciones se vuelvan cada vez más insignificantes (Aproximadamente cero), lo que causa que el algoritmo tenga dificultades al momento de trabajar con secuencias largas de datos (Deja de aprender).
- **Gradiente grande (Gradiente de explosión):** En el caso contrario cuando el gradiente tiende a crecer exponencialmente, los pesos de manera similar tras cada actualización tenderán a crecer de manera acelerada, lo cual desencadenara que la red neuronal sea inestable.

Adicionalmente, surge un problema con la capacidad de retención de información de lo que se entiende por memoria en una RNN, debido a que la información pasada a la que tiene acceso una neurona es la generada por las neuronas inmediatamente anteriores a ella, es decir, se tiene una memoria a corto plazo que no es capaz de retener información lejana en el tiempo, por lo que se puede perder información relevante para la predicción. Para solucionar estos problemas con el gradiente y la memoria se diseñó una RNN con interacciones más complejas conocida como LSTM (Long Short Term Memory) [51].

2.4.1. Long Short Term Memory - LSTM

Es una de las variaciones o arquitecturas de redes neuronales recurrentes más empleadas, que soluciona los problemas anteriormente mencionados, siendo capaz de aprender y olvidar dependencias en el largo plazo, es decir, poder almacenar la información importante y olvidarse de la información irrelevante, mediante la inclusión de un nuevo bloque entre las capas ocultas, el cual consta con funciones de activación conocidas como “Compuertas” como se observa en la figura 2-13, las cuales cumplen determinadas funciones para controlar el flujo de información requerido para predecir la salida, introduciendo un nuevo estado llamado “Estado de celda” que será el encargado de describir la información que será retenida para ser usada por las capas ocultas [52].

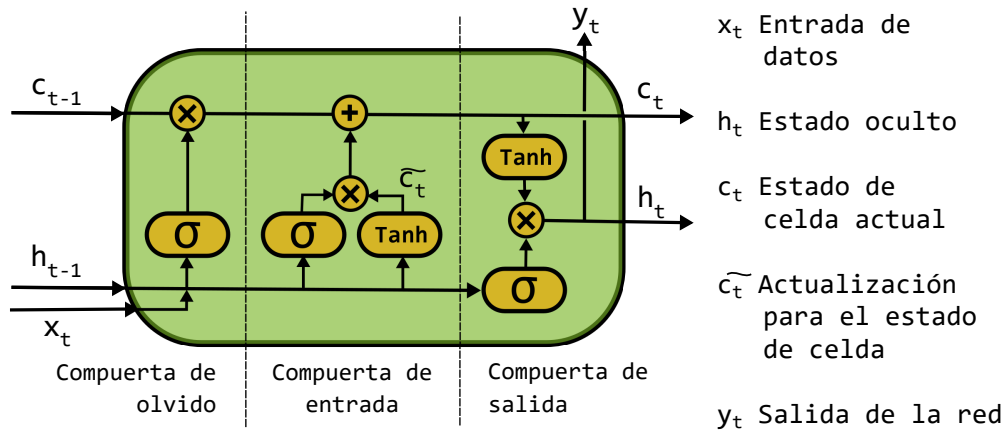


Figura 2-13.: Arquitectura interna red LSTM.

- Compuerta de olvido:** La función principal de esta compuerta es decidir sobre la información relevante que ingresa al bloque, es decir que información se mantiene o se descarta dado el estado oculto anterior h_{t-1} y los nuevos datos de entrada x_t , estos dos valores se pasan por una función de activación sigmoide, por lo tanto se podrá dar una salida cercana a cero, lo que significara que la información se considerara irrelevante y será olvidada, mientras que al obtener valores cercanos a 1, la información será tomada como relevante para aportar a la salida, por lo que se mantendrá para ser usada por la red. Posteriormente este valor se multiplica por el estado de la celda anterior C_{t-1} lo cual funciona como un filtro, pues los componentes del estado que han sido considerados irrelevantes por la red de la compuerta de olvido serán multiplicados por cero o un numero cercano a este y, por tanto, tendrán serán anulados o tendrán menos influencia en los siguientes pasos [53].

$$C_o = \sigma(W_x \cdot x_t + W_h \cdot h_{t-1}) \cdot c_{t-1}$$

- Compuerta de entrada:** Su objetivo es determinar la información que deba retenerse (Añadir) en la memoria de largo plazo para ser usada en la red neuronal, de manera similar a la compuerta de olvido se emplea una función de activación sigmoide para filtrar entre información relevante y no relevante a partir de la combinación del estado oculto anterior h_{t-1} y los nuevos datos de entrada x_t , los cuales también se pasan por una función de activación Tanh (Al tener rango $[-1,1]$ brinda la oportunidad de reducir el impacto de un componente no deseado) para generar el nuevo vector que actualizara la memoria del bloque LSTM [53].

$$c_t = \sigma(W_x \cdot x_t + W_h \cdot h_{t-1}) \cdot \text{Tanh}(W_x \cdot x_t + W_h \cdot h_{t-1}) + C_o$$

- Compuerta de Salida:** Por último, tenemos la compuerta que calcula nuestro estado oculto actual a partir de una función sigmoide, la cual nos permite controlar que

información se entrega por la compuerta de salida, empleando nuevamente el estado oculto anterior h_{t-1} y los nuevos datos de entrada x_t multiplicados por el estado actual de la celda c_t activado con una función Tanh.

$$h_t = \sigma(W_x \cdot x_t + W_h \cdot h_{t-1}) \cdot \text{Tanh}(c_t)$$

A partir de estas compuertas se resuelve el problema de los gradiente de desaparición y explosión, pues en una red LSTM el término del gradiente no posee un patrón fijo ya que puede tomar cualquier valor positivo en cualquier paso de tiempo, lo que permite que, en una cantidad infinita de pasos en el futuro, el termino no tienda a cero o diverja a infinito, pues al empezar a tender a cero, los pesos de las compuertas se ajustan para cambiar esa tendencia hacia uno. En el mundo real nada es perfecto, si bien las redes LSTM solucionan algunos inconvenientes que tienen otras arquitecturas RNN, traen consigo también algunas desventajas empezando por el aumento en el grado de complejidad tanto de análisis como de construcción de la celda lo que requiere más recursos [54].

2.4.2. Gated recurrent units - GRU

La arquitectura LSTM no es la única red creada para resolver el problema de la memoria a corto plazo de las configuraciones RNN, existe una arquitectura con ciertas variaciones o “evoluciones” conocida como GRU, la cual ya no emplea un segundo estado (“Estado de celda”) para regular la información que se almacena o se olvida, para esto se vale únicamente de los estados ocultos, como se observa en la figura 2-14, donde también se evidencia la simplificación de las compuertas empleadas, pues se disminuye en uno la cantidad de compuertas teniendo ahora solo la compuerta de reinicio y la compuerta de actualización, cumpliendo la misma función de controlar la información que se debe retener, disminuyendo a la par la complejidad interna de la celda [55].

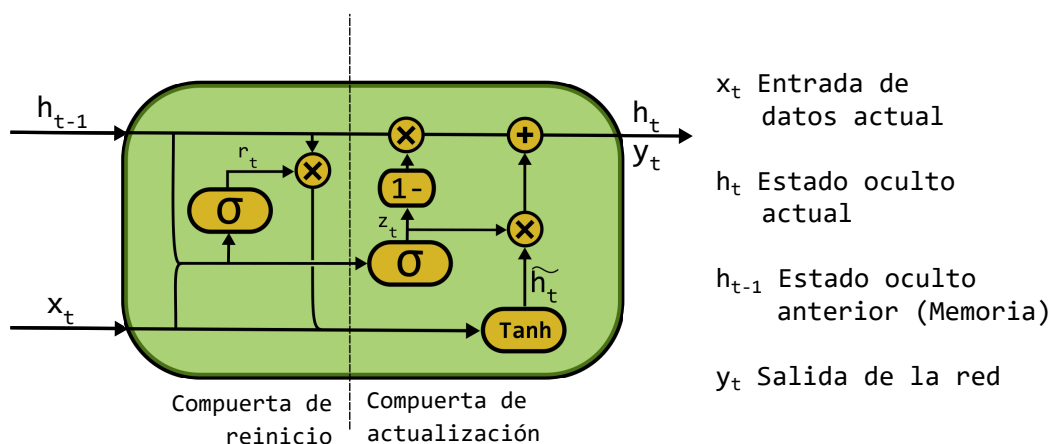


Figura 2-14.: Arquitectura interna red GRU.

- **Compuerta de actualización:** En similitud a una red LSTM esta compuerta funciona como una unión entre algunas características de la compuerta de olvido y de entrada, pues decide qué información no debe ser tenida en cuenta y cual es relevante para ser añadida a la memoria de la celda, a partir de la entrada actual \mathbf{x}_t y la información pasada \mathbf{h}_{t-1} con sus respectivos pesos a través de una función sigmoide:

$$z_t = \sigma(W_x \cdot x_t + W_h \cdot h_{t-1})$$

- **Compuerta de reinicio:** Esta compuerta a diferencia de la anterior es la encargada de decidir cuanta información almacenada hay que olvidar o descartar, calculándolo de la misma manera que la compuerta de actualización a partir de una función sigmoide, la diferencia radica en el uso que se le da a este valor calculado:

$$r_t = \sigma(W_x \cdot x_t + W_h \cdot h_{t-1})$$

A partir de este término se introduce la información que actualizará la memoria y con la cual la compuerta de reinicio almacenará la información relevante o determinará que eliminar de los pasos anteriores de tiempo:

$$\tilde{h}_t = \mathbf{tanh}(W_x \cdot x_t + r_t \cdot W_h \cdot h_{t-1})$$

Por último, la red calcula la información que estará contenida actualmente en la celda GRU, determinando que recopilar de la memoria actual $\tilde{\mathbf{h}}_t$ y de los pasos anteriores \mathbf{h}_{t-1} :

$$h_t = z_t \cdot \tilde{h}_t + (1 - z_t) \cdot h_{t-1}$$

A través de estas dos compuertas de actualización y reinicio las redes GRU son capaces de almacenar y filtrar información, eliminando así el problema de los gradientes, pues esta arquitectura no permite que la entrada nueva sea eliminada, por el contrario, mantiene la información relevante de ella y la mantiene en los siguientes pasos de tiempo. En la actualidad se consideran que las arquitecturas GRU son superiores a las arquitecturas LSTM, debido a la simplificación para abarcar el problema del gradiente sin requerir unidades de memoria, sin embargo, no siempre son mejores que las redes LSTM, las cuales son predilectas al momento de trabajar con series de datos grandes, razón por la cual el establecer cuál de los dos tipos de arquitectura es mejor depende de la aplicación y el adecuado entrenamiento que se realice [56][57].

2.4.3. Arquitectura Encoder-Decoder

En las arquitecturas LSTM y GRU se observó que en cada instante de tiempo t a cada entrada le correspondía una respectiva salida, es decir que eran dos secuencias con la misma longitud, sin embargo, en problemas y aplicaciones de la vida real se desea predecir secuencias de salida con secuencias de entrada de diferente longitud, donde no exista una correspondencia entrada – salida, este tipo de modelos entra dentro de una categoría conocida como secuencia a secuencia (Seq-to-Seq) teniendo como modelo principal la arquitectura codificador-decodificador que internamente se compone de dos modelos como su nombre lo indica, que tienen como objetivo principal generar como salida la secuencia más probable de ocurrencia, donde el codificador se encarga de codificar y recorrer la secuencia completa de entrada en un vector único de longitud fija (Vector de contexto) para ser utilizado por el decodificador que es el encargado de recorrer la secuencia de salida [58].

Codificador

La construcción del módulo de codificación parte de redes recurrentes simples, LSTM o GRU apiladas una tras otra, por lo que puede ser considerado como una extensión de estas arquitecturas y no un modelo aparte y diferente en sí, cada red que se emplee utiliza un único elemento de la secuencia de entrada para recopilar la información y propagarla en el tiempo, lo cual es nuevamente logrado a partir del estado oculto en la red, formado por los estados ocultos anteriores \mathbf{h}_{t-1} y la entrada \mathbf{x}_t , la función realizada por la red escogida \mathbf{f} y los determinados pesos, el cual contendrá la información codificada de cada estado oculto y entrada de pasos de tiempo anteriores:

$$h_t = f(W_h \cdot h_{t-1} + W_x \cdot x_t)$$

La comunicación entre el módulo codificador y el módulo decodificador se realiza a través de un vector llamado “Vector de contexto” el cual no es más que el último estado oculto generado por el codificador, el cual funcionara como el estado oculto inicial para el decodificador, de esta manera se comparte toda la información correspondiente de la entrada para proveer al decodificador de herramientas y datos para realizar predicciones más precisas.

Decodificador

Al igual que el codificador el módulo de codificación, el decodificador también se compone de la unión de varias arquitecturas RNN, las cuales reciben un estado oculto anterior \mathbf{s}_{t-1} y generan la salida \mathbf{y}_t y el siguiente estado oculto \mathbf{s}_t :

$$s_t = f(W_h \cdot h_{t-1})$$

Mientras que la salida para este tipo de arquitectura se calcula utilizando la función `softmax` y el estado oculto \mathbf{s}_t , lo cual dará como resultado la salida \mathbf{y}_t más probable para el sistema:

$$y_t = \text{softmax}(W_s \cdot s_t)$$

Como se había mencionado la ventaja de emplear este tipo de modelos parte del hecho de que pueden modelar problemas a partir de secuencias de diferentes longitudes entre sí, ya que las entradas y salidas pueden ser de diferentes longitudes, es decir, no están correlacionadas. Esta técnica funciona bien para secuencias pequeñas, pero cuando la longitud de la secuencia aumenta, es muy difícil resumir una secuencia larga en un solo vector, y luego el modelo a menudo olvida las partes anteriores de la secuencia de entrada al procesar las últimas partes [59].

2.4.4. Mecanismos de atención

Al igual que ocurrió con la arquitectura GRU como evolución de las redes LSTM, las redes neuronales basadas en mecanismos de atención parten de una evolución de los modelos Encoder-Decoder, para mejorar el desempeño del modelo frente a secuencias largas de datos, lo cual es su principal desventaja, esto se logra a partir de una modificación en el vector de contexto, el cual ya no será único en toda la red, si no que se construye de la misma manera ahora para cada paso de tiempo del decodificador, lo que permite que este acceda de manera selectiva a la información alojada en el codificador [60][61].

Codificador

En los modelos de atención el funcionamiento del codificador es igual a la arquitectura básica Encoder-Decoder, calculando el estado oculto a partir del estado oculto anterior y de la entrada actual con sus respectivos pesos:

$$h_t = f(W_h \cdot h_{t-1} + W_x \cdot x_t)$$

Ahora bien, como se sabe la principal diferencia radica en el vector de contexto que ya no será el último estado oculto del codificador, ahora para cada combinación de pasos de tiempo \mathbf{t}_c del codificador y \mathbf{t}_d del decodificador, se calcula un puntaje $\mathbf{e}(\mathbf{t}_c, \mathbf{t}_d)$ conocido como “Puntaje de alineación” que surge a partir de unos nuevos pesos entrenables llamados pesos de atención:

$$e(t_c, t_d) = V_a \tanh(U_a \cdot s_{t_d-1} + W_a \cdot h_{t_c})$$

Los pesos de atención corresponden a los términos \mathbf{W}_a , \mathbf{U}_a y \mathbf{V}_a :

- \mathbf{W}_a : Pesos asociados a los estados ocultos del codificador.
- \mathbf{U}_a : Pesos asociados a los estados ocultos del decodificador.
- \mathbf{V}_a : Pesos que definen que función calcula el puntaje de alineación.

A partir de este puntaje de alineación se calcula un último peso de atención α , con el cual se puede obtener la importancia de la entrada del codificador en el tiempo \mathbf{t}_c para la salida

del decodificador en el tiempo t_d :

$$\alpha(t_c, t_d) = \frac{\exp(e(t_c, t_d))}{\sum \exp(e(t_c, t_d))}$$

Por último, el vector de contexto se calcula a partir de la suma ponderada de todos los estados ocultos del codificador según el peso de atención correspondiente [62], lo que permitirá que preste más atención a las entradas más relevantes del conjunto de datos:

$$c_{t_d} = \sum \alpha(t_c, t_d) \cdot h_{t_c}$$

Decodificador

El nuevo vector de contexto se pasa al decodificador, para calcular la distribución de probabilidad para la siguiente salida posible, aplicándose a todos los pasos de tiempo en la entrada, donde el vector de estado oculto s_{t_d} se calcula a partir de:

$$s_{t_d} = f(s_{t_d-1}, y_{t_d-1}, c_{t_d})$$

Con lo cual para de los diferentes componentes de la salida se pueden hallar correlaciones respectivas con componentes de la secuencia de entrada, donde nuevamente se calcula la salida a partir de una función **softmax**:

$$y_{t_d} = \text{softmax}(W_s \cdot s_{t_d})$$

3. Metodología

El diseño metodológico llevado a cabo para el desarrollo del proyecto se estructuró primero con la construcción de un marco conceptual para posteriormente llevar a cabo el desarrollo de tres actividades o etapas principales como se observa en la figura 3-1 y de los resultados obtenidos de estas actividades se realizó la escritura de un artículo de investigación sobre las novedades desarrolladas para los campos de estudio pertinentes, este artículo se encuentra en el **Anexo A**. En la construcción del marco conceptual se profundizó en el estudio de redes neuronales para el aprendizaje profundo, los modelos existentes y las técnicas de implementación de estos para su uso en predicción de series de tiempo, posteriormente se llevó a cabo la revisión detallada de trabajos realizados por otros autores, tanto en el campo de los hornos de arco eléctrico como en el campo de la predicción de series de tiempo, se continuó con la revisión de la información brindada en los manuales de funcionamiento de los hornos línea 1 y línea 2 por parte de CMSA, así como los informes de procesos y resultados de las fases anteriores del proyecto.

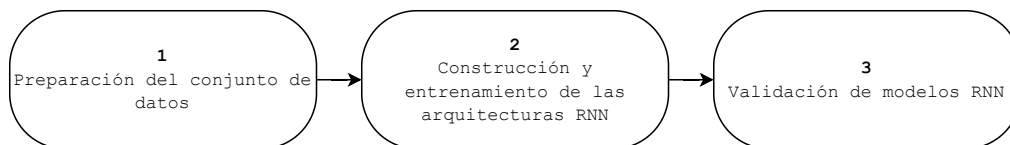


Figura 3-1.: Etapas generales de desarrollo y construcción de modelos de aprendizaje profundo para la predicción de temperatura en un horno de arco eléctrico.

En la primera etapa, se obtuvo un conjunto de datos actualizados correspondiente a las variables de funcionamiento del horno ya establecidas dentro de la fase actual proyecto, con esta información se continuó a realizar el análisis de datos exploratorio y la preparación del conjunto de datos para ser utilizados por los modelos de aprendizaje profundo, para lo cual se utilizó Colab el servicio en la nube de Google basado en ambientes de ejecución de Jupyter para programar con Python y sus librerías especiales de estadística y matemática para el manejo de datos, una vez realizado este procedimiento, se ajustó el conjunto de datos para poder trabajar con modelos de aprendizaje profundo (Normalización y división en conjuntos de entrenamiento/validación). Continuando con la siguiente etapa se llevó a cabo la construcción y entrenamiento de modelos para la predicción de temperaturas de pared media en los hornos línea 1 y línea 2, estableciendo como función de costos la función

RMSE para la evaluación y comparación de los mismo. Por último en esta etapa se obtienen las gráfica del comportamiento predicho vs el comportamiento real de la temperatura en los hornos, lo anteriormente descrito se desarrolló en una instancia de Colab con el apoyo de la librería específica para aprendizaje profundo de Python llamada TensorFlow. En la última etapa se llevó a cabo la validación de las predicciones realizadas en ambos hornos, donde se comparó y analizó el rendimiento de los modelos entrenados ante diversos casos de estudio. Los capítulos siguientes muestran los resultados en cada una de las etapas mostradas en este metodología los cuales se alinean con los objetivos planteados al inicio de este proyecto.

4. Preparación del conjunto de datos

El flujo de trabajo empleado para el análisis, la limpieza y la preparación del conjunto de datos que se utilizó en el entrenamiento y prueba de los modelos de aprendizaje profundo se compone de 6 pasos detallados en la figura 4-1 que serán detallados en las siguientes subsecciones.

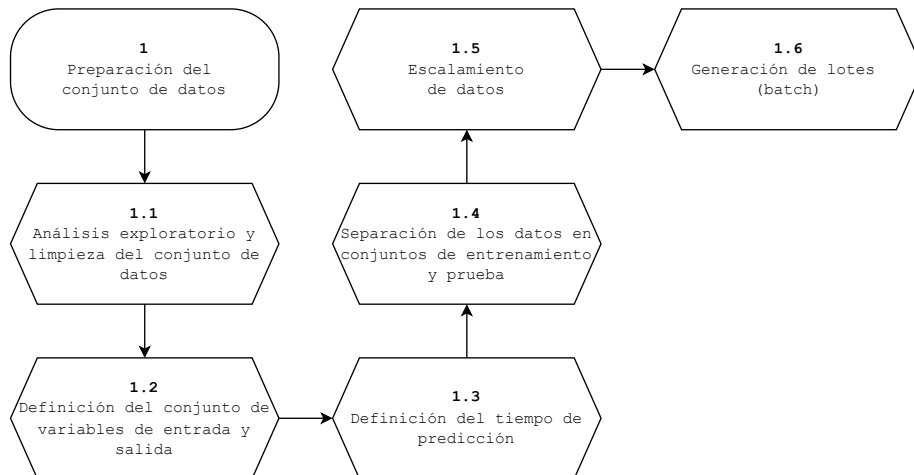


Figura 4-1.: Etapas específicas para la preparación del conjunto de datos que alimentara los modelos de aprendizaje profundo.

4.1. Análisis exploratorio y limpieza de datos

El primer paso dentro del proceso metodológico para la construcción de un conjunto de datos adecuados para el desarrollo y entrenamiento de redes neuronales, empieza con el entendimiento y preprocesamiento del mismo, para esto se sigue el paso a paso planteado por los autores en [13], aplicado a cada uno de los conjuntos de datos de los hornos línea 1 y línea 2, los cuales cuentan con un sistema de que registra, monitorea y almacena las variables de proceso involucradas en los hornos de manera periódica en un intervalo de 15 minutos.

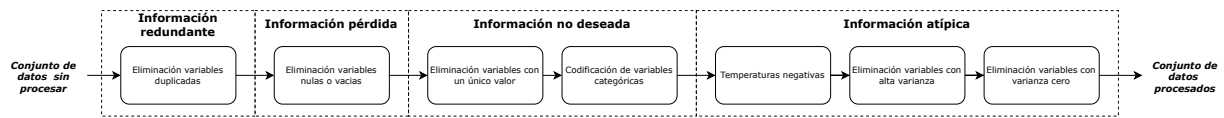


Figura 4-2.: Flujo de trabajo para la limpieza de datos en los hornos línea 1 y línea 2.

El flujo de trabajo desarrollado para la limpieza de datos establece 4 filtros, cada uno con sus respectivas etapas como se observa en la figura 4-2, los cuales tienen por objetivo asegurar la calidad del conjunto de datos final.

1. **Variables duplicadas:** Variables que registro a registro tienen el mismo valor en cada uno de ellos, por lo que son variables idénticas y basta con conservar solo una de ellas.
2. **Variables con registros vacíos o nulos:** Es normal que algunas variables dentro de un conjunto de datos presenten registros con valores nulos o faltantes, sin embargo, existen variables donde el porcentaje de estos valores respecto al número total de registros empieza a ser una cantidad considerable.
3. **Variables únicas:** Variables que a lo largo de sus registros almacenan un único valor, por lo tanto, no aportan información alguna al proceso.
4. **Variables categóricas:** Variables no numéricas que pueden registrar modos de operación o información relevante del proceso, para su utilización se realiza una codificación numérica.
5. **Variables de temperatura negativas:** Al tratarse de un horno industrial, el registrar temperaturas en funcionamiento por valores debajo de cero grados está por fuera del rango de operación, esto se puede deber a fallas en los protocolos de comunicación o en sistemas de adquisición de datos, por lo que no reflejan con veracidad el comportamiento del sistema.
6. **Variables con alta varianza:** Variables con cambios abruptos y valores extremos por fuera del rango de operación (Normal-Emergencia). Se define como criterio de eliminación variables con datos por arriba de un 10 % de tolerancia del valor máximo expresado en los manuales de operación.
7. **Variables con varianza cero:** Variables que se mantienen constantes en el tiempo por un largo periodo de tiempo (Varianza cero), el criterio empleado corresponde a la eliminación de variables que posean más de 50 % de sus datos con varianza cero.

Las variables que componen los conjuntos de datos entregados sin ningún tipo de procesamiento corresponden a variables de entrada, proceso y salida involucradas en la operación

de los hornos línea 1 y línea 2. Entre estas se encuentran mediciones que se tienen antes de la entrada del horno, entre ellas se encuentra la química de la calcina, variable de interés que difiere de las demás debido a que sus valores se obtienen tras el proceso de extracción; variables de proceso, que miden parámetros de configuración y control del horno y por último variables de salida provenientes de los productos y subproductos de las coladas y sangrías, también pertenecen a este grupo las variables asociadas al monitoreo de control estructural del horno, los flujos de calor y temperaturas en los diferentes componentes:

- Voltaje, corriente e impedancia del electrodo
- Posición relativa del electrodo
- Tap del transformador
- Potencia del horno eléctrico
- Alimentación en tubos centrales, semicentrales y laterales
- Temperaturas de termocuplas a predecir de plate coolers.

4.1.1. Conjunto de datos horno línea 1

El conjunto de datos correspondiente al horno línea 1 presenta las siguientes características generales:

- **Fecha primer registro:** 30 de septiembre de 2014.
- **Fecha ultimo registro:** 30 de septiembre de 2019.
- **Cantidad de variables (TAG's):** 1180 Variables.
- **Cantidad de registros:** 175,297 registros periódicos.

El flujo de trabajo descrito anteriormente para la limpieza y análisis de datos aplicados al horno línea 1 se lista en la tabla 4-1, a partir del cual se listan también la cantidad de variables eliminadas por dicho procedimiento.

Procedimiento	TAGs eliminados
Eliminación variables duplicadas	76
Eliminación variables nulas o vacías	2
Eliminación variables con un único valor	60
Codificación de variables categóricas	5
Eliminación temperaturas negativas	74
Eliminación variables con alta varianza	97
Eliminación variables con varianza cero	22
Variables restantes:	844

Tabla 4-1.: Limpieza del conjunto de datos para el horno línea 1.

4.1.2. Conjunto de datos horno línea 2

El conjunto de datos correspondiente al horno línea 2 presenta las siguientes características generales:

- **Fecha primer registro:** 01 de marzo de 2021.
- **Fecha ultimo registro:** 15 de octubre de 2021.
- **Cantidad de variables (TAG's):** 1112 Variables.
- **Cantidad de registros:** 21,984 registros periódicos.

Como se observa las fechas de los datos utilizados para este horno, difieren de los usados para el horno línea 1 dado que este horno fue reparado y entro en funcionamiento al principio del año 2022, con lo cual se quería evaluar si igualmente para un periodo más corto de datos se podían lograr resultados similares en la predicción de variables. Al igual que en el conjunto de datos del horno línea 1, en **4-2**, se lista el flujo de trabajo descrito anteriormente para la limpieza y análisis.

Procedimiento	TAGs eliminados
Eliminación variables duplicadas	187
Eliminación variables nulas o vacías	1
Eliminación variables con un único valor	63
Codificación de variables categóricas	1
Eliminación temperaturas negativas	74
Eliminación variables con alta varianza y varianza cero	115
Variables restantes:	671

Tabla 4-2.: Limpieza del conjunto de datos para el horno línea 2.

4.2. Variables de entrada y salida

Una vez realizado el proceso de limpieza de datos se realizó la selección de variables de acuerdo con los análisis realizados y resultados obtenidos en los informes del proyecto “Informe preliminar con análisis estadístico de datos y correlaciones posibles” [63], “Informe técnico de caracterización e identificación de variables del horno línea 1 FC01” [64] e “Informe técnico de caracterización e identificación de variables del horno línea 2 FC150” [65]. En estos informes a partir de un estudio de correlaciones entre las variables resultantes de los conjuntos de datos del horno línea 1 y horno línea 2 posteriores a la limpieza de datos, se determinó el coeficiente de correlación de Pearson el cual varía entre valores de -1 a 1, siendo una correlación negativa muestra de asociaciones negativas entre variables, es decir, el aumento de una variable se vera reflejado como una disminución en otra variable e inversamente. En caso contrario al existir una correlación mayor que 0 entre dos variables se obtiene una asociación positiva, a medida que aumenta el valor de una variable, también lo hace el valor de la otra. Un valor de correlación centrado en 0 indica una disociación entre las dos variables, el comportamiento de una variable no se vera afectado por el comportamiento de la otra. De aquí se seleccionaron las variables de proceso que estaban más correlacionadas mas positivamente entre sí, para después ser filtradas nuevamente por información brindada por CMSA debido a la importancia de cada una en el proceso de fundición de ferroníquel en los hornos de arco eléctrico.

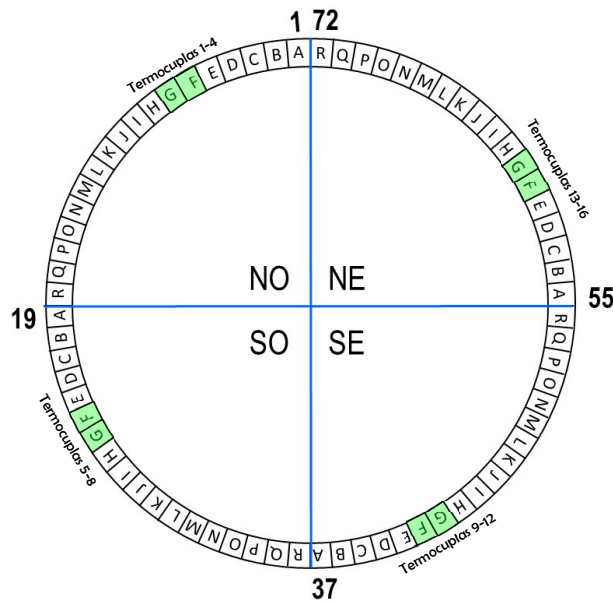


Figura 4-3.: Paneles seleccionados en el horno para la predicción de temperatura en sus termocuplas.

La selección inicial de las 16 termocuplas para la predicción de temperatura se realizó

de forma radialmente distribuida a lo largo del horno como se observa en la figura 4-3, donde por cada una de las cuatro secciones del horno en sus respectivos paneles G y F se seleccionaron 4 termocuplas para ser empleadas como variables de entrada y salida en los modelos de predicción de temperatura.

4.2.1. Horno línea 1

El conjunto de datos resultante para el horno línea 1 consta de 49 variables entrada y 16 variables de salida correspondientes a las termocuplas seleccionadas como se observa en la tabla 4-3.

Proceso	Cantidad de variables
Potencia horno	1
Voltaje electrodos	3
Corriente electrodos	3
Arco de electrodos	3
Posición electrodos	3
Potencia electrodos	3
Modo de control	1
Totalizador alimentación	1
Química calcina	15
Temperatura	16
Variables Totales:	49
Cantidad de variables a predecir	
Termocuplas (Temperatura)	16

Tabla 4-3.: Cantidad de variables seleccionadas por proceso para la entrada y salida del modelo de predicción de temperaturas en el horno línea 1.

4.2.2. Horno línea 2

El conjunto de datos resultante para el horno línea 2 consta de 76 variables entrada y 16 variables de salida correspondientes a las termocuplas seleccionadas como se observa en la tabla 4-4.

Proceso	Cantidad de variables
Potencia	1
Voltaje electrodo	3
Corriente electrodo	3
Impedancia de electrodos	6
Posición electrodos	3
Tap del transformador	3
Alimentación tubos centrales	3
Alimentación tubos semicentrales	9
Alimentación tubos laterales	12
Química calcina	17
Temperatura	16
Variables Totales:	76
Cantidad de variables a predecir	
Termocuplas (Temperatura)	16

Tabla 4-4.: Cantidad de variables seleccionadas por proceso para la entrada y salida del modelo de predicción de temperaturas en el horno línea 2.

4.3. Tiempo de predicción

En la predicción de series de tiempo, un parámetro importante a definir es la ventana de tiempo en el futuro para la cual se quiere predecir, pues a partir de los modelos de aprendizaje profundo podemos predecir los valores futuros de las 16 termocuplas desde el siguiente paso de tiempo correspondiente a la próxima lectura de 15 minutos en el futuro, como para una cantidad determinada de pasos en el futuro dependiendo del interés de la predicción dada por la siguiente relación:

$$\text{Tiempo de predicción [Horas]} = \frac{\text{Pasos de tiempo}}{4}$$

Entre mayor sea el tiempo de predicción menor va a ser la precisión de las predicciones, dado que al predecir ventanas de tiempo más distantes en el futuro los modelos de aprendizaje profundo requieren recordar información del pasado que cada vez se encuentra más alejada, lo cual se busca solucionar con las redes neuronales recurrentes y su concepto de memoria. En el campo de la salud estructural y el funcionamiento de un horno a nivel industrial, el tiempo de predicción entra a ser un factor aún más importante, pues un tiempo de predicción corto le da a los operarios en planta un tiempo de análisis y reacción bastante limitado, mientras que un tiempo de predicción largo conlleva a predicciones cada vez menos fiables. De esta manera se equilibró entre los costos de tener una buena precisión y una utilidad funcional de

las predicciones estableciendo una ventana de tiempo de predicción de tres horas o 12 pasos de tiempo en el futuro.

4.4. Conjunto de entrenamiento y prueba

En la construcción y evaluación de modelos de aprendizaje automático es importante estudiar el comportamiento y la precisión de las predicciones realizadas para datos nuevos que nunca han sido observados por los modelos, para esto se realiza una partición en dos subconjuntos (Entrenamiento y Prueba) del conjunto de datos construido para trabajar y de esta manera no volver a la tarea de obtener y preprocesar un conjunto de datos nuevos. El primero de estos subconjuntos se denomina conjunto de entrenamiento, el cual se utiliza como su nombre lo indica para entrenar y ajustar los parámetros del modelo, mientras que el segundo subconjunto conocido como conjunto de prueba, se emplea para realizar las predicciones con datos nunca vistos por el modelo y de esta manera poder comparar con los valores esperados. Esta es la forma de funcionamiento del modelo en la práctica, entrenando con los datos disponibles con entradas y salidas conocidas, para luego hacer predicciones sobre nuevos datos en donde no se conocían las salidas esperadas.

En series de tiempo, la partición del conjunto total de datos en estos dos subconjuntos se realiza con especial cuidado, pues el conjunto de datos de prueba debe crearse de manera obligatoria con una cantidad definida de los últimos datos registrados, pues de hacerse al revés se estaría prediciendo el pasado con datos futuros lo cual no tiene sentido. Se realizó la partición para los conjuntos de datos línea 1 y línea 2 en una proporción 90-10, es decir, el conjunto de datos de entrenamiento correspondió al 90 % de los datos disponibles mientras que el conjunto de datos de prueba fue el 10 % restante.

4.4.1. Horno línea 1

En el horno línea 1 debido a problemas con la convergencia de los modelos de aprendizaje profundo relacionados con la cantidad de variables y el número de registros del conjunto de datos, se limitó el mismo a los últimos 40.000 registros, lo que corresponde a aproximadamente 1.14 años de información entre el 9 de agosto de 2018 y el 30 de septiembre de 2019.

	Porcentaje [%]	Cantidad de datos
Conjunto sin partición	100 %	40000
Conjunto de entrenamiento	90 %	36000
Conjunto de prueba	10 %	4000

Tabla 4-5.: Cantidad de datos por partición del conjunto de datos para prueba y entrenamiento en el horno línea 1.

4.4.2. Horno línea 2

En el caso del horno línea 2 como la cantidad de registros con los que se cuenta no supera la cantidad de los 40.000 registros, por lo que no se requirió de ninguna limitación.

	Porcentaje [%]	Cantidad de datos
Conjunto sin partición	100 %	21984
Conjunto de entrenamiento	90 %	19785
Conjunto de prueba	10 %	2199

Tabla 4-6.: Cantidad de datos por partición del conjunto de datos para prueba y entrenamiento en el horno línea 2.

4.5. Escalamiento de datos

El conjunto de datos como ya se ha observado se compone de distintas variables, las cuales vienen medidas en diferentes unidades y escalas, por ejemplo, no es lo mismo medir y comparar la corriente de un electrodo en unidades de Amperios contra variables químicas de la calcina dadas en cantidades o porcentajes. Si bien esto facilita la comprensión para los operarios humanos, para los modelos de aprendizaje profundo causa confusión, pues estos no saben de unidades. De esta manera, se escalan todos los datos en una misma escala correspondida entre 0 y 1 (0 para el valor mínimo de la variable y 1 para su valor máximo como se observa en la figura 4-4) para ser usados por la red neuronal y evitar confusiones de la misma, esta tarea se realiza por separado para el conjunto de entrenamiento y el conjunto de prueba, pues si bien ambos conjuntos provienen del mismo conjunto de datos, en la práctica el conjunto de datos de prueba corresponde a información nueva que puede tener nuevos rangos de valores máximos y mínimos para el escalamiento.

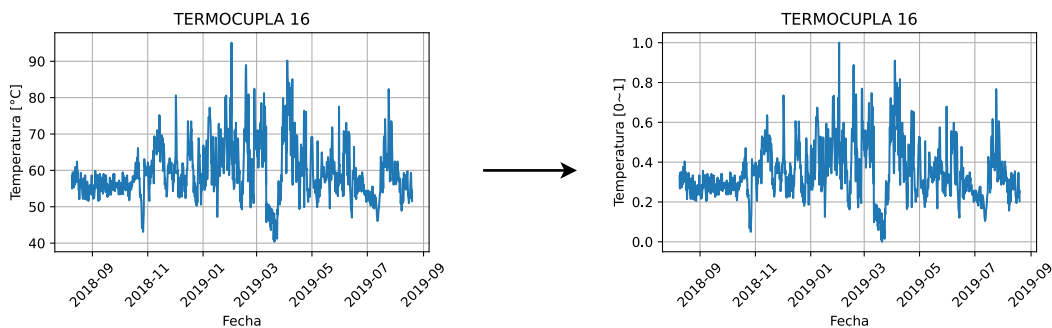


Figura 4-4.: Escalamiento aplicado a los datos de temperatura en la termocupla 16.

4.6. Generación de lotes

El conjunto de datos preparado para el entrenamiento del modelo del horno línea 1 consiste en 49 señales de entrada y 16 señales de salida con 36.000 registros cada una, es decir, el conjunto de datos contiene 2.340.000 datos individuales, mientras que el conjunto de entrenamiento para el horno línea 2 contiene 76 señales de entrada y 16 señales de salida con 21.984 registros, aproximadamente 2.022.000 datos individuales. En lugar de entrenar las redes neuronales con estas secuencias completas de datos se crean lotes (batches) de secuencias más cortas seleccionadas al azar de los datos de entrenamiento, esto debido a sus notables ventajas con el rendimiento y velocidad de entrenamiento. Para el entrenamiento de ambos hornos se generaron 250 lotes cada uno con una longitud de 1152 pasos de tiempo o 12 días.

5. Desarrollo, entrenamiento y comparación de arquitecturas RNN

El flujo de trabajo empleado para llevar a cabo la construcción y entrenamiento de los modelos de aprendizaje profundo se estructuró en 6 pasos detallados en la figura 5-1, en donde a diferencia del flujo lineal realizado para la preparación del conjunto de datos, se trabajó con un flujo de retroalimentación a partir de la selección y optimización de los hiperparámetros de los modelos, debido a esto en la sección 5.1 para una mejor lectura y entendimiento del desarrollo realizado se trabajó de manera conjunta del paso 2.2 al paso 2.6.

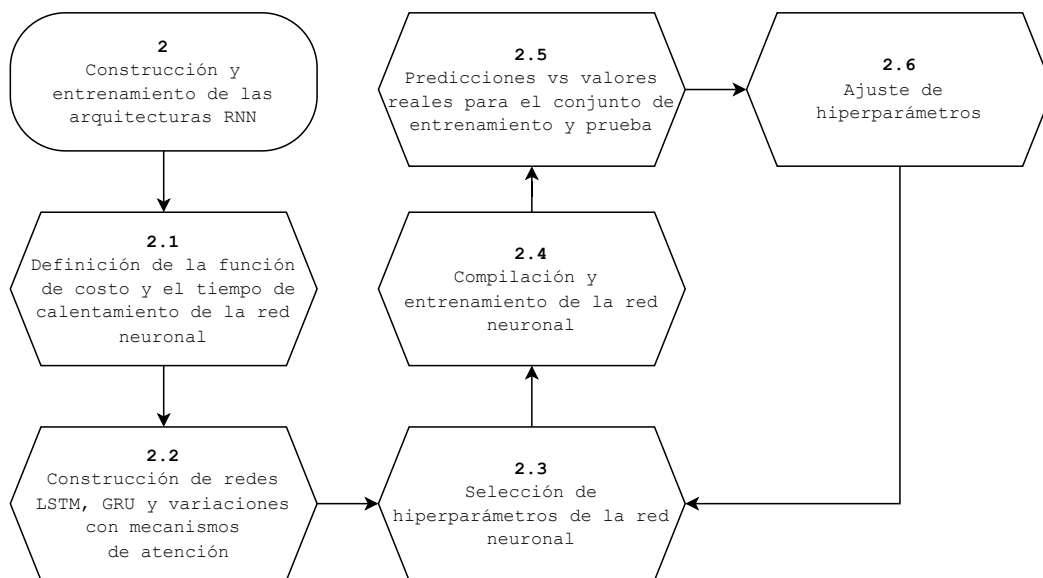


Figura 5-1.: Etapas específicas de desarrollo y entrenamiento de modelos de aprendizaje profundo para predicción de series de tiempo.

5.1. Función de costo y tiempo de calentamiento

Las funciones de costo como se había establecido tienen gran importancia al momento de evaluar las pérdidas o la precisión de las predicciones de un modelo de aprendizaje profundo.

Entre ellas se encuentra la raíz del error cuadrático medio (RMSE), en otros términos, es la raíz cuadrada de la varianza de los residuos entre las predicciones realizadas y el valor real. El RMSE indica el ajuste absoluto del modelo a los datos (Qué tan cercanas están las predicciones de los valores predichos por el modelo), como la raíz cuadrada de una varianza, esta función se puede interpretar como la desviación estándar de la varianza no explicada y tiene la propiedad de estar en las mismas unidades que las variables de salida, donde obtener un valor bajo de RMSE indica un mejor ajuste, siendo esta una buena medida de la precisión con la que el modelo predice la respuesta y es el criterio de ajuste más importante debido a que el objetivo principal del modelo es la predicción. El RMSE para el conjunto de entrenamiento y el conjunto de prueba debe ser muy similar si se ha construido un buen modelo. Si el RMSE para el conjunto de prueba es mucho más alto que el del conjunto de entrenamiento, es probable que el modelo se esté sobre ajustando a los datos.

En la predicción de series de tiempo mediante modelos de redes neuronales recurrentes ocurre que la predicción no es muy precisa para los primeros 30-50 pasos de tiempo porque el modelo ha visto muy pocos datos de entrada en este momento. El modelo genera un solo paso de tiempo de datos de salida para cada paso de tiempo de los datos de entrada, por lo que cuando el modelo solo se ha ejecutado durante unos pocos pasos de tiempo, sabe muy poco del historial de las señales de entrada y no se puede hacer una predicción precisa. El modelo necesita “calentarse” procesando aproximadamente 50 pasos de tiempo antes de que se puedan usar sus señales de salida predichas. Razón por la cual se ignora este “período de calentamiento” de 50 pasos de tiempo al calcular el error cuadrático medio en la función de costo:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=50}^N (y_i - \hat{y}_i)^2}$$

El “período de calentamiento” se muestra con un cuadro gris como se observa en la figura **5-2**, para lo cual se tomó como muestra las termocuplas 1 y 16, en los primeros pasos de tiempo la predicción de temperatura en las termocuplas está totalmente alejada de los valores reales teniendo un pico máximo para la primera predicción, al pasar los 50 pasos de tiempo se observa como los valores predichos se acercan más a los valores reales.

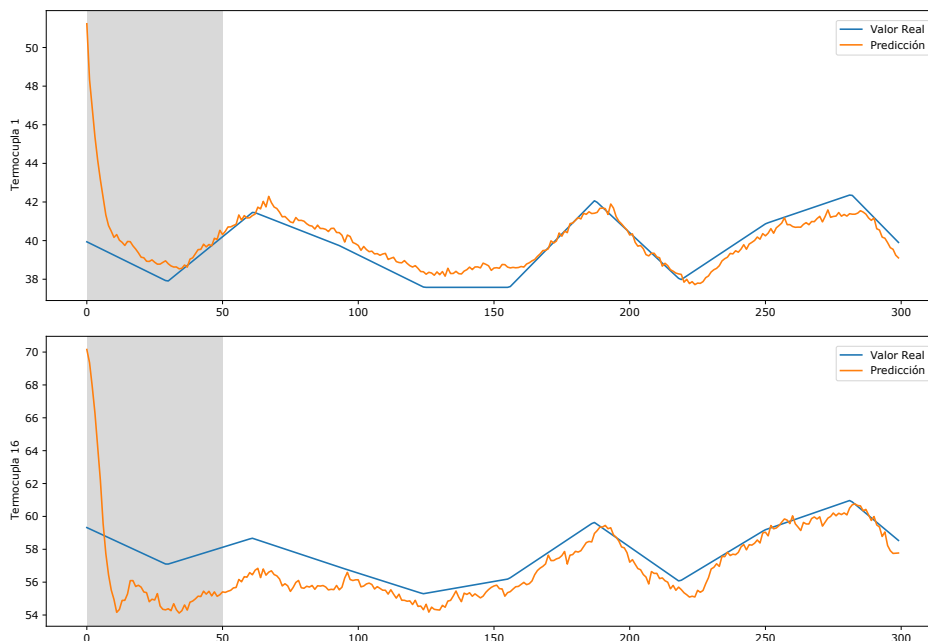


Figura 5-2.: Tiempo de calentamiento RMSE.

5.2. Arquitecturas RNN, entrenamiento, hiperparámetros y ajuste de hiperparámetros

Las principales ventajas y diferencias descritas en el documento sobre las arquitecturas recurrentes, no son las únicas condiciones para seleccionar entre un modelo con buenos resultados sobre otro también con buenos resultados, pues entre un conjunto de datos y otro para un mismo caso de estudio, pueden existir dos comportamientos totalmente diferentes, razón por la cual se diseñaron diversas arquitecturas RNN tanto para el horno línea 1 como para el horno línea 2, variando el número de épocas de entrenamiento entre 1 época y 20 épocas y la cantidad de unidades para cada arquitectura RNN, de esta manera se tiene:

1. Modelo LSTM (32-64-96 Celdas)
2. Modelo GRU (100-200-300 Unidades)
3. Modelo LSTM + Atención (Mejor Resultado del modelo LSTM)
4. Modelo GRU + Atención (Mejor Resultado del modelo GRU)

Las señales de salida en el conjunto de datos se han limitado a estar entre 0 y 1 debido a la función de escalamiento empleada, con lo cual se limitó la salida de la red neuronal usando la función de activación Sigmoide en una capa densa, que encierra la salida para que esté entre 0 y 1.

5.2.1. Modelos Horno línea 1

Modelos LSTM

Para el horno línea 1 se desarrollaron 3 variaciones de modelos LSTM aumentando cada vez más el número de celdas LSTM empleadas para el entrenamiento y predicción del modelo, se empezó con una arquitectura de 32 celdas LSTM (**Tabla 5-1**), siguiendo con un aumento del doble de celdas para 64 celdas LSTM (**Tabla 5-2**) y finalizando con 96 celdas LSTM (**Tabla 5-3**).

Arquitectura con 32 celdas LSTM				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0187	0.014	5.44	5.88
2	0.0076	0.0063	4.04	4.15
3	0.0041	0.0041	2.88	3.19
4	0.003	0.0033	2.72	2.93
5	0.0025	0.0029	2.46	2.75
6	0.0023	0.0026	2.4	2.72
7	0.002	0.0024	2.26	2.72
8	0.0019	0.0023	2.16	2.53
9	0.0018	0.0021	2.14	2.46
10	0.0017	0.002	2.09	2.41
11	0.0016	0.0019	2.04	2.37
12	0.0015	0.0019	2.01	2.45
13	0.0015	0.0019	1.9	2.27
14	0.0014	0.0018	1.92	2.37
15	0.0014	0.0018	1.98	2.3
16	0.0013	0.0018	1.87	2.22
17	0.0013	0.0017	1.84	2.12
18	0.0013	0.0017	1.8	2.14
19	0.0012	0.0016	1.85	2.17
20	0.0012	0.0016	1.81	2.21

Tabla 5-1.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 32 celdas LSTM.

Arquitectura con 64 celdas LSTM				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0142	0.0078	3.98	4.52
2	0.0038	0.0033	2.99	3.32
3	0.0025	0.0028	2.43	2.76
4	0.002	0.0024	2.26	2.64
5	0.0017	0.0022	2.10	2.60
6	0.0016	0.002	1.95	2.35
7	0.0014	0.002	1.88	2.41
8	0.0013	0.0018	1.93	2.32
9	0.0013	0.0017	1.88	2.39
10	0.0012	0.0017	1.79	2.38
11	0.0011	0.0016	1.77	2.28
12	0.0011	0.0016	1.71	2.38
13	0.0011	0.0015	1.72	2.19
14	0.001	0.0015	1.67	2.32
15	0.0009854	0.0016	1.63	2.14
16	0.00095905	0.0015	1.67	2.15
17	0.000932291	0.0015	1.64	2.14
18	0.00090677	0.0015	1.58	2.10
19	0.00088654	0.0016	1.59	2.20
20	0.00086792	0.0015	1.57	2.18

Tabla 5-2.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM.

Arquitectura con 96 celdas LSTM				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0112	0.0046	3.28	3.63
2	0.0028	0.0028	2.48	2.88
3	0.002	0.0024	1.93	2.46
4	0.0017	0.0021	2.02	2.49
5	0.0014	0.0021	1.99	2.55
6	0.0013	0.0019	1.9	2.44
7	0.0012	0.0018	1.91	2.45
8	0.0011	0.0019	1.83	2.34

9	0.0011	0.0018	1.78	2.32
10	0.00095712	0.0018	1.95	2.41
11	0.00091736	0.0018	1.75	2.33
12	0.00085717	0.0019	1.79	2.31
13	0.0008653	0.0018	1.71	2.35
14	0.00078737	0.002	1.92	2.60
15	0.00074864	0.002	1.82	2.26
16	0.0007211	0.002	1.75	2.33
17	0.00070236	0.0021	1.7	2.30
18	0.00066977	0.0021	1.39	2.53
19	0.00065503	0.0021	1.39	2.45
20	0.00062505	0.0023	1.33	2.50

Tabla 5-3.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 96 celdas LSTM.

En las figuras 5-3 y 5-4 se recopiló la información obtenida en las tres tablas anteriores para poder evidenciar de manera gráfica el comportamiento que presentaban los modelos ante la variación de las épocas de entrenamiento y las celdas LSTM, se observa que con pocas épocas de entrenamiento los tres modelos ya alcanzan un valor aceptable de error en sus predicciones manteniendo una proximidad entre los valores obtenidos para el conjunto de entrenamiento y el conjunto de prueba. Cada modelo alcanza un mínimo en el conjunto de prueba en 17 épocas para 32 celdas LSTM, 18 épocas para 64 celdas LSTM y 15 épocas para 96 celdas LSTM.

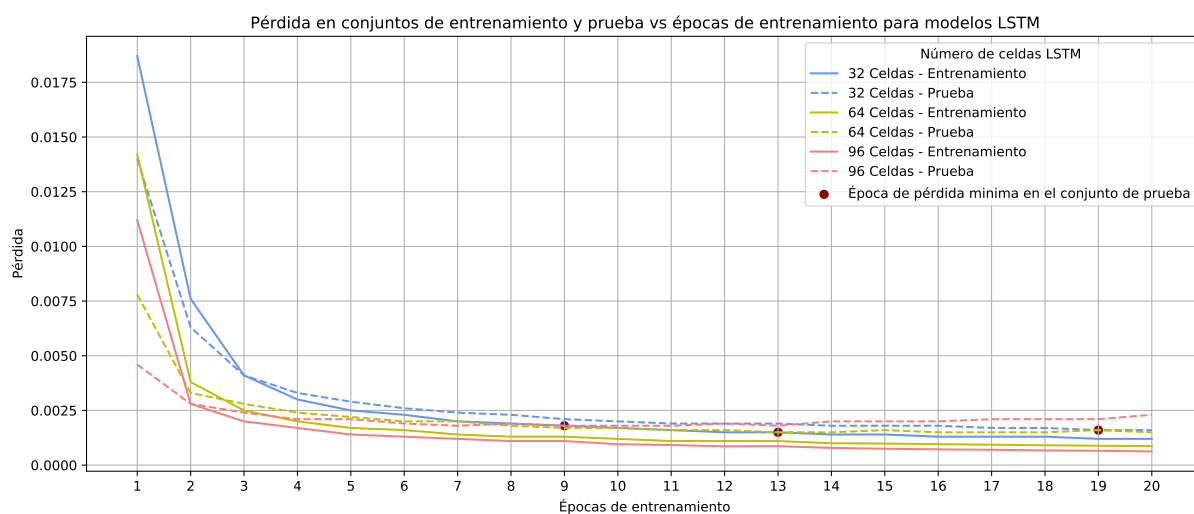


Figura 5-3.: Pérdida de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.

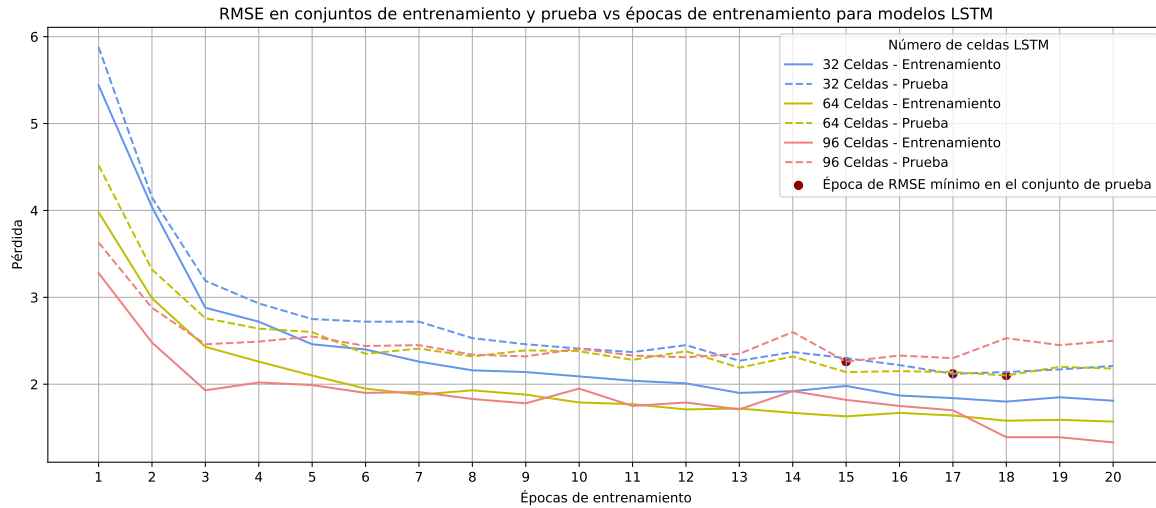


Figura 5-4.: RMSE de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.

A partir de los mínimos valores obtenidos para los tres modelos LSTM en el horno línea 1 en la tabla 5-4 se compararon los valores RMSE para el conjunto de prueba que obtuvo cada uno, siendo el modelo con 64 celdas y 18 épocas de entrenamiento el que obtuvo un mejor RMSE. A partir de esto se seleccionó para ser comparado junto con las demás arquitecturas desarrolladas a lo largo de la sección. En la tabla 5-5 se describe la dimensión de esta red neuronal junto con el total de parámetros que están disponibles para entrenamiento.

Unidades LSTM	Época de entrenamiento	Pérdida Prueba	RMSE Prueba
32 Celdas	17	0.0017	2.12
64 Celdas	18	0.0015	2.10
96 Celdas	15	0.0020	2.26

Tabla 5-4.: Mejores resultados para cada tipo de arquitectura LSTM.

Capa	Dimensiones de la capa de salida	Numero de parámetros
LSTM	(Ninguno, Ninguno, 64)	29.184
Densa	(Ninguno, Ninguno, 16)	1.040
Total de parámetros entrenables:		30.224

Tabla 5-5.: Arquitectura interna red neuronal LSTM seleccionada - Horno línea 1.

Modelos GRU

Al igual que en la arquitectura LSTM para la arquitectura GRU se desarrollaron 3 variaciones de modelos GRU a partir de las unidades que lo componen, partiendo de una arquitectura con 100 unidades GRU (**Tabla 5-6**), aumentado a 200 unidades GRU (**Tabla 5-7**) y finalizando con 300 unidades GRU (**Tabla 5-8**).

Arquitectura con 100 unidades GRU				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0129	0.0053	3.89	4.33
2	0.0034	0.0031	2.62	2.90
3	0.0022	0.0025	2.42	2.7
4	0.0018	0.0021	2.14	2.39
5	0.0016	0.0018	1.99	2.21
6	0.0014	0.0017	1.90	2.14
7	0.0013	0.0016	1.82	2.16
8	0.0012	0.0014	1.80	2.00
9	0.0011	0.0013	1.74	2.00
10	0.0011	0.0013	1.69	2.05
11	0.001	0.0013	1.69	2.02
12	0.00096621	0.0012	1.66	1.91
13	0.00093411	0.0012	1.59	1.89
14	0.00089396	0.0012	1.58	1.91
15	0.0008659	0.0013	1.58	1.93
16	0.0008393	0.0016	1.52	1.89
17	0.0008173	0.0018	1.50	1.94
18	0.00078606	0.0023	1.48	1.98
19	0.00076314	0.0023	1.46	2.10
20	0.00075198	0.0027	1.44	2.57

Tabla 5-6.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 100 unidades GRU.

Arquitectura con 200 unidades GRU				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0102	0.0041	3.07	3.28
2	0.0026	0.0027	2.41	2.66
3	0.0018	0.0018	2.13	2.37
4	0.0015	0.0015	1.91	2.11
5	0.0013	0.0014	1.8	2.07
6	0.0011	0.0014	1.74	2.01
7	0.0011	0.0012	1.64	1.93
8	0.00096677	0.0012	1.61	1.86
9	0.0009239	0.0012	1.54	1.84
10	0.00087748	0.0012	1.52	1.89
11	0.00082975	0.0014	1.52	1.87
12	0.00078816	0.0014	1.46	1.98
13	0.00079927	0.0016	1.5	2.03
14	0.00072974	0.0017	1.64	2.01
15	0.00072154	0.0014	1.39	2.21
16	0.00069185	0.0021	1.46	1.89
17	0.00066383	0.0022	1.31	2.43
18	0.0006398	0.002	1.33	1.86
19	0.0006107	0.0014	1.27	2.05
20	0.00059652	0.0015	1.29	2.08

Tabla 5-7.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 200 unidades GRU.

Arquitectura con 300 unidades GRU				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0083	0.0036	2.81	3.13
2	0.0022	0.0024	2.15	2.46
3	0.0016	0.0017	1.89	2.10
4	0.0012	0.0014	1.84	2.00
5	0.0011	0.0012	1.68	1.90
6	0.00099312	0.0012	1.62	1.87
7	0.00090953	0.0013	1.59	1.85
8	0.000847	0.0011	1.51	1.84

9	0.00080302	0.0011	1.51	1.84
10	0.00076149	0.0011	1.45	1.87
11	0.00074087	0.0011	1.41	1.98
12	0.00068824	0.0011	1.40	2.00
13	0.00064082	0.0012	1.37	2.35
14	0.00060453	0.0012	1.32	2.11
15	0.00056	0.0013	1.28	2.04
16	0.0005242	0.0013	1.25	1.99
17	0.0004937	0.0015	1.22	2.08
18	0.000466	0.0014	1.19	2.21
19	0.0004402	0.0014	1.15	2.16
20	0.00042545	0.0015	1.08	2.06

Tabla 5-8.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU.

En las figuras 5-5 y 5-6 se recopiló la información obtenida en las tres tablas anteriores para poder evidenciar de manera gráfica el comportamiento que presentaban los modelos ante la variación de las épocas de entrenamiento y las unidades GRU, se observa que a diferencia de los modelos LSTM se demora un poco más de épocas de entrenamiento para alcanzar, sin embargo alcanza valores mínimos en muchas menos épocas para el conjunto de prueba siendo 13 épocas para 100 unidades GRU, 9 épocas para 200 unidades GRU y 8 épocas para 300 unidades GRU. Adicionalmente se evidencia el fenómeno de sobreajuste pues a medida que avanzan el número de épocas de entrenamiento el RMSE para el conjunto de prueba tiende a aumentar mientras que para el conjunto de entrenamiento sigue disminuyendo.

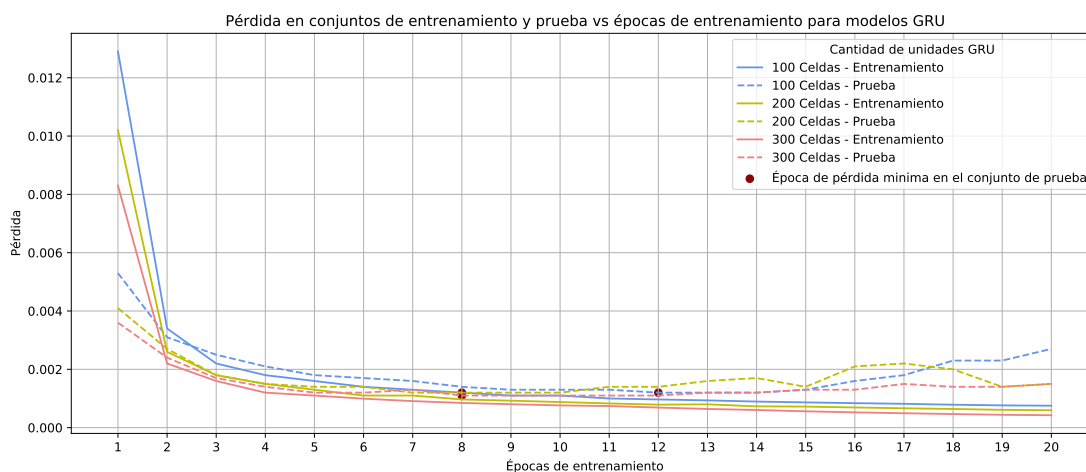


Figura 5-5.: Pérdida de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.

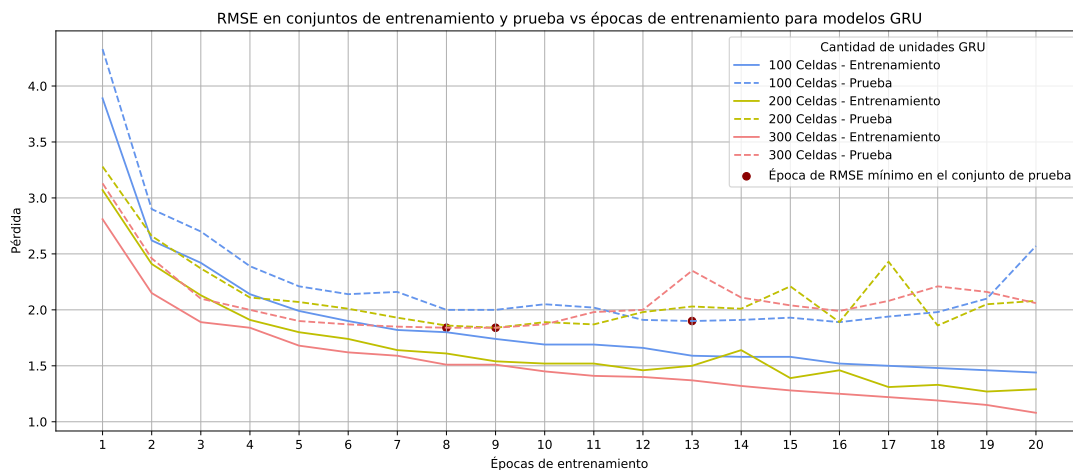


Figura 5-6.: RMSE de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.

A partir de los mínimos valores obtenidos para los tres modelos GRU en el horno línea 1 en la tabla 5-9 se compararon los valores RMSE para el conjunto de prueba que obtuvo cada uno, siendo el mejor igual para los modelos con 200 y 300 unidades, sin embargo, al modelo con 300 unidades GRU le toma una época menos de entrenamiento llegar a dicho valor, lo que supone una ventaja en tiempos de entrenamiento, A partir de esto se seleccionó para ser comparado junto con las demás arquitecturas desarrolladas a lo largo de la sección. En la tabla 5-10 se describe la dimensión de esta red neuronal junto con el total de parámetros que están disponibles para entrenamiento.

Unidades GRU	Época de entrenamiento	Pérdida Prueba	RMSE Prueba
100 unidades	13	0.0012	1.89
200 unidades	9	0.0012	1.84
300 unidades	8	0.0011	1.84

Tabla 5-9.: Mejores resultados para cada tipo de arquitectura GRU.

Capa	Dimensiones de la capa de salida	Numero de parámetros
GRU	(Ninguno, Ninguno, 300)	315.900
Densa	(Ninguno, Ninguno, 16)	4.816
Total de parámetros entrenables:		320.716

Tabla 5-10.: Arquitectura interna red neuronal GRU seleccionada - Horno línea 1.

Modelos LSTM con atención

Una vez realizado el entrenamiento y selección de un modelo basado en la arquitectura LSTM se le incorporo una arquitectura de atención para estudiar los efectos que tendría sobre este modelo. En la tabla **5-11** se observa que el mínimo RMSE para el conjunto de prueba del modelo LSTM con atención disminuyo y se alcanzó en dos épocas menos de entrenamiento que su predecesor, lo cual muestra las ventajas y evoluciones de la inclusión de las arquitecturas de atención en los modelos RNN básicos.

Arquitectura con 64 celdas LSTM con mecanismos de atención.				
Épocas	Pérdida entrenamiento	Pérdida Prueba	RMSE entrenamiento	RMSE Prueba
1	0.0088	0.0037	2.73	2.82
2	0.0024	0.0032	2.45	2.64
3	0.002	0.0028	2.24	2.37
4	0.0018	0.0026	2.17	2.25
5	0.0018	0.0026	2.2	2.33
6	0.0017	0.0025	2.04	2.13
7	0.0017	0.0025	2.08	2.15
8	0.0017	0.0025	2.05	2.13
9	0.0016	0.0025	2.12	2.27
10	0.0016	0.0025	2.09	2.22
11	0.0016	0.0025	2.05	2.15
12	0.0016	0.0026	2.03	2.14
13	0.0016	0.0025	2.01	2.09
14	0.0016	0.0025	1.99	2.05
15	0.0016	0.0025	2.04	2.11
16	0.0015	0.0025	1.989	2.03
17	0.0015	0.0024	1.99	2.05
18	0.0015	0.0024	2.038	2.133
19	0.0015	0.0024	2.031	2.108
20	0.0014	0.0024	1.99	2.07

Tabla 5-11.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM con mecanismos de atención.

En las figura **5-7** y **5-8** se recopiló la información obtenida en la tabla anterior para poder evidenciar de manera gráfica el comportamiento que presentaba el modelos ante la variación de las épocas de entrenamiento y las celdas LSTM, se observa que nuevamente con pocas épocas de entrenamiento ya se alcanza un valor aceptable de error en las predicciones disminuyendo aún más la proximidad existente entre los valores obtenidos para el conjunto de

entrenamiento y el conjunto de prueba, lo cual muestra un buen desempeño del modelo y de la arquitectura de atención.

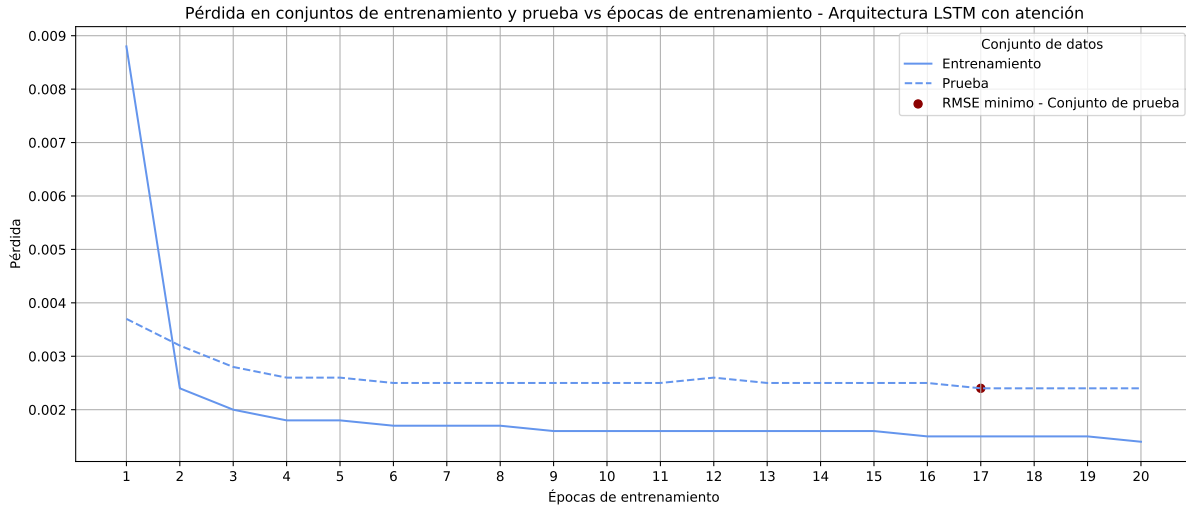


Figura 5-7.: Pérdida del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.

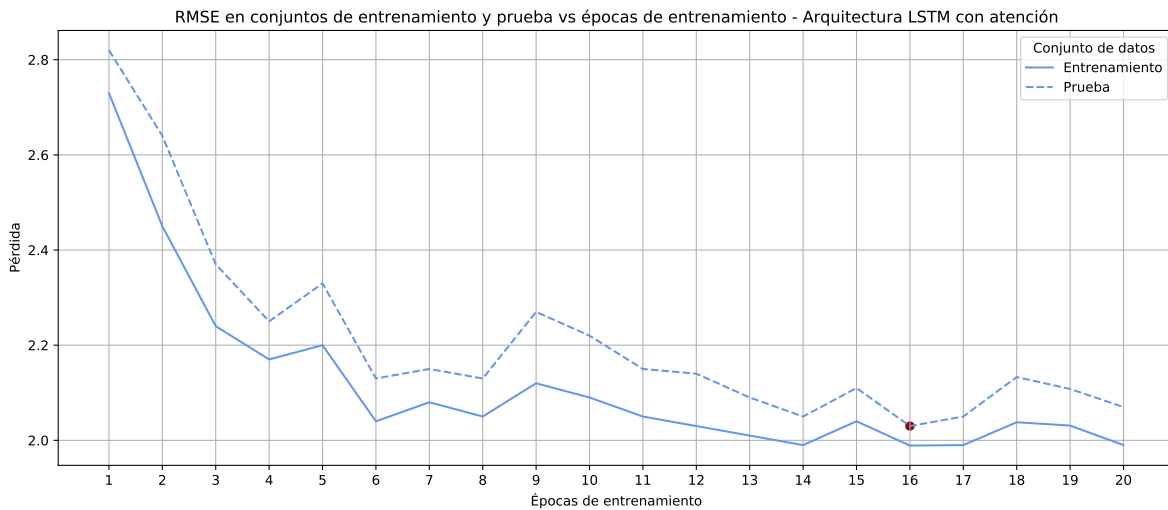


Figura 5-8.: RMSE del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.

Por último, en la tabla 5-12 se observa la compleja arquitectura de los modelos con atención, añadiendo más capas y conexiones lo que aumenta las dimensiones y parámetros entrenables en comparación a los modelos sin atención.

Capa	Dimensiones de la capa de salida	Numero de parámetros	Conexión
Capa de entrada	(Ninguno, Ninguno, 49)	0	–
LSTM	(Ninguno, Ninguno, 64)	29.184	Capa de entrada
Ultimo estado oculto	(Ninguno, 64)	0	LSTM
Vector de puntajes de atención	(Ninguno, Ninguno, 64)	4.096	LSTM
Puntaje de atención	(Ninguno, Ninguno)	0	Ultimo estado oculto y Vector de puntajes de atención
Pesos de atención	(Ninguno, Ninguno)	0	Puntaje de atención
Vector de contexto	(Ninguno, 64)	0	GRU y Pesos de atención
Salida de atención	(Ninguno, 128)	0	Vector de contexto y Ultimo estado oculto
Vector de atención	(Ninguno, 96)	12.288	Salida de atención
Densa	(Ninguna, 16)	1.552	Vector de atención
Total de parámetros entrenables:			47.120

Tabla 5-12.: Arquitectura interna red neuronal LSTM con mecanismos de atención - Horno línea 1.

Modelos GRU con atención

Una vez realizado el entrenamiento y selección de un modelo basado en la arquitectura GRU se le incorporo una arquitectura de atención para estudiar los efectos que tendría sobre este modelo. En la tabla **5-13** se observa que el mínimo RMSE para el conjunto de prueba del modelo GRU con atención aumentó un poco y por el contrario requirió de mas épocas de entrenamiento para alcanzar el valor RMSE mínimo.

Arquitectura con 300 unidades GRU con mecanismos de atención				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE Prueba
1	0.0093	0.0045	3.25	3.21
2	0.0033	0.0039	2.72	2.97
3	0.0024	0.0031	2.39	2.57
4	0.002	0.0029	2.22	2.42
5	0.0016	0.0027	2.19	2.3
6	0.0017	0.0026	2.06	2.15
7	0.0017	0.0024	2.05	2.09

8	0.0017	0.0025	2.04	2.08
9	0.0016	0.0024	2	2.02
10	0.0016	0.0024	1.99	2.05
11	0.0016	0.0023	1.973	1.98
12	0.0015	0.0023	1.96	1.99
13	0.0015	0.0023	1.96	1.99
14	0.0015	0.0023	1.94	1.96
15	0.0015	0.0023	1.93	1.96
16	0.0015	0.0023	1.93	1.95
17	0.0015	0.0024	1.92	1.99
18	0.0015	0.0022	1.92	1.93
19	0.0015	0.0023	1.91	1.96
20	0.0015	0.0022	1.91	1.94

Tabla 5-13.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU con mecanismos de atención.

En las figura 5-9 y 5-10 se recopiló la información obtenida en la tabla anterior para poder evidenciar de manera gráfica el comportamiento que presentaba el modelos ante la variación de las épocas de entrenamiento y las celdas GRU, se observa que nuevamente con pocas épocas de entrenamiento ya se alcanza un valor aceptable de error en las predicciones disminuyendo aún más la proximidad existente entre los valores obtenidos para el conjunto de entrenamiento y el conjunto de prueba, lo cual muestra un buen desempeño del modelo y de la arquitectura de atención.



Figura 5-9.: Pérdida del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.



Figura 5-10.: RMSE del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.

Por último, en la tabla 5-14 se observa la compleja arquitectura de los modelos con atención, añadiendo más capas y conexiones lo que aumenta las dimensiones y parámetros entrenables en comparación a los modelos sin atención.

Capa	Dimensiones de la capa de salida	Numero de parámetros	Conexión
Capa de entrada	(Ninguno, Ninguno, 49)	0	—
GRU	(Ninguno, Ninguno, 300)	315.900	Capa de entrada
Ultimo estado oculto	(Ninguno, 300)	0	GRU
Vector de puntajes de atención	(Ninguno, Ninguno, 300)	90.000	GRU
Puntaje de atención	(Ninguno, Ninguno)	0	Ultimo estado oculto y Vector de puntajes de atención
Pesos de atención	(Ninguno, Ninguno)	0	Puntaje de atención
Vector de contexto	(Ninguno, 300)	0	GRU y Pesos de atención
Salida de atención	(Ninguno, 600)	0	Vector de contexto y Ultimo estado oculto
Vector de atención	(Ninguno, 96)	57.600	Salida de atención
Densa	(Ninguna, 16)	1.552	Vector de atención
Total de parámetros entrenables:		465.052	

Tabla 5-14.: Arquitectura interna red neuronal GRU con mecanismos de atención - Horno línea 1.

5.2.2. Modelos horno línea 2

Modelos LSTM

Para el horno línea 2 se desarrollaron bajo el mismo patrón 3 variaciones de modelos LSTM aumentando cada vez más el número de celdas LSTM empleadas para el entrenamiento y predicción del modelo, se empezó con una arquitectura de 32 celdas LSTM (**Tabla 5-15**), siguiendo con un aumento del doble de celdas para 64 celdas LSTM (**Tabla 5-16**) y finalizando con 96 celdas LSTM (**Tabla 5-17**).

Arquitectura con 32 celdas LSTM				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0154	0.0119	3.38	4.59
2	0.0032	0.0036	2.05	2.46
3	0.0022	0.0029	1.86	2.2
4	0.0017	0.0029	1.66	2.07
5	0.0014	0.0023	1.54	1.93
6	0.0012	0.0028	1.46	2.04
7	0.000984	0.0025	1.35	1.97
8	0.001	0.002	1.36	1.74
9	0.00098052	0.002	1.32	1.75
10	0.00084304	0.0024	1.25	1.87
11	0.00096967	0.0015	1.31	1.58
12	0.00069805	0.0017	1.13	1.62
13	0.00069025	0.0017	1.12	1.59
14	0.00063747	0.0017	1.08	1.6
15	0.0006493	0.0017	1.1	1.68
16	0.00063353	0.0019	1.08	1.8
17	0.00058349	0.0017	1.03	1.6
18	0.00053931	0.0016	1	1.59
19	0.00057461	0.0012	1.04	1.38
20	0.0005646	0.0012	1.04	1.41

Tabla 5-15.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 32 celdas LSTM.

Arquitectura con 64 celdas LSTM				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0106	0.0048	2.43	2.94
2	0.0022	0.0039	1.81	2.55
3	0.0014	0.0028	1.48	2.02
4	0.0012	0.0021	1.39	1.86
5	0.00096414	0.0023	1.34	1.92
6	0.00081257	0.002	1.24	1.88
7	0.00071013	0.0019	1.14	1.67
8	0.00064715	0.0016	1.08	1.57
9	0.00064655	0.0017	1.07	1.66
10	0.00056233	0.0019	1	1.68
11	0.00054376	0.002	0.99	1.78
12	0.00052791	0.0018	0.98	1.64
13	0.00047112	0.0014	0.94	1.47
14	0.00044418	0.0024	0.93	1.9
15	0.00041713	0.0019	0.88	1.77
16	0.00039116	0.0016	0.87	1.58
17	0.00039738	0.0016	0.92	1.58
18	0.00037349	0.002	0.85	1.72
19	0.00037485	0.0016	0.86	1.61
20	0.00034675	0.0015	0.8	1.56

Tabla 5-16.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM.

Arquitectura con 96 celdas LSTM				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0088	0.004	1.98	2.47
2	0.0018	0.0028	1.58	2.07
3	0.0011	0.0022	1.27	1.86
4	0.00091513	0.0025	1.23	1.98
5	0.0007249	0.0022	1.11	1.86
6	0.00060366	0.0022	1.01	1.82
7	0.000572	0.0019	1.1	1.76
8	0.00044948	0.0018	0.91	1.71

9	0.00046928	0.0017	0.9	1.63
10	0.00040345	0.0019	0.86	1.72
11	0.00038634	0.0014	0.83	1.55
12	0.00039607	0.0018	0.86	1.68
13	0.00035938	0.0017	0.81	1.7
14	0.000321	0.002	0.77	1.75
15	0.00032229	0.0018	0.82	1.72
16	0.0002939	0.0021	0.74	1.89
17	0.00024357	0.0019	0.68	1.7
18	0.00025943	0.002	0.7	1.82
19	0.00026024	0.0019	0.7	1.76
20	0.00022838	0.0017	0.68	1.69

Tabla 5-17.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 96 celdas LSTM.

En las figuras 5-11 y 5-12 se recopiló la información obtenida en las tres tablas anteriores para poder evidenciar de manera gráfica el comportamiento que presentaban los modelos ante la variación de las épocas de entrenamiento y las celdas LSTM, se observa que con pocas épocas de entrenamiento los tres modelos al igual que en el horno línea 1, ya alcanzan un valor aceptable de error en sus predicciones a diferencia de una leve separación entre los valores obtenidos para el conjunto de entrenamiento y el conjunto de prueba. Cada modelo alcanzó un mínimo en el conjunto de prueba en 19 épocas para 32 celdas LSTM, 13 épocas para 64 celdas LSTM y 11 épocas para 96 celdas LSTM.

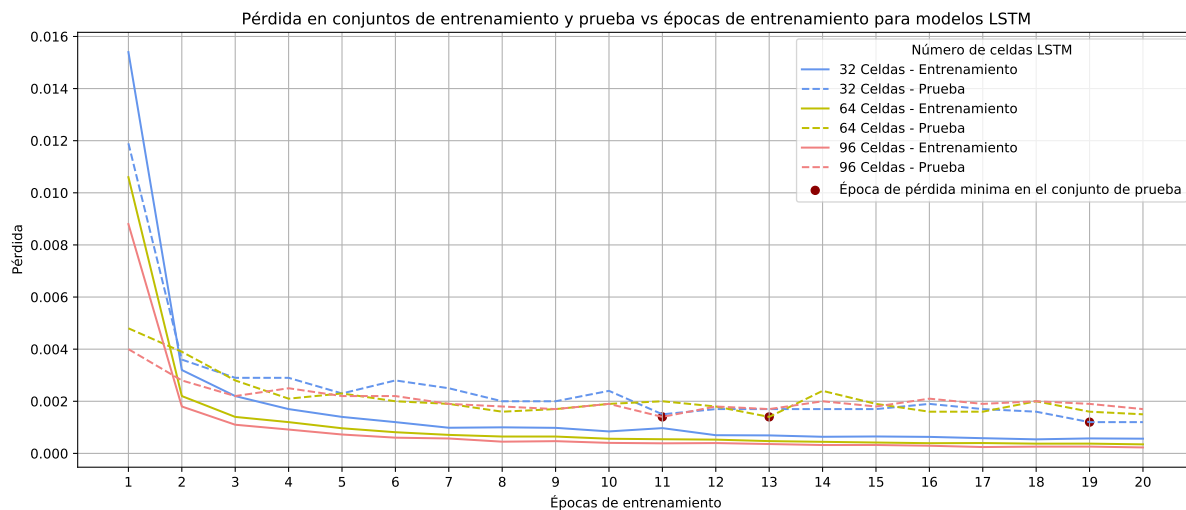


Figura 5-11.: Pérdida de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.

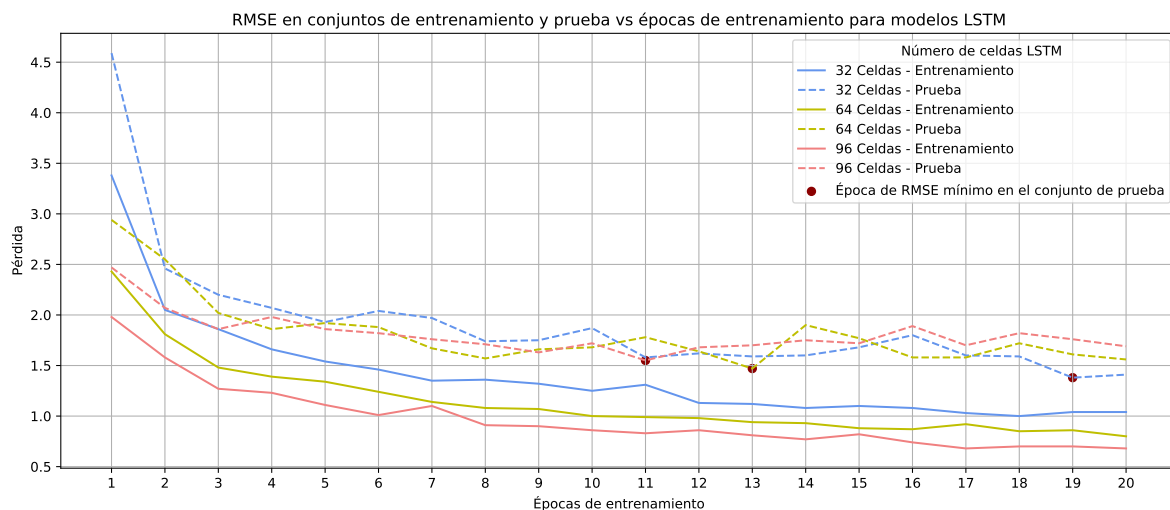


Figura 5-12.: RMSE de los modelos LSTM con 32, 64 y 96 celdas a partir de las épocas de entrenamiento.

A partir de los mínimos valores obtenidos para los tres modelos LSTM en el horno línea 2 en la tabla 5-18 se compararon los valores RMSE para el conjunto de prueba que obtuvo cada uno, siendo el modelo con 32 celdas y 19 épocas de entrenamiento el que obtuvo un mejor RMSE. A partir de esto se seleccionó para ser comparado junto con las demás arquitecturas que se desarrollaran más adelante, en la tabla 5-19 se describe la dimensión de esta red neuronal junto con el total de parámetros que están disponibles para entrenamiento, los cuales son mucho menores que en el caso del horno línea 1 debido a la variación del tamaño de 64 a 32 celdas LSTM.

Unidades LSTM	Época de entrenamiento	Pérdida Prueba	RMSE Prueba
32 Celdas	19	0.0012	1.38
64 Celdas	13	0.0014	1.47
96 Celdas	11	0.0014	1.55

Tabla 5-18.: Mejores resultados para cada tipo de arquitectura LSTM.

Capa	Dimensiones de la capa de salida	Numero de parámetros
LSTM	(Ninguno, Ninguno, 32)	13.952
Densa	(Ninguno, Ninguno, 16)	528
Total de parámetros entrenables:		14.480

Tabla 5-19.: Arquitectura interna red neuronal LSTM seleccionada - Horno línea 2.

Modelos GRU

Nuevamente para la arquitectura GRU aplicada al horno línea 2 se desarrollaron 3 variaciones de modelos GRU a partir de las unidades que lo componen, partiendo de una arquitectura con 100 unidades GRU (**Tabla 5-20**), aumentado a 200 unidades GRU (**Tabla 5-21**) y finalizando con 300 unidades GRU (**Tabla 5-22**).

Arquitectura con 100 unidades GRU				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0115	0.0056	2.24	2.26
2	0.0022	0.0024	1.77	1.98
3	0.0015	0.002	1.51	1.91
4	0.0011	0.0019	1.36	1.8
5	0.00092113	0.0018	1.24	1.61
6	0.00078596	0.0017	1.18	1.69
7	0.00069974	0.0016	1.15	1.46
8	0.00062305	0.0017	1.07	1.44
9	0.00057221	0.0015	1.02	1.47
10	0.00052893	0.0014	0.97	1.4
11	0.00049815	0.0018	0.94	1.41
12	0.00047054	0.0013	0.94	1.44
13	0.00044024	0.0011	0.88	1.36
14	0.00041465	0.0011	0.93	1.3
15	0.00040228	0.0011	0.86	1.25
16	0.00039146	0.001	0.86	1.28
17	0.00036051	0.00099479	0.84	1.20
18	0.00035444	0.001	0.82	1.27
19	0.00033819	0.001	0.81	1.25
20	0.00034386	0.00096094	0.81	1.24

Tabla 5-20.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 100 unidades GRU.

Arquitectura con 200 unidades GRU				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0076	0.0027	1.92	2.06
2	0.0016	0.0021	1.5	1.85
3	0.0011	0.0017	1.29	1.67
4	0.00083954	0.001	1.29	1.49
5	0.00068753	0.0014	1.14	1.53
6	0.00059611	0.0015	1	1.36
7	0.00052343	0.0012	0.95	1.3
8	0.00047649	0.0011	0.94	1.33
9	0.00043555	0.00097451	0.87	1.25
10	0.00041097	0.00094459	0.82	1.21
11	0.00038107	0.00085133	0.8	1.24
12	0.00034187	0.00084618	0.82	1.19
13	0.00033169	0.00084888	0.79	1.24
14	0.00030649	0.00089513	0.75	1.19
15	0.00029786	0.00081904	0.74	1.19
16	0.00027248	0.00087422	0.73	1.22
17	0.00026667	0.00080395	0.69	1.12
18	0.00025463	0.00082721	0.74	1.2
19	0.00024803	0.00095942	0.69	1.23
20	0.0002335	0.00088526	0.64	1.15

Tabla 5-21.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 200 unidades GRU.

Arquitectura con 300 unidades GRU				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0065	0.0027	1.79	1.95
2	0.0014	0.0018	1.39	1.67
3	0.0009549	0.0016	1.2	1.54
4	0.00075013	0.0014	1.08	1.46
5	0.00058189	0.0013	1.3	1.74
6	0.00050281	0.0011	0.93	1.29
7	0.00046528	0.0011	0.86	1.2
8	0.00038105	0.0008942	0.94	1.33

9	0.00035944	0.00090535	0.79	1.19
10	0.00034041	0.00078909	0.76	1.18
11	0.00033613	0.00084633	0.73	1.15
12	0.00027426	0.00083749	0.72	1.12
13	0.0002685	0.00079783	0.71	1.15
14	0.00026791	0.0008472	0.65	1.14
15	0.00023004	0.00083739	0.65	1.16
16	0.00022836	0.00091019	0.7	1.21
17	0.00021539	0.00089236	0.7	1.17
18	0.00020625	0.00091765	0.61	1.15
19	0.00021452	0.00091797	0.62	1.25
20	0.00018367	0.00096625	0.57	1.23

Tabla 5-22.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU.

En las figuras 5-13 y 5-14 se recopiló la información obtenida en las tres tablas anteriores para poder evidenciar de manera gráfica el comportamiento que presentaban los modelos ante la variación de las épocas de entrenamiento y las unidades GRU, se observa nuevamente el efecto obtenido en el horno línea 1 donde este tipo de modelos se demoran más de épocas de entrenamiento para alcanzar un valor RMSE aceptable, el valor mínimo de cada modelo se obtiene en 17 épocas para 100 unidades GRU, 17 épocas para 200 unidades GRU y 12 épocas para 300 unidades GRU.

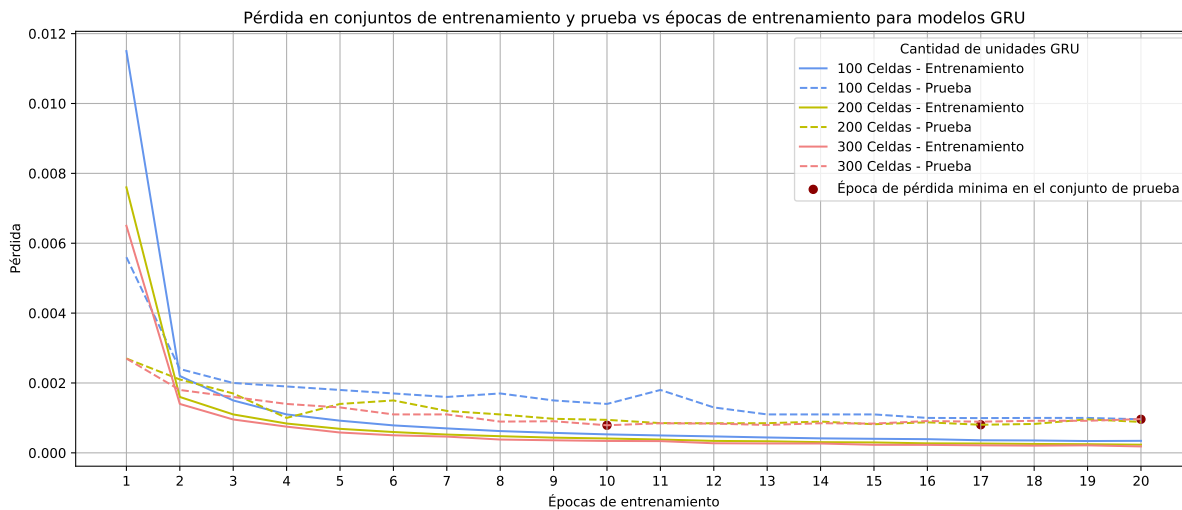


Figura 5-13.: Pérdida de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.

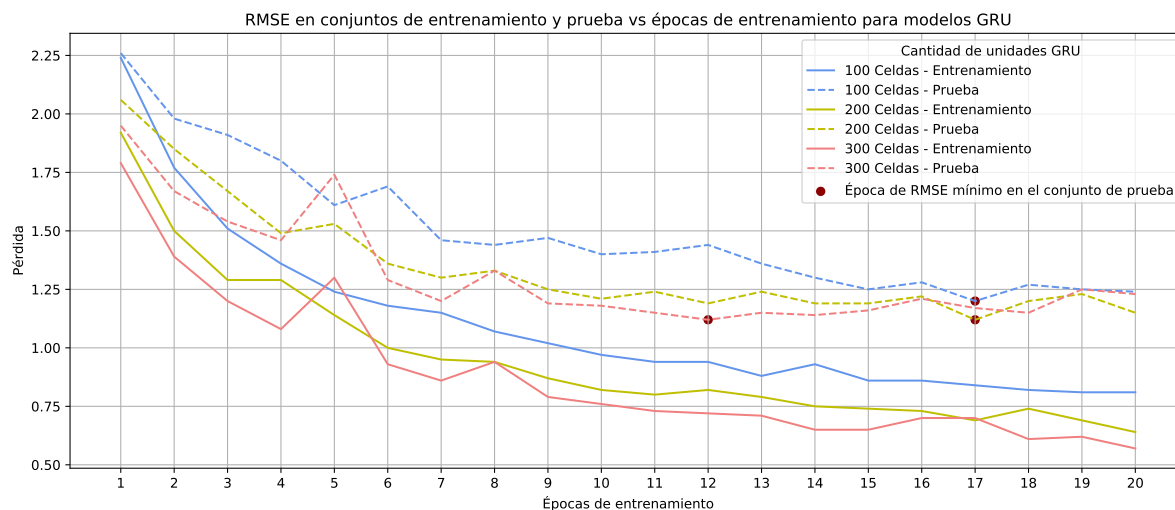


Figura 5-14.: RMSE de los modelos GRU con 100, 200 y 300 unidades a partir de las épocas de entrenamiento.

A partir de los mínimos valores obtenidos para los tres modelos GRU en el horno línea 1 en la tabla 5-23 se compararon los valores RMSE para el conjunto de prueba que obtuvo cada uno, siendo el mejor el modelo 300 unidades, A partir de esto se seleccionó para ser comparado junto con las demás arquitecturas desarrolladas a lo largo de la sección. En la tabla 5-24 se describe la dimensión de esta red neuronal junto con el total de parámetros que están disponibles para entrenamiento.

Unidades GRU	Época de entrenamiento	Pérdida Prueba	RMSE Prueba
100 unidades	17	0.00099	1.20
200 unidades	17	0.00080	1.12
300 unidades	12	0.00083749	1.12

Tabla 5-23.: Mejores resultados para cada tipo de arquitectura GRU.

Capa	Dimensiones de la capa de salida	Numero de parámetros
GRU	(Ninguno, Ninguno, 300)	340.200
Densa	(Ninguno, Ninguno, 16)	4.816
Total de parámetros entrenables:		345.016

Tabla 5-24.: Arquitectura interna red neuronal GRU seleccionada - Horno línea 2.

Modelos LSTM con atención

Una vez realizado el entrenamiento y selección de un modelo basado en la arquitectura LSTM para el horno línea 2, se incorporo una arquitectura de atención para estudiar los efectos que tendría sobre este modelo. En la tabla 5-25 se observa que el mínimo RMSE para el conjunto de prueba del modelo LSTM con atención aumento levemente y alcanzo su mínimo en las mismas épocas que su predecesor, lo cual muestra las ventajas y evoluciones de la inclusión de las arquitecturas de atención en los modelos RNN básicos.

Arquitectura con 300 unidades GRU con mecanismos de atención				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0095	0.0509	2.76	3.18
2	0.0021	0.0446	2.06	2.09
3	0.0017	0.0421	1.84	1.9
4	0.0015	0.0426	1.74	2.03
5	0.0014	0.0386	1.66	1.76
6	0.0013	0.0384	1.61	1.74
7	0.0012	0.0384	1.61	1.76
8	0.0012	0.0358	1.52	1.61
9	0.0011	0.0351	1.48	1.65
10	0.0011	0.0326	1.44	1.52
11	0.001	0.0348	1.45	1.55
12	0.001	0.0338	1.39	1.6
13	0.00099046	0.0343	1.42	1.55
14	0.00096329	0.0342	1.35	1.58
15	0.00093826	0.0348	1.35	1.6
16	0.00092512	0.0356	1.37	1.53
17	0.00090646	0.0362	1.34	1.47
18	0.00089343	0.037	1.34	1.54
19	0.000881	0.039	1.34	1.44
20	0.00086209	0.0378	1.34	1.59

Tabla 5-25.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 64 celdas LSTM con mecanismos de atención.

En las figura 5-15 y 5-16 se recopiló la información obtenida en la tabla anterior para poder evidenciar de manera gráfica el comportamiento que presentaba el modelos ante la variación de las épocas de entrenamiento y las celdas LSTM, sin embargo, para este modelo LSTM se observa que a diferencia de todos las anteriores se requiere de muchas mas épocas para alcanzar un valor aceptable de error en las predicciones, pero manteniendo la proximidad

existente entre los valores obtenidos para el conjunto de entrenamiento y el conjunto de prueba, lo cual muestra un buen desempeño del modelo y de la arquitectura de atención.

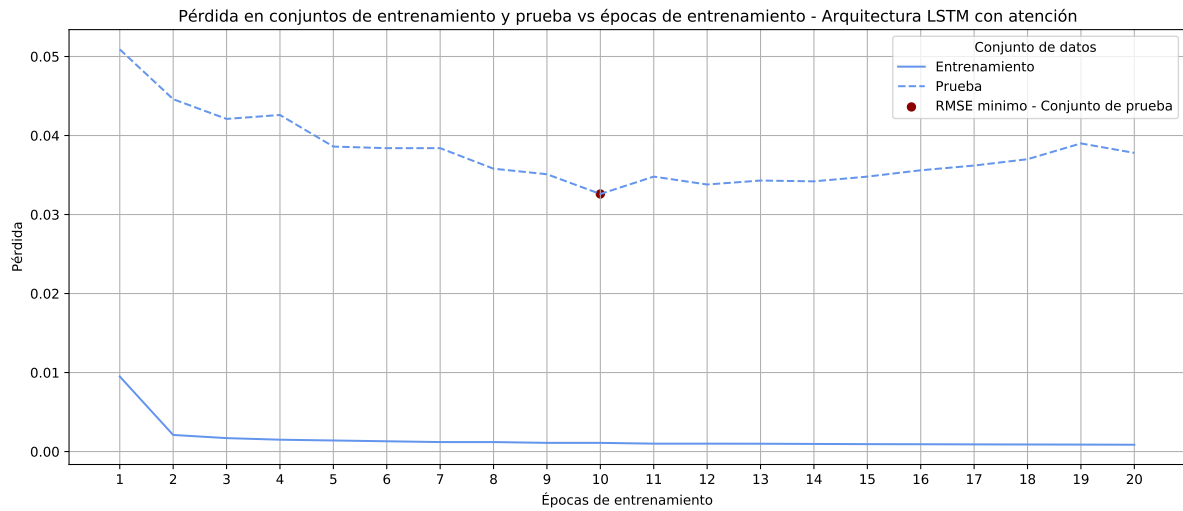


Figura 5-15.: Pérdida del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.

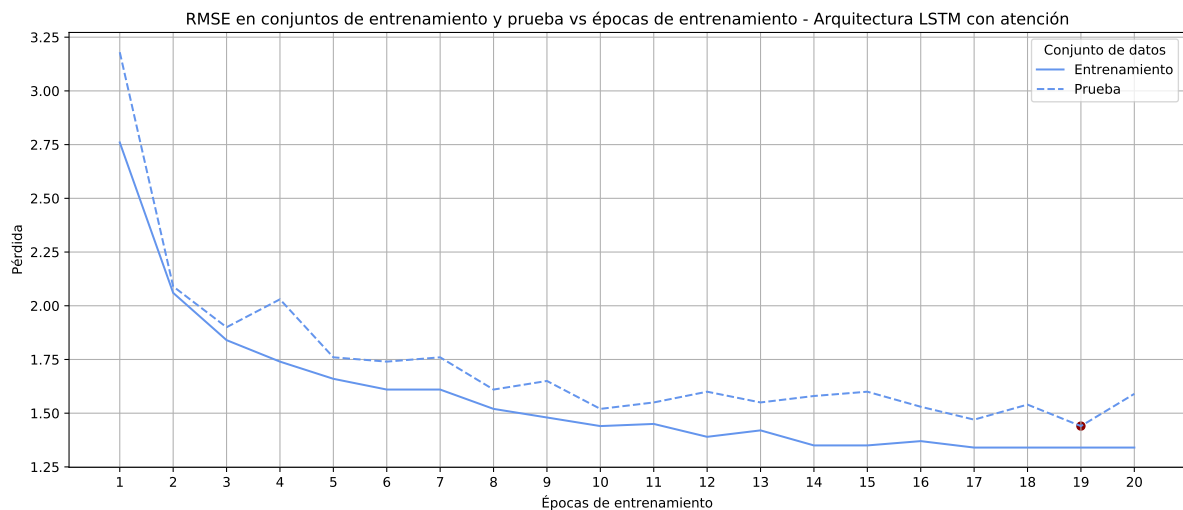


Figura 5-16.: RMSE del modelo con 64 celdas LSTM y mecanismos de atención a partir de las épocas de entrenamiento.

Por último, en la tabla 5-26 se observa la compleja arquitectura de los modelos con atención, añadiendo más capas y conexiones lo que aumenta las dimensiones y parámetros entrenables en comparación a los modelos sin atención.

Capa	Dimensiones de la capa de salida	Numero de parámetros	Conexión
Capa de entrada	(Ninguno, Ninguno, 76)	0	–
LSTM	(Ninguno, Ninguno, 32)	13.952	Capa de entrada
Ultimo estado oculto	(Ninguno, 32)	0	LSTM
Vector de puntajes de atención	(Ninguno, Ninguno, 32)	1.024	LSTM
Puntaje de atención	(Ninguno, Ninguno)	0	Ultimo estado oculto y Vector de puntajes de atención
Pesos de atención	(Ninguno, Ninguno)	0	Puntaje de atención
Vector de contexto	(Ninguno, 32)	0	GRU y Pesos de atención
Salida de atención	(Ninguno, 64)	0	Vector de contexto y Ultimo estado oculto
Vector de atención	(Ninguno, 96)	6.144	Salida de atención
Densa	(Ninguna, 16)	1.552	Vector de atención
Total de parámetros entrenables:			22.672

Tabla 5-26.: Arquitectura interna red neuronal LSTM con mecanismos de atención - Horno línea 2.

Modelos GRU con atención

Una vez realizado el entrenamiento y selección de un modelo basado en la arquitectura GRU se le incorporo una arquitectura de atención para estudiar los efectos que tendría sobre este modelo. En la tabla 5-27 se observa que el mínimo RMSE para el conjunto de prueba del modelo GRU con atención disminuyo requiriendo mas épocas de entrenamiento para alcanzar este valor.

Arquitectura con 300 unidades GRU con mecanismos de atención				
Épocas	Pérdida entrenamiento	Pérdida prueba	RMSE entrenamiento	RMSE prueba
1	0.0049	0.0418	1.701	1.743
2	0.0014	0.037	1.643	1.927
3	0.0011	0.0459	1.371	1.532
4	0.001	0.054	1.363	1.412
5	0.00094861	0.0577	1.308	1.381
6	0.00090506	0.064	1.287	1.367
7	0.00086542	0.0515	1.211	1.386

8	0.0008381	0.059	1.254	1.438
9	0.00079997	0.0388	1.162	1.159
10	0.00078792	0.0363	1.139	1.141
11	0.00078107	0.0285	1.164	1.219
12	0.00074992	0.0327	1.127	1.232
13	0.00074524	0.0269	1.115	1.141
14	0.00071486	0.0247	1.118	1.081
15	0.00070618	0.025	1.093	1.084
16	0.00068719	0.0269	1.124	1.136
17	0.00068396	0.0264	1.128	1.103
18	0.00066777	0.0273	1.092	1.17
19	0.00066822	0.0285	1.065	1.172
20	0.00063848	0.0276	1.048	1.147

Tabla 5-27.: Pérdidas de entrenamiento y RMSE para 20 épocas de entrenamiento con una arquitectura RNN de 300 unidades GRU con mecanismos de atención.

En las figura 5-17 y 5-18 se recopiló la información obtenida en la tabla anterior para poder evidenciar de manera gráfica el comportamiento que presentaba el modelos ante la variación de las épocas de entrenamiento y las celdas GRU, con la adición del mecanismo de atención se requiere para este modelo GRU de mas épocas de entrenamiento para alcanzar un valor aceptable de error en las predicciones, teniendo épocas de entrenamiento con una separación variable entre el RMSE del conjunto de entrenamiento y el conjunto de prueba.

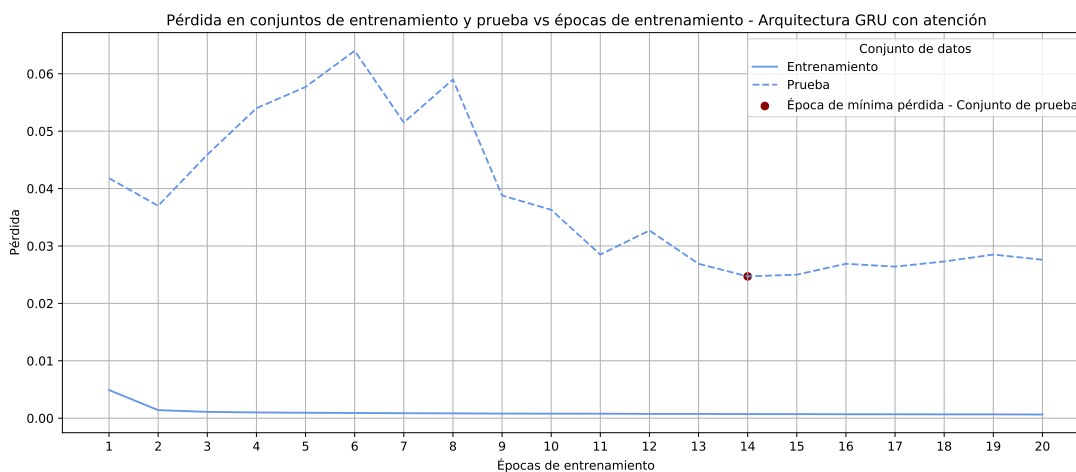


Figura 5-17.: Pérdida del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.

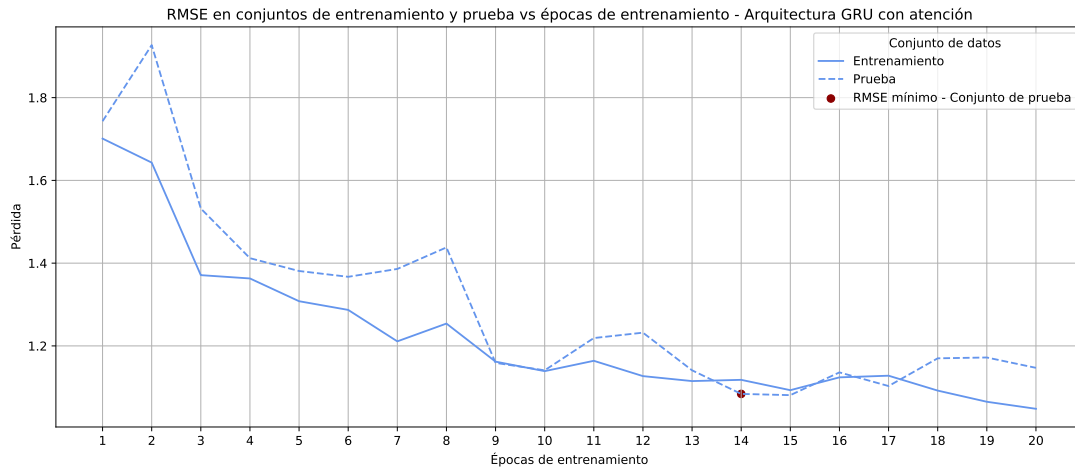


Figura 5-18.: RMSE del modelo con 300 unidades GRU y mecanismos de atención a partir de las épocas de entrenamiento.

Por último, en la tabla 5-28 se observa la compleja arquitectura de los modelos con atención, añadiendo más capas y conexiones lo que aumenta las dimensiones y parámetros entrenables en comparación a los modelos sin atención.

Capa	Dimensiones de la capa de salida	Numero de parámetros	Conexión
Capa de entrada	(Ninguno, Ninguno, 76)	0	—
GRU	(Ninguno, Ninguno, 300)	340.200	Capa de entrada
Ultimo estado oculto	(Ninguno, 300)	0	GRU
Vector de puntajes de atención	(Ninguno, Ninguno, 300)	90.000	GRU
Puntaje de atención	(Ninguno, Ninguno)	0	Ultimo estado oculto y Vector de puntajes de atención
Pesos de atención	(Ninguno, Ninguno)	0	Puntaje de atención
Vector de contexto	(Ninguno, 300)	0	GRU y Pesos de atención
Salida de atención	(Ninguno, 600)	0	Vector de contexto y Ultimo estado oculto
Vector de atención	(Ninguno, 96)	57.600	Salida de atención
Densa	(Ninguna, 16)	1.552	Vector de atención
Total de parámetros entrenables:			489.352

Tabla 5-28.: Arquitectura interna red neuronal GRU con mecanismos de atención - Horno línea 2.

6. Validación de arquitecturas RNN

A diferencia de las etapas anteriores de desarrollo llevadas a cabo para la preparación del conjunto de datos y el desarrollo de modelos RNN donde se siguieron flujos de trabajo secuenciales, para la validación de los modelos creados se estudió sobre ellos algunos escenarios como se observa en la figura 6-1, inicialmente se observó el comportamiento de los modelos en relación al tiempo de predicción, se continuo con el aumento de termocuplas a predecir y la validación cruzada para finalizar con la distribución del RMSE en las predicciones para el conjunto de prueba en cada una de las termocuplas.

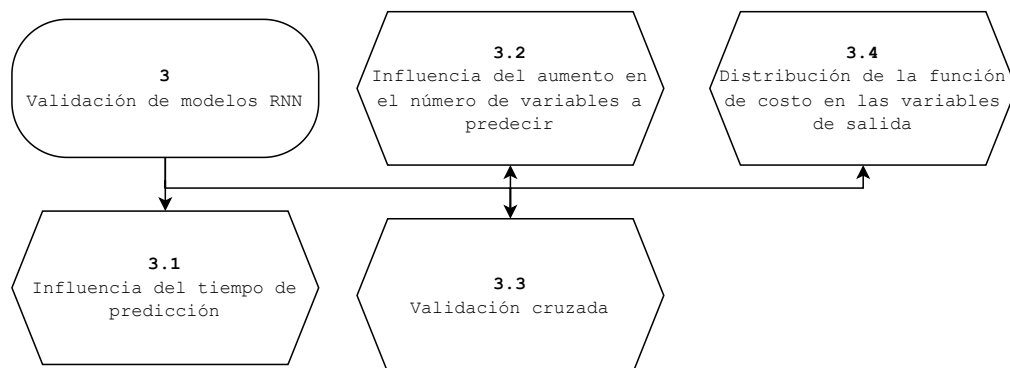


Figura 6-1.: Etapas específicas de validación de modelos desarrollados para la predicción de temperatura en los hornos línea 1 y línea 2.

El desarrollo realizado en las siguientes subsecciones está planteado a partir de los modelos con mejores resultados obtenidos en la sección anterior, de esta manera, se establecieron los mismos hiperparámetros para ser usados en cada una de las validaciones, esto con el fin de observar el comportamiento de las arquitecturas RNN ante los casos anteriormente mencionados, pues se entiende que en la práctica ante cualquier modificación que sufra el conjunto de datos (Aumento del tamaño de variables a predecir) y cualquier otro parámetro del sistema como por ejemplo el tiempo de predicción, requieren un reentrenamiento del modelo junto con un nuevo ajuste de los hiperparámetros del mismo, pues cosas como el número de épocas de entrenamiento o la cantidad de celdas LSTM o GRU pueden variar para obtener el menor RMSE.

6.1. Tiempo de predicción

El tiempo de predicción en modelos de series de tiempo es un parámetro importante a definir, pues un tiempo muy corto a pesar de brindar buenas predicciones puede no aportar mucha información cuando se requiere, mientras que un tiempo de predicción muy lejano hará que las predicciones tiendan a estar más alejadas de los valores reales cada vez más, es por esto, que se estudió el efecto de este parámetro en las predicciones realizadas para cada uno de los modelos desarrollados en los hornos línea 1 y línea 2, escogiendo como tiempo menor de predicción 1 hora y como tiempo máximo de predicción 7 horas, en pasos de 1 hora como se observa en la tabla **6-1**.

Modelos Línea 1	Tiempo de predicción [Horas]						
	1	2	3	4	5	6	7
LSTM (64 celdas - 18 épocas)	1.04	1.49	2.1	2.87	3.35	4.47	4.68
GRU (300 unidades - 8 épocas)	1	1.36	1.84	2.22	2.72	3.29	3.89
ATT + LSTM (64 celdas - 16 épocas)	1.01	1.51	2.03	2.43	2.89	3.36	3.64
ATT + GRU (300 unidades - 18 épocas)	0.9	1.46	1.93	2.44	2.88	3.24	3.59
Modelos Línea 2	Tiempo de predicción [Horas]						
	1	2	3	4	5	6	7
LSTM (32 celdas - 19 épocas)	1.07	1.25	1.38	1.96	2.56	2.63	2.78
GRU (300 unidades - 12 épocas)	0.8	1.08	1.12	1.34	1.65	1.93	2.11
ATT + LSTM (32 celdas - 19 épocas)	0.97	1.21	1.44	1.52	1.76	2.04	2.29
ATT + GRU (300 unidades - 15 épocas)	0.78	0.91	1.08	1.33	1.49	1.71	1.88

Tabla 6-1.: Comportamiento del RMSE para los modelos de predicción horno línea 1 y 2 ante la variación del tiempo de predicción.

En las figuras **6-2** y **6-3** se observa como a partir del aumento en el tiempo de predicción, el valor RMSE obtenido para las predicciones de cada uno de los modelos aplicado al conjunto de datos respectivo del horno línea 1 y línea 2, tiene una tendencia a aumentar proporcionalmente. Junto a esto se evidencia que la arquitectura con peor comportamiento ante tiempos de predicción más largos son aquellas basadas en celdas LSTM, lo cual se ve solucionado con la inclusión de mecanismos de atención para este tipo de arquitecturas, pues los modelos con atención como se describió en la teoría y como se evidencia en los resultados presentaron el mejor desempeño. Sin embargo, la mejora evidenciada con la inclusión de estos mecanismos en modelos LSTM no es igual para la inclusión de estos en los modelos GRU, pues se observa que estos modelos de por sí presentan una buena respuesta ante tiempos de predicción más grandes, lo cual era de esperar al ser la evolución de los modelos LSTM que buscaban minimizar un poco esa problemática.

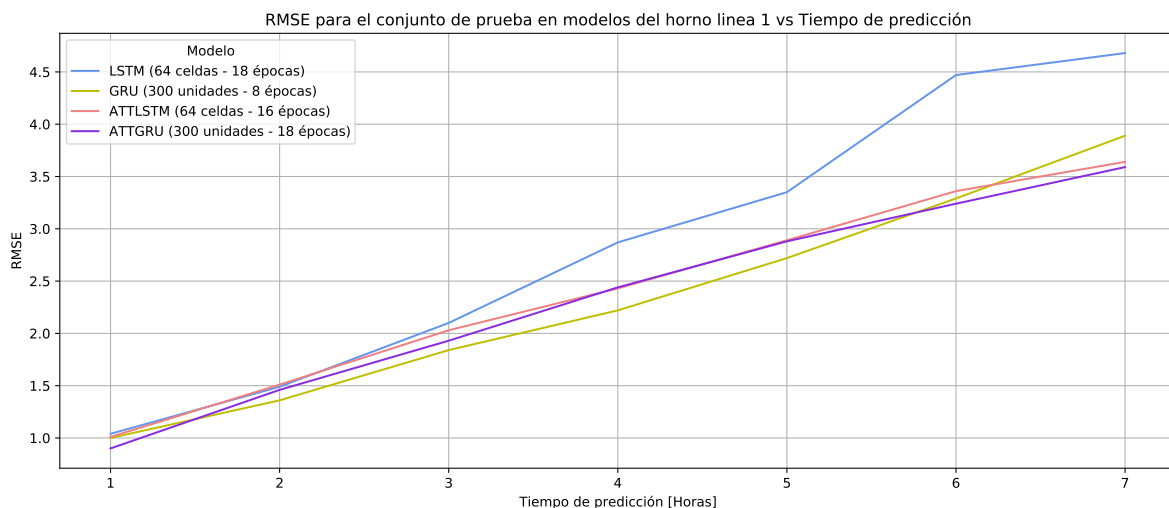


Figura 6-2.: Influencia del tiempo de predicción en el RMSE para el conjunto de prueba en modelos de predicción del horno línea 1.

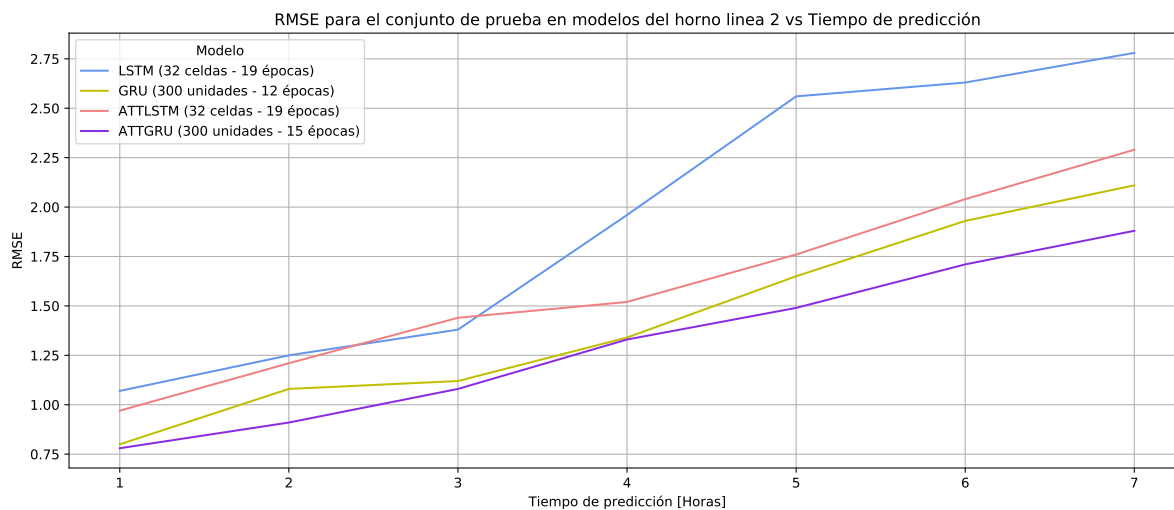


Figura 6-3.: Influencia del tiempo de predicción en el RMSE para el conjunto de prueba en modelos de predicción del horno línea 2.

Por último, en la figura 6-4 se observa el comportamiento de las predicción realizadas por los modelos GRU con y sin atención para el horno línea 1, donde para un tiempo de predicción de una hora se obtienen predicciones fieles y precisas a los valores reales, pero a medida que aumenta este tiempo empiezan a aparecer dos fenómenos en las predicciones, el primero de ellos corresponde a un desfase de tiempo, pues como se sabe el modelo tiene que ver cierta cantidad de datos iguales a los que va a predecir para poder realizar una predicción en el tiempo definido, de esta manera a mayor tiempo de predicción mayor será el desfase existente entre las predicciones y el valor real de las mismas. El segundo fenómeno corresponde a la

pérdida de calidad de las predicciones, pues a medida que aumenta el tiempo las predicciones realizadas sin tener en cuenta el desfase empeoran y se evidencian la aparición de más oscilaciones (Ruido) en los datos, siendo peor en este caso para los modelos GRU sin atención. De esto se concluye que el tiempo elegido de tres horas es el adecuado para balancear entre la calidad de las predicciones realizadas por todos los modelos y la ventana de tiempo para realizar acciones por parte de los operarios en planta.

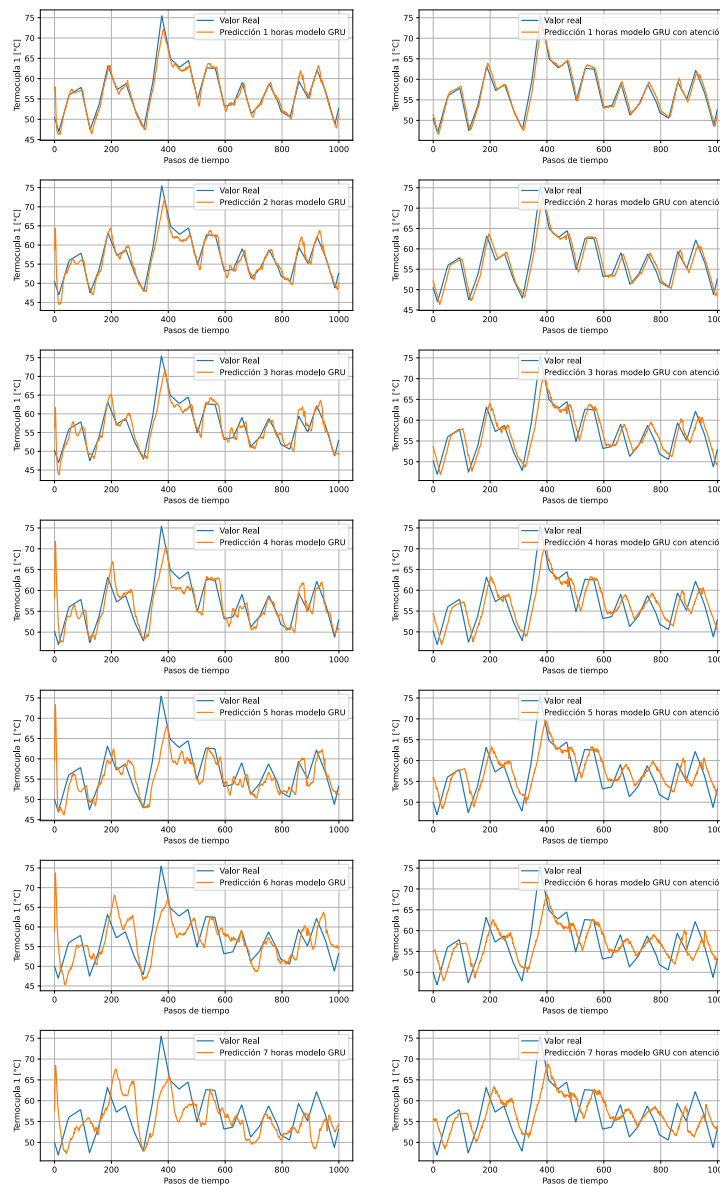


Figura 6-4.: Comportamiento de las predicciones para el conjunto de prueba en modelos de predicción GRU y GRU con atención del horno línea 1 a partir del tiempo de predicción.

6.2. Cantidad de variables en la predicción

La arquitectura de las redes neuronales en las capas de entrada y salida dependen en gran medida del tamaño del conjunto de datos con el que se va a trabajar, es por esto que el aumento en el número de variables de entrada y de la misma manera de variables a predecir tendrá efectos en las predicciones realizadas por las arquitecturas de los modelos diseñados. Para esto se decidió aumentar el número de termocuplas a predecir de a 16 termocuplas de manera progresiva hasta llegar a 64 termocuplas siguiendo el patrón radial como se observa en la figura 6-5, donde para cada par de paneles en verde se mantuvo la selección de 4 termocuplas en los niveles A, B, C y D.

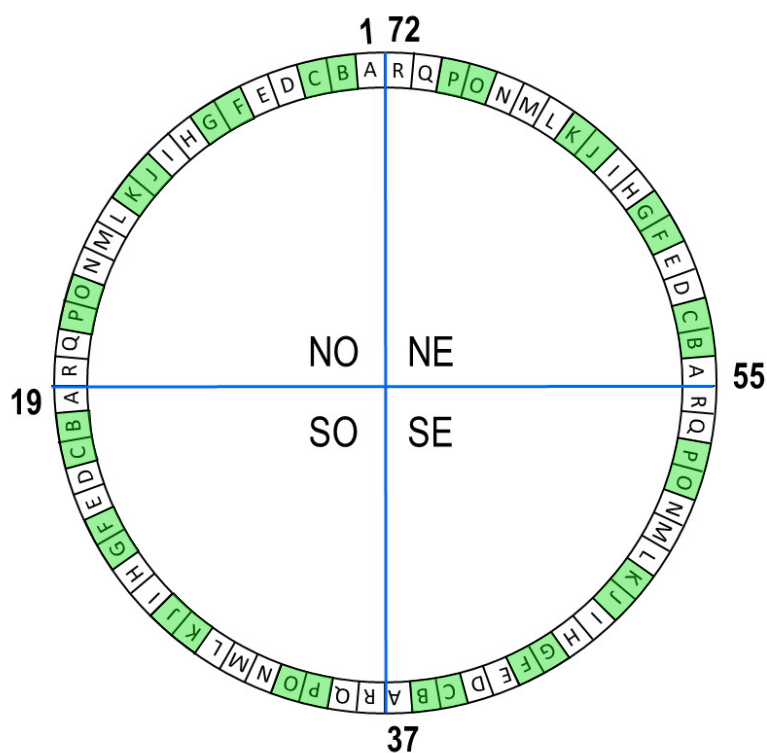


Figura 6-5.: Patrón de selección de paneles para el aumento de termocuplas a predecir.

Como se observa en la tabla 6-2, los cuatro modelos desarrollados para cada uno de los hornos fueron sometidos al aumento del número de termocuplas a predecir, donde se evidencia que en el primer aumento a 32 termocuplas los ocho modelos presentan un empeoramiento en el RMSE obtenido, tendencia que se mantiene hasta el aumento a 64 termocuplas. Sin embargo, a pesar del aumento de termocuplas sin haber realizado un ajuste de los hiperparámetros de los modelos debido a los nuevos datos el RMSE obtenido en los conjuntos de prueba sigue siendo aceptable, mostrando así el gran desempeño que tienen cada una de las arquitecturas desarrolladas ante sistemas más complejos con cantidades de datos a trabajar aún más

grandes.

Modelos Línea 1	Numero de termocuplas a predecir			
	16	32	48	64
LSTM	2.1	2.52	2.65	2.48
GRU	1.84	2.35	2.31	2.33
ATT + LSTM	2.03	2.35	2.74	2.68
ATT + GRU	1.93	2.41	2.53	2.44
Modelos Línea 2	Numero de termocuplas a predecir			
	16	32	48	64
LSTM	1.38	1.92	1.84	1.66
GRU	1.12	1.76	1.49	1.53
ATT + LSTM	1.44	1.95	1.72	1.76
ATT + GRU	1.08	1.82	1.62	1.54

Tabla 6-2.: Comportamiento del RMSE para los modelos de predicción horno línea 1 y 2 ante el aumento de termocuplas a predecir.

6.3. Validación cruzada

El rendimiento de los modelos desarrollados para los hornos línea 1 y línea 2 se evalúa a partir de su capacidad para seguir realizando predicciones adecuadas a medida que los datos nuevos entrantes se alejan de los datos de entrenamiento, pues en términos prácticos no es lo mismo predecir datos del año 2019 con modelos entrenados a partir de datos de 2016, pues es una ventana de tiempo lo suficientemente grande y los sistemas sufren variaciones y desgastes a lo largo del tiempo, lo cual nunca fue visto por el modelo conduciendo a tener malas predicciones. Es por esto que se empleó el método de la validación cruzada, el cual es un método estadístico que sirve para estudiar el rendimiento en el tiempo de un modelo de predicción de series de tiempo. En las siguientes subsecciones se llevaron a cabo dos métodos de validación cruzada para los modelos GRU y GRU con atención del horno línea 1 debido a la gran cantidad de registros existentes que su conjunto de datos posee.

6.3.1. Ventana móvil de tiempo

Las predicciones realizadas por un modelo de aprendizaje profundo dependen totalmente del conjunto empleado para el entrenamiento y la forma en la que fue realizado el entrenamiento, es por esto que al obtener resultados adecuados surge la incógnita de si esto se debe al desempeño que posee el tipo de arquitectura RNN usada o se debe por otra parte al conjunto de datos que se utilizó, pues para dos ventanas de tiempo diferentes en una misma serie de tiempo se pueden obtener comportamientos totalmente diferentes. A partir de esto se decidió

emplear la totalidad de los datos del horno línea 1 correspondientes a los 175,297 registros de 5 años realizando 11 particiones como se observa en la figura **6-6**, siguiendo lo establecido para la partición del conjunto de entrenamiento en aproximadamente un año (36,000 registros) y el conjunto de prueba en 42 días (4,000 registros) empezando el 30 de septiembre de 2014, se estableció una ventana móvil de tiempo de 140 días (13,500 registros), que luego de 11 iteraciones, finaliza el 30 de septiembre de 2019.

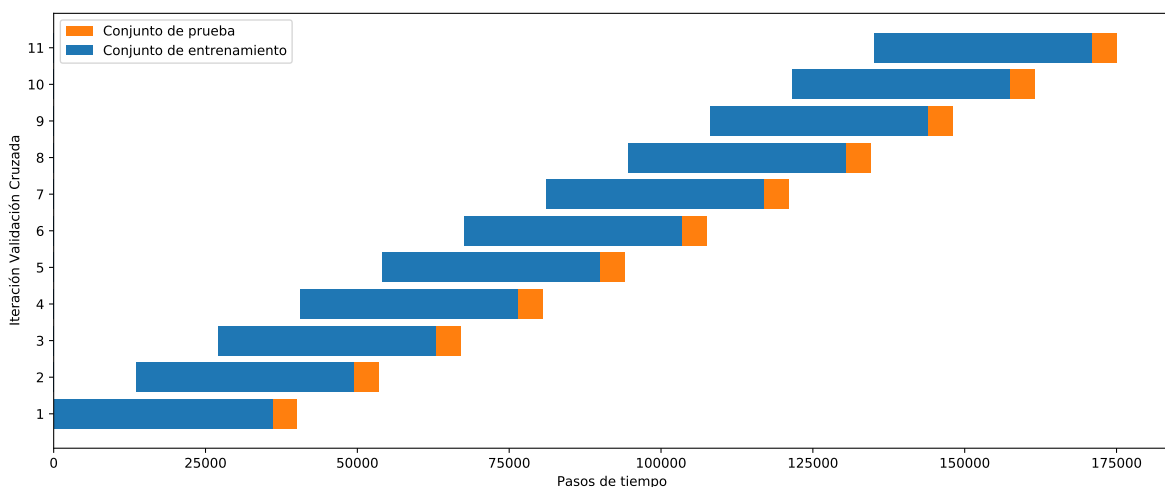


Figura 6-6.: Selección de datos para validación cruzada con desplazamiento en conjuntos de entrenamiento y prueba.

En la tabla **6-3** se detalla el RMSE obtenido para los conjuntos de prueba de cada una de las iteraciones realizadas en los dos modelos del horno línea 1, donde para la arquitectura GRU se tiene un valor mínimo de 0,88 y un máximo de 2,22, mientras que para el modelo GRU con atención se tiene un valor mínimo de 1,13 y máximo de 2,21, con lo cual se puede establecer que la incógnita mencionada anteriormente sobre la obtención de buenos resultados, está relacionada tanto por el buen desempeño que presentan las arquitecturas RNN para la predicción de series de tiempo como por el conjunto de datos de entrenamiento. Sin embargo, en cada una de las 11 iteraciones se obtuvo un valor RMSE adecuado para las predicciones.

Iteración	MODELO	
	GRU	ATT + GRU
1	0,88	1,13
2	1,15	1,48
3	0,92	1,20
4	1,14	1,48
5	2,07	2,19
6	1,07	1,29
7	1,54	2,04
8	1,02	1,27
9	1,77	2,00
10	2,22	2,21
11	1,84	1,93

Tabla 6-3.: RMSE obtenido por iteración de ventana móvil para el conjunto de prueba.

6.3.2. Ventana móvil de tiempo - Conjunto de prueba estático

En el anterior modelo de validación cruzada si bien se analizaron predicciones con diferentes conjuntos de datos el conjunto de prueba siempre empezaba exactamente donde el conjunto de entrenamiento terminaba, por lo que observar el desempeño a largo plazo del modelo es imposible, para esto se volvieron a realizar las 11 iteraciones a partir de una ventana de tiempo móvil de 140 días pero únicamente para el conjunto de entrenamiento, manteniendo estático el conjunto de prueba en los últimos 42 días del total de registros como se observa en la figura 6-7.

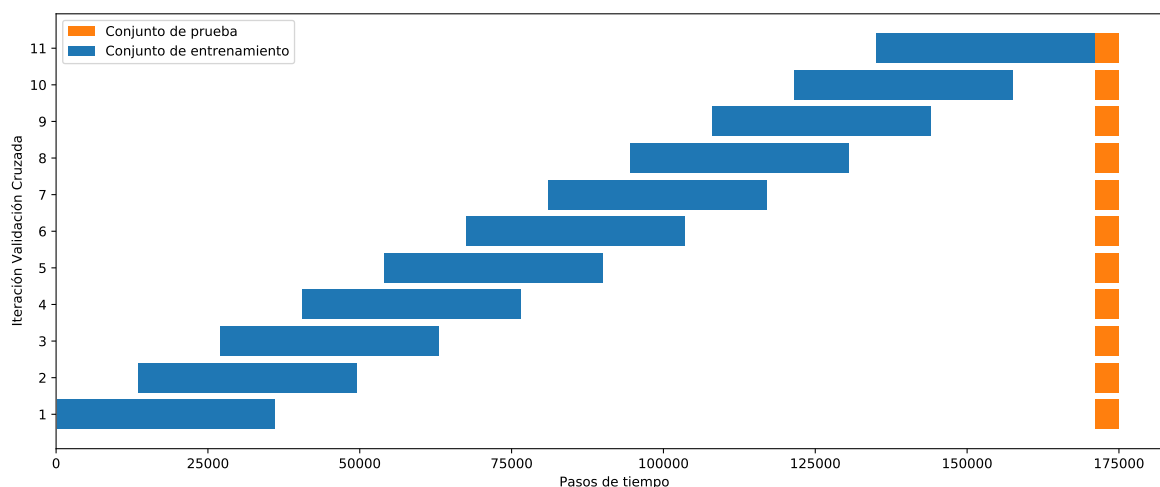


Figura 6-7.: Selección de datos para validación cruzada con desplazamiento del conjunto de entrenamiento manteniendo el conjunto de prueba estático.

En la tabla **6-4** se detalla el RMSE obtenido para el conjunto de prueba estático a partir de cada una de las iteraciones realizadas en los dos modelos del horno línea 1, las primeras iteraciones que son las de mayor separación entre el conjunto de entrenamiento y el conjunto de prueba presentan un RMSE mucho más alto en sus predicciones, el cual va disminuyendo a medida que estos dos conjuntos se acercan, empezando a tener valores RMSE adecuados a partir de la sexta iteración que corresponde a una separación de dos años entre los conjuntos, es decir que el desempeño del modelo se mantendrá a lo largo de dos años sin requerir un reentrenamiento siempre y cuando las condiciones en el horno no cambien drásticamente.

Iteración	MODELO	
	GRU	ATT + GRU
1	5,29	5,11
2	5,46	5,46
3	5,44	5,53
4	3,71	3,34
5	3,11	2,75
6	2,4	2,48
7	2,23	2,34
8	2,72	2,63
9	2,57	2,17
10	2,31	2,14
11	1,84	1,93

Tabla 6-4.: RMSE obtenido por iteración de ventana móvil para el conjunto de prueba estático.

6.4. Distribución promedio de la función de costo

Se sabe que los valores RMSE observados en todos los resultados y análisis anteriores a esta subsección recopilan la información promedio de los valores RMSE obtenidos en la predicción para cada una de las termocuplas empleadas, lo que significa que al momento de estudiar los modelos y las predicciones realizadas con un único valor RMSE en promedio se obtendrá un predicción con un error cercano a este valor, sin embargo, al ser un promedio no significa que las 16 termocuplas por separado tengan un RMSE cercano entre sí, pues como se puede evidenciar en la figura **6-8** para el modelo GRU del horno línea 1 existe una gran varianza entre el RMSE de las predicciones de cada una de las termocuplas, mientras las termocuplas 4, 8 y 12 presentan una distribución pequeña de error, termocuplas como la 13 y 14 tienen una distribución mucho mayor respecto al RMSE promedio.

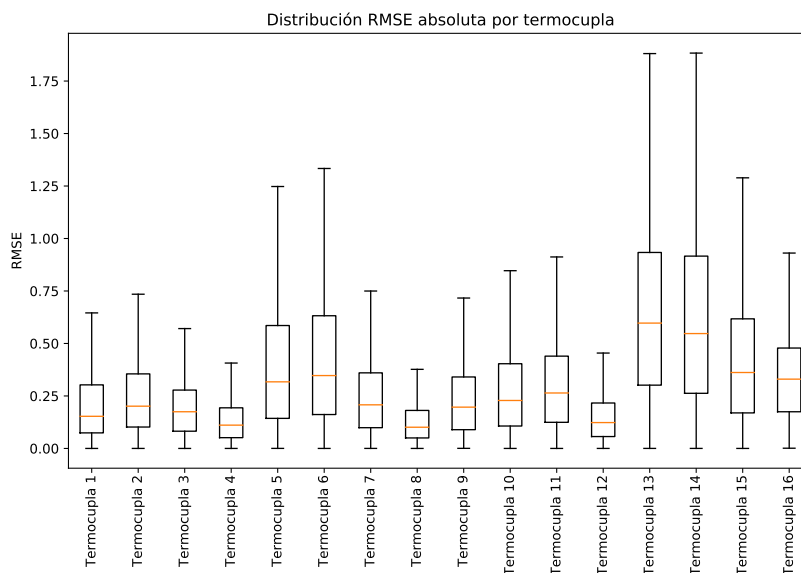


Figura 6-8.: Distribución RMSE en las predicciones para cada una de las 16 termocuplas con el modelo GRU en el horno línea 1.

En los modelos de atención se observa como este fenómeno se atenúa en cierta medida, en la figura 6-9 el RMSE promedio por termocupla presenta menos varianza para el modelo GRU con atención del horno línea 2.

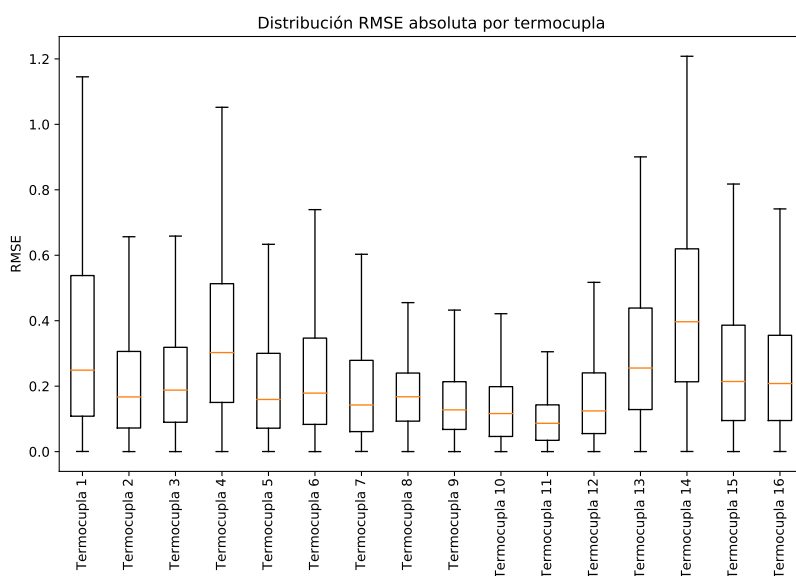


Figura 6-9.: Distribución RMSE en las predicciones para cada una de las 16 termocuplas con el modelo GRU con atención en el horno línea 2.

7. Conclusiones

- Múltiples resultados fueron encontrados en el desarrollo del presente trabajo. Como se mostró a lo largo del documento fue posible el desarrollo de una metodología de predicción de temperaturas para una aplicación práctica tal como la predicción de temperaturas en dos hornos de arco eléctrico. En concreto, el desarrollo de esta tesis mostró adicionalmente que es posible la colaboración de la Universidad para la solución de un problema en el sector industrial.
- Aunque la metodología fue aplicada a la predicción de variables de temperatura en un horno de arco eléctrico en el contexto de la empresa CMSA, esta puede ser extendida a la predicción de variables en otros contextos considerando que el proceso de limpieza de datos debe contextualizarse a la naturaleza y condiciones de operación de las variables a predecir.
- El adecuado conocimiento de las variables de funcionamiento de los hornos línea 1 y línea 2 junto con la implementación de una metodología para la identificación y limpieza de datos en el conjunto de información brindado por Cerro Matoso S.A permitió la obtención de un conjunto de datos para ser usado en el entrenamiento de modelos de aprendizaje profundo.
- El desarrollo y entrenamiento de diversas arquitecturas de redes neuronales recurrentes a partir del ajuste de hiperparámetros permitió establecer las ventajas y desventajas de cada una de estas arquitecturas para su aplicabilidad en series de tiempo, logrando así la obtención de cuatro modelos para la estimación de las variables de temperatura de pared media para cada uno de los hornos de Cerro Matoso S.A.
- De los cuatro modelos desarrollados para el horno línea 1 y línea 2, el modelo obtenido con mayor desempeño para cada caso tenía los siguientes parámetros:
 - **Horno línea 1**
 - Arquitectura RNN:** 300 unidades GRU con capa densa a la salida.
 - Épocas de entrenamiento:** 8 Épocas.
 - RMSE conjunto de prueba:** 1.84 [°C]
 - Tiempo de predicción:** 3 Horas.

- **Horno línea 2**

Arquitectura RNN: 300 unidades GRU con atención y capa densa a la salida.

Épocas de entrenamiento: 15 Épocas.

RMSE conjunto de prueba: 1.08 [°C]

Tiempo de predicción: 3 Horas.

- La validación de los modelos desarrollados permitió conocer el desempeño que tienen los mismos ante variaciones en el tiempo de predicción, factor importante si se desea tener previsiones del comportamiento de las temperaturas en los hornos para periodos de tiempo más largos, también ante el aumento de variables de temperatura a predecir, lo cual deja una ventana abierta a la expansión del proyecto para la predicción de temperatura en todas las termocuplas de los hornos en CMSA y de esta manera tener una aproximación más fiel de los comportamientos reales de estas variables.
- El proceso de validación cruzada permitió establecer el tiempo de desempeño de aproximadamente dos años durante el cual el modelo aplicado al horno línea 1 tendrá predicciones adecuadas y confiables antes de que sea necesario volver a entrenarlo. Este proceso no se pudo llevar a cabo en el horno línea 2 debido a la escasez de datos que este presentaba, sin embargo, cuando el conjunto de datos adquiera más registros de las variables empleadas será posible llevar a cabo también un estudio de validación cruzada para poder establecer el tiempo de desempeño de los modelos aplicados a este horno.
- Como trabajo futuro, un estudio de validación a desarrollar podría ser sobre el comportamiento de las predicciones realizados por los modelos ante casos de falla en las termocuplas o la no disponibilidad de algunas de las variables de proceso seleccionadas para la entrada de datos. Esto permitiría mostrar la robustez de los modelos desarrollados.

Bibliografía

- [1] D. Tibaduiza et al., “Structural Health Monitoring System for Furnace Refractory Wall Thickness Measurements at Cerro Matoso SA”, *Lecture Notes in Civil Engineering*, pp. 414-423, 2021. DOI: 10.1007/978-3-030-64594-6_41.
- [2] F. Pozo et al., “Structural health monitoring and condition monitoring applications: sensing, distributed communication and processing”, *International Journal of distributed sensor networks*, vol 16, no. 9, p 1-3, 2020. DOI: 10.1177/1550147720963270.
- [3] J. Birat, “A futures study analysis of the technological evolution of the EAF by 2010”, *Revue de Métallurgie*, vol. 97, no. 11, pp. 1347-1363, 2000. DOI: 10.1051/metal:2000114.
- [4] “Redes neuronales profundas - Tipos y Características - Código Fuente”, *Código Fuente*, 2021. [Online]. Disponible: <https://www.codigofuente.org/redes-neuronales-profundas-tipos-caracteristicas/>. [Acceso: 17- Jul- 2021].
- [5] “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation”, *Medium*, 2021. [Online]. Disponible: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>. [Acceso: 17- Jul- 2021].
- [6] “Major Mines & Projects | Cerro Matoso Mine”, *Miningdataonline.com*, 2021. [Online]. Disponible: <https://miningdataonline.com/property/336/Cerro-Matoso-Mine.aspx>. [Acceso: 25- Nov- 2021].
- [7] Janzen, J.; Gerritsen, T.; Voermann, N.; Veloza, E.R.; Delgado, R.C. Integrated Furnace Controls: Implementation on a Covered-Arc (Shielded Arc) Furnace at Cerro Matoso. In *Proceedings of the 10th International Ferroalloys Congress*, Cape Town, South Africa, 1–4 Feb. 2004; pp. 659–669.
- [8] R. Garcia-Segura, J. Vázquez Castillo, F. Martell-Chavez, O. Longoria-Gandara, and J. Ortegón Aguilar, “Electric Arc Furnace Modeling with Artificial Neural Networks and Arc Length with Variable Voltage Gradient,” *Energies*, vol. 10, no. 9, p. 1424, Sep. 2017.
- [9] C. Chen, Y. Liu, M. Kumar, and J. Qin, “Energy Consumption Modelling Using Deep Learning Technique — A Case Study of EAF”, *Procedia CIRP*, vol. 72, pp. 1063-1068, 2018. DOI: 10.1016/j.procir.2018.03.095.

-
- [10] S. Ismaeel, A. Miri, A. Sadeghian, and D. Chourishi, “An Extreme Learning Machine (ELM) Predictor for Electric Arc Furnaces’ v-i Characteristics,” 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, 2015, pp. 329-334, DOI: 10.1109/CSCloud.2015.94.
- [11] J. Mesa Fernández, V. Cabal, V. Montequin and J. Balsera, “Online estimation of electric arc furnace tap temperature by using fuzzy neural networks”, *Engineering Applications of Artificial Intelligence*, vol. 21, no. 7, pp. 1001-1012, 2008. DOI: 10.1016/j.engappai.2007.11.008.
- [12] M. Kordos, M. Blachnik and T. Wiczorek, “Temperature Prediction in Electric Arc Furnace with Neural Network Tree”, *Lecture Notes in Computer Science*, pp. 71-78, 2011. DOI: 10.1007/978-3-642-21738-8_10.
- [13] J. Camacho et al., “A Data Cleaning Approach for a Structural Health Monitoring System in a 75 MW Electric Arc Ferronickel Furnace”, *Proceedings of 7th International Electronic Conference on Sensors and Applications*, 2020. DOI: 10.3390/ecsa-7-08245.
- [14] J. Leon-Medina et al., “Deep Learning for the Prediction of Temperature Time Series in the Lining of an Electric Arc Furnace for Structural Health Monitoring at Cerro Matoso S.A. (CMSA)”, *Proceedings of 7th International Electronic Conference on Sensors and Applications*, 2020. DOI: 10.3390/ecsa-7-08246.
- [15] J. Leon-Medina et al., “Temperature Prediction Using Multivariate Time Series Deep Learning in the Lining of an Electric Arc Furnace for Ferronickel Production”, *Sensors*, vol. 21, no. 20, p. 6894, 2021. DOI: 10.3390/s21206894.
- [16] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, “Multivariate Temporal Convolutional Network: A Deep Neural Networks Approach for Multivariate Time Series Forecasting”, *Electronics*, vol. 8, no. 8, p. 876, 2019. DOI: 10.3390/electronics8080876.
- [17] S. Shih, F. Sun, and H. Lee, “Temporal pattern attention for multivariate time series forecasting”, *Machine Learning*, vol. 108, no. 8-9, pp. 1421-1441, 2019. DOI: 10.1007/s10994-019-05815-0.
- [18] S. Du, T. Li, Y. Yang and S. Horng, “Multivariate time series forecasting via attention-based encoder–decoder framework”, *Neurocomputing*, vol. 388, pp. 269-279, 2020. DOI: 10.1016/j.neucom.2019.12.118.
- [19] S. Huang, D. Wang, X. Wu, and A. Tang, “DSANet: Dual Self-Attention Network for Multivariate Time Series Forecasting”, *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019. DOI: 10.1145/3357384.3358132.

- [20] CMSA, PR032018OP - Manual del Sistema de Control Estructural del Horno Eléctrico 412-FC-01, 02 ed., 2017.
- [21] D. F. Godoy-Rojas et al., “Attention-Based Deep Recurrent Neural Network to Forecast the Temperature Behavior of an Electric Arc Furnace Side-Wall,” *Sensors*, vol. 22, no. 4, p. 1418, Feb. 2022, doi: 10.3390/s22041418.
- [22] American Petroleum Institute (API), “API RP 551 - Process Measurement”, 2da edición, pp. 30-36, Febrero 2016, Disponible: <https://standards.globalspec.com/std/9988220/API%20RP>
- [23] “Specification for temperature-electromotive force (EMF) tables for standardized thermocouples” ASTM DOI: 10.1520/e0230_e0230m-17.
- [24] W. W. S. Wei, “Time Series analysis”, Oxford Handbooks Online, pp. 458–485, 2013.
- [25] J. D. Hamilton, “Time Series analysis”, Princeton, NJ: Princeton University Press, 2020.
- [26] P. P. Shinde and S. Shah, “A Review of Machine Learning and Deep Learning Applications”, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697857.
- [27] LeCun, Y., Bengio, Y. and Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>.
- [28] L. Zhang, J. Tan, D. Han, and H. Zhu, “From machine learning to Deep Learning: Progress in Machine Intelligence for Rational Drug Discovery”, *Drug Discovery Today*, vol. 22, no. 11, pp. 1680–1685, 2017.
- [29] C. M. Bishop, “Neural networks and their applications”, *Review of Scientific Instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.
- [30] K. Suzuki, Ed., “Artificial Neural Networks - Architectures and Applications”, Jan. 2013, doi: 10.5772/3409.
- [31] C. Zanchettin and T. B. Ludermir, “A methodology to train and improve artificial neural networks weights and connections”, *The 2006 IEEE International Joint Conference on Neural Network Proceedings*.
- [32] Sibi, P., S. Allwyn Jones, and P. Siddarth., “Analysis of different activation functions using back propagation neural networks”, *Journal of theoretical and applied information technology* 47.3 (2013): 1264-1268.

-
- [33] A. D. Rasamoelina, F. Adjailia and P. Sinčák, “A Review of Activation Function for Artificial Neural Network”, 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany, Slovakia, 2020, pp. 281-286, doi: 10.1109/SAMI48414.2020.9108717.
- [34] Sharma, Sagar, Simone Sharma, and Anidhya Athaiya, “Activation functions in neural networks”, *towards data science* 6.12 (2017): 310-316.
- [35] Elliott, David L., “A better activation function for artificial neural networks”, 1993.
- [36] XU, Jingyi, et al., “A semantic loss function for deep learning with symbolic knowledge”, *International conference on machine learning*. PMLR, 2018. p. 5502-5511.
- [37] LEE, Tae-Hwy., “Loss functions in time series forecasting”, *International encyclopedia of the social sciences*, 2008, p. 495-502.
- [38] HODSON, Timothy O., “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not”, *Geoscientific Model Development*, 2022, vol. 15, no 14, p. 5481-5487.
- [39] C. Alippi, “Weight update in back-propagation neural networks: The role of activation functions”, 1991 IEEE International Joint Conference on Neural Networks, 1991.
- [40] D. Svozil, V. Kvasnicka, and Pospichal Jirí, “Introduction to multi-layer feed-forward neural networks”, *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43–62, 1997.
- [41] Ruder, S., “An overview of gradient descent optimization algorithms”, arXiv:1609.04747.
- [42] S.-ichi Amari, “Backpropagation and stochastic gradient descent method,” *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.
- [43] Smith, Leslie N., “A disciplined approach to neural network hyper-parameters: Part 1—learning rate, batch size, momentum, and weight decay” arXiv:1803.09820, 2018.
- [44] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, and M. Tuba, “Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics”, *Algorithms*, vol. 13, no. 3, p. 67, 2020.
- [45] M. Kuan and K. Hornik, “Convergence of learning algorithms with constant learning rates”, *IEEE Transactions on Neural Networks*, vol. 2, no. 5, pp. 484-489, Sept. 1991, doi: 10.1109/72.134285.






-
- [46] D. R. Wilson and T. R. Martinez, “The need for small learning rates on large problems” IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222), Washington, DC, USA, 2001, pp. 115-119 vol.1, doi: 10.1109/IJCNN.2001.939002.
- [47] Y. Bengio, “Gradient-based optimization of hyperparameters”, *Neural Computation*, vol. 12, no. 8, pp. 1889–1900, 2000.
- [48] “Recurrent neural networks architectures”, *Wiley Series in Adaptive and Learning Systems for Signal Processing, Communications and Control*, pp. 69–89.
- [49] A. L. Caterini and D. E. Chang, “Recurrent neural networks”, *Deep Neural Networks in a Mathematical Framework*, pp. 59–79, 2018.
- [50] Sutskever, Ilya, “Training recurrent neural networks” Toronto, ON, Canada: University of Toronto, 2013.
- [51] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and long short-term memory (LSTM) network”, *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.
- [52] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory”, In *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [53] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu and H. Zhang, “Deep Learning with Long Short-Term Memory for Time Series Prediction”, In *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114-119, June 2019, doi: 10.1109/MCOM.2019.1800155.
- [54] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng, “Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model”, *Journal of Petroleum Science and Engineering*, vol. 186, p. 106682, 2020.
- [55] R. Dey and F. M. Salem, ”Gate-variants of Gated Recurrent Unit (GRU) neural networks”, 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1597-1600, doi: 10.1109/MWSCAS.2017.8053243.
- [56] Y. Wang, W. Liao, and Y. Chang, “Gated recurrent unit network-based short-term photovoltaic forecasting”, *Energies*, vol. 11, no. 8, p. 2163, 2018.
- [57] H. Lin, A. Gharehbaghi, Q. Zhang, S. S. Band, H. T. Pai, K.-W. Chau, and A. Mosavi, “Time Series-based groundwater level forecasting using gated recurrent unit deep neural networks”, *Engineering Applications of Computational Fluid Mechanics*, vol. 16, no. 1, pp. 1655–1672, 2022.

-
- [58] S. H. Park, B. Kim, C. M. Kang, C. C. Chung and J. W. Choi, “Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture”, 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 2018, pp. 1672-1678, doi: 10.1109/IVS.2018.8500658.
- [59] R. Laubscher, “Time-series forecasting of coal-fired power plant reheater metal temperatures using encoder-decoder recurrent neural networks”, *Energy*, vol. 189, p. 116187, 2019.
- [60] C. Olah and S. Carter, “Attention and augmented recurrent neural networks”, *Distill*, 08-Sep-2016. [Online]. Disponible: <https://distill.pub/2016/augmented-rnns/>. [Acceso: 15-Sep-2022].
- [61] Z. Niu, G. Zhong, and H. Yu, “A review on the attention mechanism of Deep Learning”, *Neurocomputing*, vol. 452, pp. 48–62, 2021.
- [62] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction”, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.
- [63] IT-0003-A28-C3-V1-18.11.2019 - Informe preliminar con análisis estadístico de datos y correlaciones posibles.
- [64] IT-O3O4-C15C34.2.3-V1-17.06.2020 - Informe técnico de caracterización e identificación de variables del horno línea 1 FC01.
- [65] IT-O3O4.C38.2.1-V1-04.10.2021 - Informe técnico de caracterización e identificación de variables del horno línea 2 FC150.

A. Anexo: Publicación revista indexada

Article

Attention-Based Deep Recurrent Neural Network to Forecast the Temperature Behavior of an Electric Arc Furnace Side-Wall

Diego F. Godoy-Rojas ¹, Jerisson X. Leon-Medina ^{2,3}, Bernardo Rueda ⁴, Whilmar Vargas ⁴, Juan Romero ⁴, Cesar Pedraza ⁵, Francesc Pozo ^{2,6,*} and Diego A. Tibaduiza ¹

- ¹ Departamento de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Colombia, Cra 45 No. 26-85, Bogotá 111321, Colombia; dfgodoyr@unal.edu.co (D.F.G.-R.); dtibaduizab@unal.edu.co (D.A.T.)
 - ² Control, Modeling, Identification and Applications (CoDALab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Campus Diagonal-Besòs (CDB), Universitat Politècnica de Catalunya (UPC), Eduard Maristany 16, 08019 Barcelona, Spain; jerisson.xavier.leon@upc.edu
 - ³ Departamento de Ingeniería Mecánica y Mecatrónica, Universidad Nacional de Colombia, Cra 45 No. 26-85, Bogotá 111321, Colombia
 - ⁴ South32-Cerro Matoso S.A., Km 22 Highway SO Montelibano, Córdoba 234001, Colombia; Bernardo.S.Rueda@south32.net (B.R.); whilmar.p.vargas@south32.net (W.V.); JuanAlonso.Romero@south32.net (J.R.)
 - ⁵ Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Cra 45 No. 26-85, Bogotá 111321, Colombia; capedrazab@unal.edu.co
 - ⁶ Institute of Mathematics (IMTech), Universitat Politècnica de Catalunya (UPC), Pau Gargallo 14, 08028 Barcelona, Spain
- * Correspondence: francesc.pozo@upc.edu



Citation: Godoy-Rojas, D.F.; Leon-Medina, J.X.; Rueda, B.; Vargas, W.; Romero, J.; Pedraza, C.; Pozo, F.; Tibaduiza, D.A. Attention-Based Deep Recurrent Neural Network to Forecast the Temperature Behavior of an Electric Arc Furnace Side-Wall. *Sensors* **2022**, *22*, 1418. <https://doi.org/10.3390/s22041418>

Academic Editors: Adam Glowacz, Jose A Antonino-Daviu, Wahyu Caesarendra and Marcin Woźniak

Received: 22 December 2021

Accepted: 10 February 2022

Published: 12 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Structural health monitoring (SHM) in an electric arc furnace is performed in several ways. It depends on the kind of element or variable to monitor. For instance, the lining of these furnaces is made of refractory materials that can be worn out over time. Therefore, monitoring the temperatures on the walls and the cooling elements of the furnace is essential for correct structural monitoring. In this work, a multivariate time series temperature prediction was performed through a deep learning approach. To take advantage of data from the last 5 years while not neglecting the initial parts of the sequence in the oldest years, an attention mechanism was used to model time series forecasting using deep learning. The attention mechanism was built on the foundation of the encoder–decoder approach in neural networks. Thus, with the use of an attention mechanism, the long-term dependency of the temperature predictions in a furnace was improved. A warm-up period in the training process of the neural network was implemented. The results of the attention-based mechanism were compared with the use of recurrent neural network architectures to deal with time series data, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The results of the Average Root Mean Square Error (ARMSE) obtained with the attention-based mechanism were the lowest. Finally, a variable importance study was performed to identify the best variables to train the model.

Keywords: structural health monitoring; temperature forecasting; recurrent neural network; attention; GRU; LSTM; electric arc furnace

1. Introduction

The control and monitoring of industrial processes require special attention because of their complexity, which is the result of the sub-processes and the multiple variables involved that need to be considered to know the current state of the general process. Regarding systems that make use of structures, the use of structural health monitoring (SHM) systems allows the proper monitoring of variables in the decision-making process, allowing better knowledge of the behavior of the structure and providing tools for maintaining tasks [1]. In an SHM system, some elements are required, such as the use of sensors permanently

installed in the structure, a data acquisition system for sensing/actuating over the structure, a signal conditioning step, the development of statistical models and the possibility of a decision-making process [2]. This last element can be developed by computational tools in an autonomous way or by the analysis obtained from the statistical models. The literature includes multiple examples of developed monitoring systems and applications in different kinds of structures, such as those used in aircraft [3–5], buildings [6,7], bridges [8,9] and furnaces [10,11], among others.

Concerning furnace monitoring as used in smelting processes, the number of variables and the influence on the process is highly significant. As an example, in the case of the ferronickel production industry, this process can be performed in an electric arc furnace (EAF) [12] and the structural health monitoring (SHM) of the system requires the monitoring of several parts. The refractory hearth lining of an EAF is a crucial part to improve the campaign life of the furnace [13]. The lining monitoring variables comprise temperature, heat fluxes, water quality, remaining thickness refractory, sidewall erosion and protective layer formation, among others [14]. However, the development of temperature lining prediction models in an EAF is still an open research field because of the reduced number of works in this area [15,16].

Recently, the use of deep learning models has spread due to the data availability and their success rates in classification and regression tasks in minerals processing [17]. In addition, the success of deep learning models is based on their capacity for extracting features, improving the data-driven models in terms of accuracy and efficiency; moreover, the big data coming from a sensor network allow large-scale training based on deep learning models [18].

Cerro Matoso S.A. (CMSA) is one of the world's major producers of ferronickel and it is operated by South32. This is an open-cut mine operation in Northern Colombia, /textcolorbluewith nearly 40 years of operation in the region. More details about the process developed by CMSA can be found directly on its web page <https://www.cerromatoso.com.co/> (accessed on 10 January 2022). The complex process of produce ferronickel in the EAF of (CMSA) involves a number of variables. In this work, the lining temperature in an EAF is predicted using a multivariate time series deep learning model. The developed model is able to handle the multiple input variables as well as predict multiple thermocouple output variables. The time series approach was selected in order to process variable-length sequences of inputs. This kind of model can use recurrent neural networks (RNN) to handle the temporal dynamic behavior of the data. The long-term dependency of the temperature predictions in the EAF was compared using, first, a Long Short-Term Memory (LSTM) unit and, second, a Gated Recurrent Unit (GRU) approach [19]. These kinds of cells are used in contrast with traditional RNN due to the capacity to handle the vanishing and exploding long-term gradient problems [20]. The temporal information has been incorporated into deep learning models using different encoder architectures, such as convolutional neural networks (CNN), RNN and attention-based models [21]. Attention models allow us to identify relevant parts in the input sequence data to improve the prediction behavior of the deep learning model in the target time series [22–24].

The time series forecasting deep learning model is developed with data from a 75 MW shielded arc smelting furnace of CMSA [25]. This furnace is instrumented with a large set of thermocouples radially distributed in the lining furnace. The cooling system in the furnace uses plate and waffle coolers [26]. There are four levels of plate coolers radially distributed in 72 panels in the furnace.

The novelty of this work lies in the development of a time series forecasting deep learning model using an attention-based mechanism. This model takes into account as input variables different operation variables in the furnace, such as power, current, voltage, electrode position, amount of input material and chemistry composition. As output variables, 68 thermocouples radially distributed in the furnace lining were satisfactorily predicted at different forecast times in a range from 1 h to 6 h in the future.

The remainder of the paper is structured as follows. Section 2 includes the theoretical background, where all methods are described, followed by the dataset for validation in Section 3; then, the multivariate time series temperature forecasting model is described in Section 4. Then, the results and discussion are shown in Section 5, and, finally, the conclusions are included in the last section.

2. Theoretical Background

Here, the main concepts used in the development of the attention-based deep recurrent neural network model are described. For more information, the reader is referred to each provided reference.

2.1. Electric Arc Furnace

The ferronickel production inside CMSA has several stages, including the mining and material homogenization phase, in which the material extracted from the mine is divided into smaller parts. Then, the phase of drying and storing the material is executed. Subsequently, the semi-dried material enters a rotatory kiln calciner; the material at the exit of this stage is called calcine, which is supplied to the electric arc furnace through different upper tubes distributed in three central, semi-central and lateral zones. The smelting stage is carried out within the EAF, which is detailed below. After the material is melted, it is ejected from the furnace employing two different runners, one for the ferronickel and the other for the slag. The next phase in the process is the refining and granulation phase of the material; finally, there is the finished product handling phase, where the material is packed and taken to commercialization. Figure 1 shows a picture of the building where the two furnaces are located. The dimensions of each furnace are 22 m in diameter and 7 m in height.

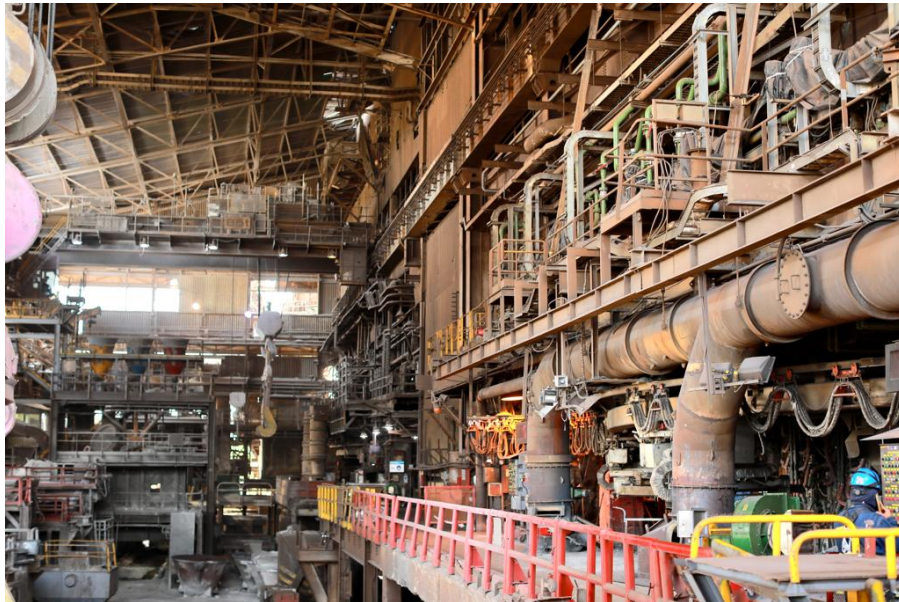
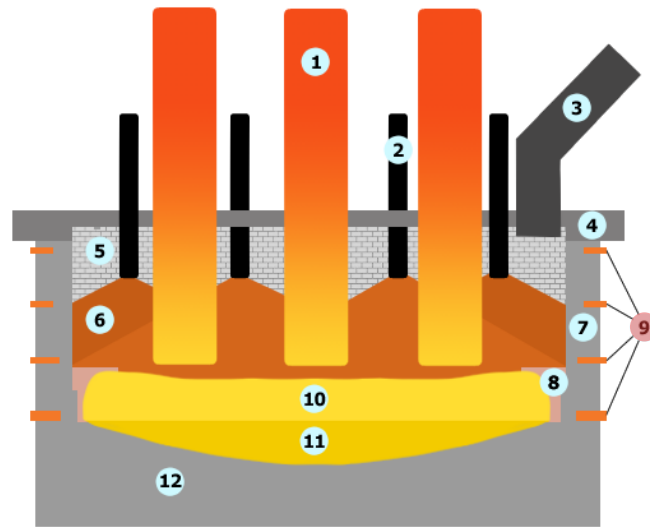


Figure 1. Panoramic of the CMSA plant.

The main stage of ferronickel production is smelting. It is performed in the EAF. Figure 2 shows an inside view of the EAF, detailing its parts: (1) electrodes, (2) feeding tubes, (3) exhaust chimney, (4) top roof, (5) back-side wall, (6) input calcine, (7) sidewall, (8) waffle coolers, (9) plate coolers, (10) smelted ferronickel, (11) slag and (12) bottom hearth furnace lining. This study is focused on the temperature monitoring and forecasting of the side-wall; in particular, this temperature is measured by a thermocouples' sensor network located at the plate coolers of the side-wall.



- | | | |
|--------------------|--------------------------|--------------------|
| 1. Electrodes | 5. Gas chamber | 9. Thermocouples |
| 2. Load feed pipe | 6. Charge | 10. Slag |
| 3. Chimney | 7. Refractory brick wall | 11. Ferronickel |
| 4. Refractory roof | 8. Slag crust | 12. Furnace bottom |

Figure 2. Electric arc furnace components description.

2.2. Multivariate Time Series Forecasting

The multivariate time series forecasting process seeks the behavior of a set of output variables at a specific future time. Several methods have been developed to model the relationships between fluctuating variables in time series data. These methods can be divided into classical and machine learning methods. Among the classical methods are Autoregressive Integrated Moving Average (ARIMA), Vector Autoregression (VAR) and Vector Autoregression Moving-Average (VARMA) [27]. In contrast to classical methods, machine learning methods are effective in more complex time series prediction problems with multiple input variables, complex nonlinear relationships and missing data [28]. Some machine learning algorithms used for regression tasks have been used for time series forecasting; among them are Support Vector Regression, Random Forest, Extreme Gradient Boosting and Artificial Neural Networks [21]. Recently, deep learning advances have emerged as a satisfactory method to perform time series forecasting. The recurrent neural networks and their variants, such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), have addressed the problem of vanishing gradient and long-term dependencies, achieving remarkable behaviors [19].

2.3. Encoder–Decoder

For each time step in the LSTM and GRU models, each input corresponds to one of the outputs. In some cases, the objective is to predict an output given a different-length input, without correspondence; the models developed for these cases are known as seq-to-seq models. A typical model has two parts, an encoder and a decoder, with two different networks combined into one network; this network can take an input sequence and generate the next most probable sequence as the output. First, the encoder traverses the input at each time step to encode the complete sequence in a vector called the context vector; this vector acts as the last hidden state of the encoder and as the first hidden state for the decoder. This will contain information about all the input elements, which will help in the realization of the predictions [29].

2.4. Attention Mechanism

One of the frontiers in deep learning is attention mechanisms, which represent an evolution of encoder–decoder models, which were developed to improve the performance of long input sequences. In attention mechanisms, the decoder can selectively access the encoded information and uses a new concept for the context vector $c(t)$, which is now calculated at each time step of the decoder, from the previous hidden state and all the hidden states of the encoder [29]. Trainable weights will be assigned to these states and produce different degrees of importance to all the elements in the input sequence. Special attention is paid to the most significant inputs—hence, they are named attention mechanisms [30]. The construction of the context vector starts from the combination of each time step j of the encoder with each time step t of the decoder. This expression is called the alignment score, and it is calculated as follows:

$$\text{score}(j, t) = V_a \tanh(U_a s(t-1) + W_a h(j)) \quad (1)$$

The terms V_a , W_a and U_a correspond to the trainable weights mentioned above, where V_a defines the function to calculate the alignment score, W_a are associated with the hidden states of the encoder and U_a with the hidden states of the decoder. The score must be normalized for each time step t ; therefore, the SoftMax function is used together with the time steps j , and we obtain the attention weights $\alpha(j, t)$, defined as follows:

$$\alpha(j, t) = \frac{e^{\text{score}(j,t)}}{\sum_{j=1}^M e^{\text{score}(j,t)}} \quad (2)$$

This weight can capture the importance of the input at time step j to adequately decode the output at time step t . Finally, the context vector is found from the weighted sum of the relationship between all the encoder hidden values and attention weights:

$$c(t) = \sum_{j=1}^M T \alpha(j, t) h(j) \quad (3)$$

The context vector allows more attention to the relevant inputs in the electric arc furnace variables. The term $c(t)$ passed through the decoder and the probability for the next possible output is calculated. This operation applies to all time steps at the input. Then, the current hidden state $s(t)$ is calculated, taking as input the context vector $c(t)$, the previous hidden state $s(t-1)$ and the output $\hat{y}(t-1)$ from the previous time step:

$$s(t) = f(s(t-1), \hat{y}(t-1), c(t)) \quad (4)$$

Therefore, using this attention mechanism, the model can find the correlations between the different parts of the input sequence to the corresponding parts of the output sequence. For each time step, the decoder output is calculated by applying the “SoftMax” function to the hidden state [31].

2.5. Root Mean Squared Error (RMSE)

The performance of the multivariate time series forecasting deep learning model is calculated using the Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{B} \sum_{i=1}^B (\hat{y}_i - y_i)^2} \quad (5)$$

where B is the number of data points in the time series to be estimated, y_i is the actual value of the time series, and \hat{y}_i is the estimated value at the time i by the prediction model.

3. Dataset for Validation

Data used to train and validate the attention-based deep RNN model were obtained from a thermocouple sensor network located at the side-wall of an EAF in CMSA. Photography of the EAF side-wall is shown in Figure 3. The EAF side-wall is composed of 72 radially distributed panels. Figure 3 details a portion of the side-wall of 1 panel. The illustrated hoses carry water, which is used to cool the refractory walls of the EAF through the plate coolers (4 for each panel) and the waffle cooler (1 for each panel).

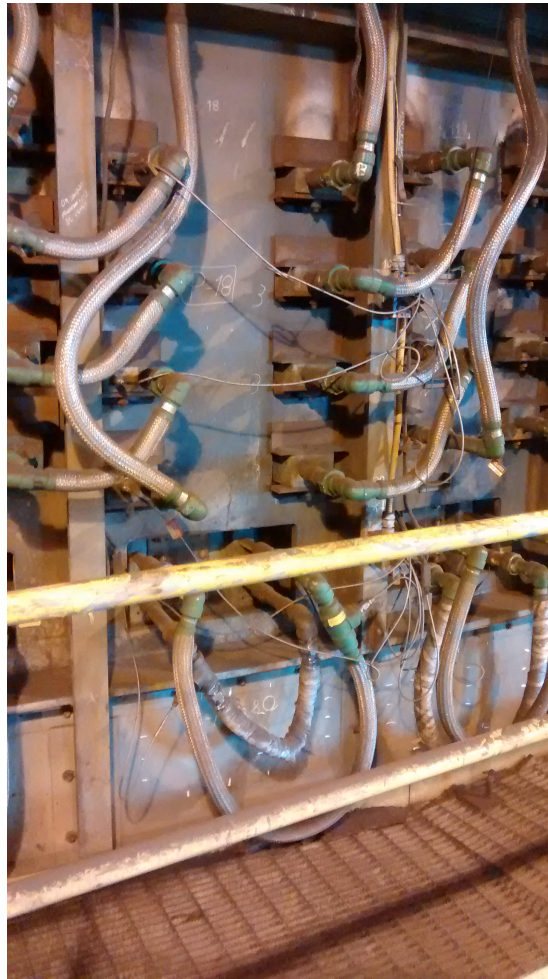


Figure 3. Photography of CMSA furnace outside wall with its coolers.

The dataset used for model training and validation is composed of data recorded during 5 years, with 177,312 instances and 49 attributes, from an EAF located in Cerro Matoso, South 32 company. Data were collected every 15 min during a period of 1847 days, from September 9th of 2016 to September 30th of 2021. The input variables in the model were related to electrode current, voltage, arc, power, calcine feed, the chemical composition of the calcine, relative electrode position and 16 thermocouples. These 16 thermocouples were also taken as output variables to predict. In particular, 4 panels radially distributed 90 degrees in each quadrant of the furnace were selected to study the behavior of their plate cooler thermocouples. Each of the selected 4 panels had 4 plate coolers; thus, a total of 16 plate coolers were analyzed. The behavior of the time series of some of these variables in a time window that allows the trend to be seen can be observed in Figure 4.

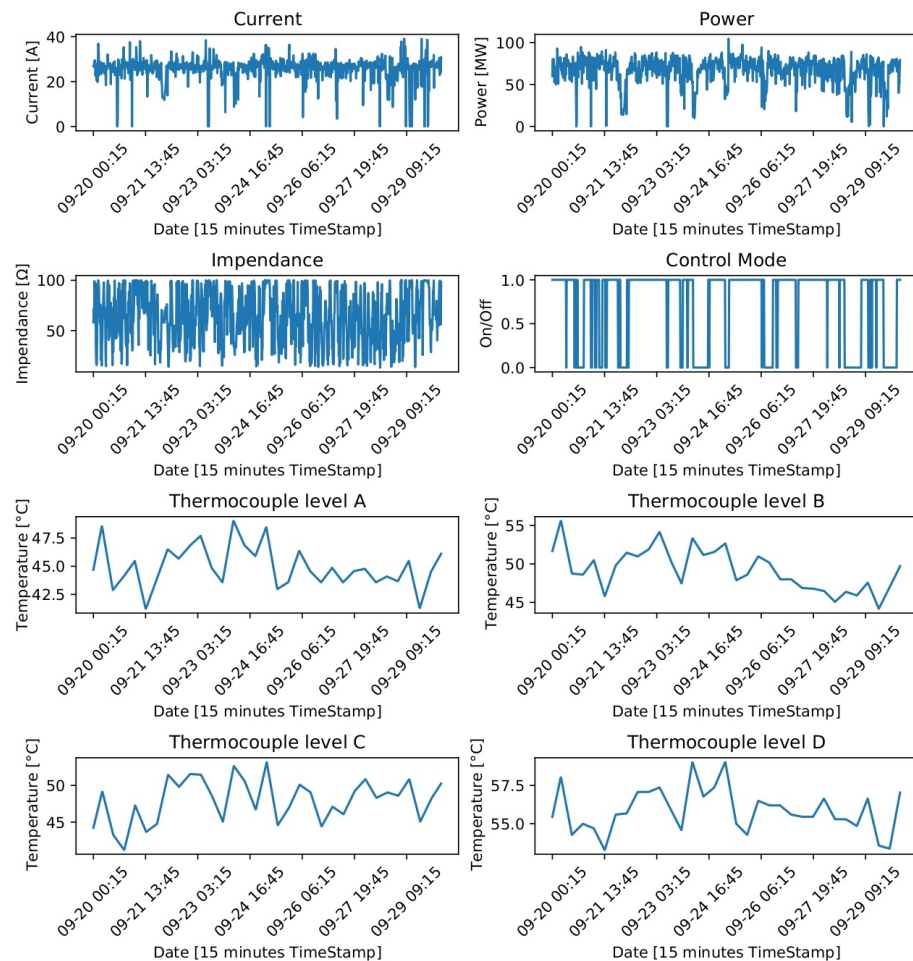


Figure 4. Time series plot of some input and output variables in the dataset.

Several data preprocessing steps were performed to detect abnormal behavior in the used variables. These data preprocessing steps are listed below [32]:

- Remove duplicates;
- Treat empty and null values;
- Treat unique values;
- Encode strings;
- Remove negative temperatures;
- Eliminate variables with high variance;
- Remove variables with zero variance.

After verifying the data preprocessing, it was concluded that the 49 variables used to train and test the models did not present abnormal behaviors.

4. Multivariate Time Series Temperature Forecasting Model

The development of the multivariate time series temperature forecasting model comprised several stages. It started with the definition of the initial set of data already preprocessed, where the input variables for the models were selected as well as the variables to be predicted, the data were normalized so that the neural networks could work with them, the forecast time was defined and, in this way, the batch set generator was created for model training, as well as the data sequences for validation. The neural network models GRU and LSTM are designed to be incorporated with the attention mechanisms, and the RMSE loss function is defined with a warm-up period of 50 steps, which was not considered for

the calculation of the evaluation metric, so we proceeded to train the model, validate it and generate the predictions to be able to compare them with the real values and make conclusions; this process is summarized in Figure 5.

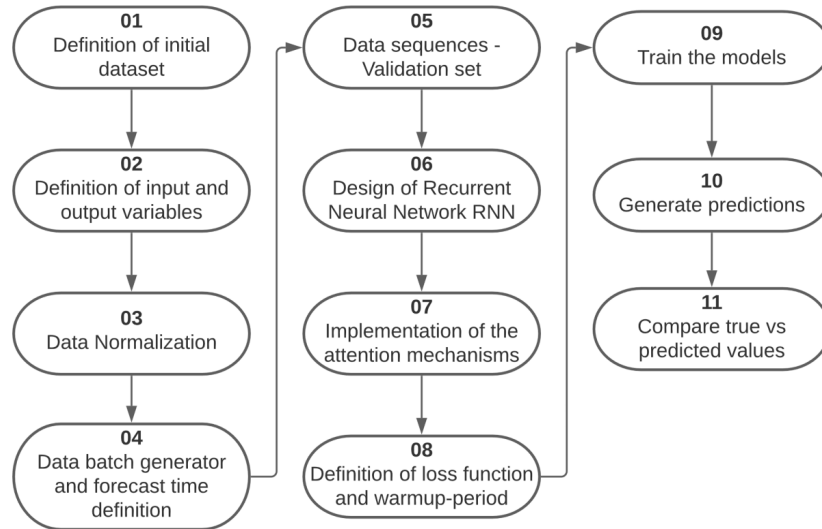


Figure 5. Step by step development of the temperature prediction model.

The different layers that compose the multivariate time series deep learning attention models are summarized in Figure 6. Details of the shape and the number of parameters and connections of the layers in the multivariate time series deep learning attention model are noted. For the the gradient descent method, we used Adam optimization; this is a stochastic gradient descent (SGD) method that is based on adaptive estimation.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, None, 49)]	0	[]
gru (GRU)	(None, None, 100)	45300	['input_1[0][0]']
last_hidden_state (Lambda)	(None, 100)	0	['gru[0][0]']
attention_score_vec (Dense)	(None, None, 100)	10000	['gru[0][0]']
attention_score (Dot)	(None, None)	0	['last_hidden_state[0][0]', 'attention_score_vec[0][0]']
attention_weight (Activation)	(None, None)	0	['attention_score[0][0]']
context_vector (Dot)	(None, 100)	0	['gru[0][0]', 'attention_weight[0][0]']
attention_output (Concatenate)	(None, 200)	0	['context_vector[0][0]', 'last_hidden_state[0][0]']
attention_vector (Dense)	(None, 64)	12800	['attention_output[0][0]']
dense (Dense)	(None, 16)	1040	['attention_vector[0][0]']

 Total params: 69,140
 Trainable params: 69,140
 Non-trainable params: 0

Figure 6. Details of the shape and layer connections in the attention-based multivariate time series forecasting model.

5. Results and Discussion

Four deep neural network configurations corresponding to a GRU model and an LSTM model, with and without attention mechanisms, were designed, trained and tested, using 49 input variables to predict the 16 output variables corresponding to the thermocouple temperature. The Average Root Mean Square Error (RMSE) of these 16 output variables was used as a performance metric for each of the models.

5.1. Influence of Changing the Prediction Time

To determine the models' behavior relating to the time interval under which they performed the prediction, the test was performed in a time window of 1 to 6 h predicting in the future for each model, increased by 1 h, as shown in Table 1 and Figure 7. From Table 1, it is evident that the Average RMSE values in the test set are larger than the train values. This is caused by the large amount of data belonging to the train set (90%) compared to the data from the test set (10%).

Table 1. Average RMSE results of the train and test sets for the four different deep learning models in 6 different times.

MODEL	SET	AVERAGE RMSE TRAIN—TEST SETS [°C]					
		1 H	2 H	3 H	4 H	5 H	6 H
LSTM	Train	1.30	1.58	1.92	2.31	2.63	2.92
	Test	1.65	1.96	2.33	2.96	3.39	3.75
GRU	Train	1.17	1.48	1.80	2.13	2.53	2.76
	Test	1.43	1.72	2.09	2.42	2.98	3.24
LSTM + Attention	Train	1.20	1.63	2.09	2.45	2.82	3.11
	Test	1.31	1.77	2.30	2.74	2.99	3.45
GRU + Attention	Train	1.05	1.48	2.00	2.38	2.74	3.06
	Test	1.15	1.62	2.13	2.54	3.01	3.47

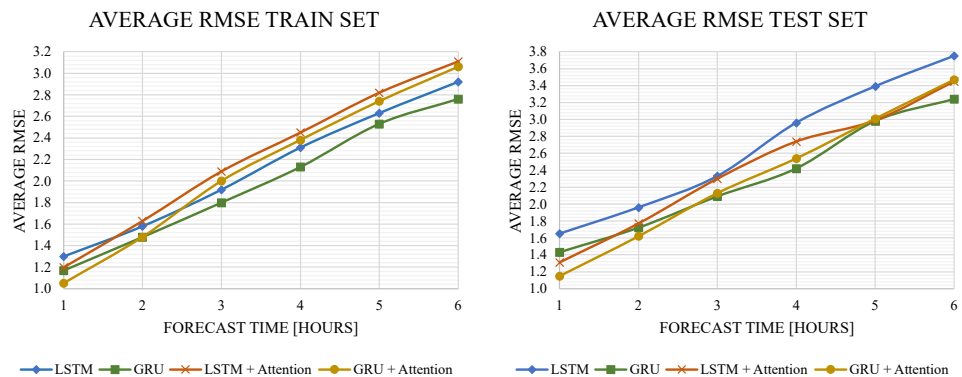


Figure 7. Average RMSE behavior over the forecast time increases.

Figure 7 shows that the models with attention mechanisms had higher performance during a shorter prediction time. As the prediction time increased from 1 to 6 h, the models without attention mechanisms outperformed the other models. The GRU model obtained the best results with attention mechanisms for short times and without attention mechanisms for long times; for the short times, the longer input sequence in the GRU and LSTM networks resulted in worse prediction accuracy of the output sequence because it focused on all input variables equally. An attention mechanism can be used to alleviate this problem by focusing on more relevant input variables, since, as already described above, attention mechanisms can adaptively assign a different weight to each input sequence to

automatically choose the most relevant features of the time series. Therefore, the model can effectively capture the long-term dependence on the time series.

As a result of the models evaluated in a 1 h forecast with and without attention, the predicted and true behaviors for a single thermocouple were compared, as shown in Figure 8. It is evident that the GRU model including attention (orange line) obtained a better representation of the true (green line) behavior. In contrast, the only GRU model (blue line) presented a more curly and distant behavior from the true data.

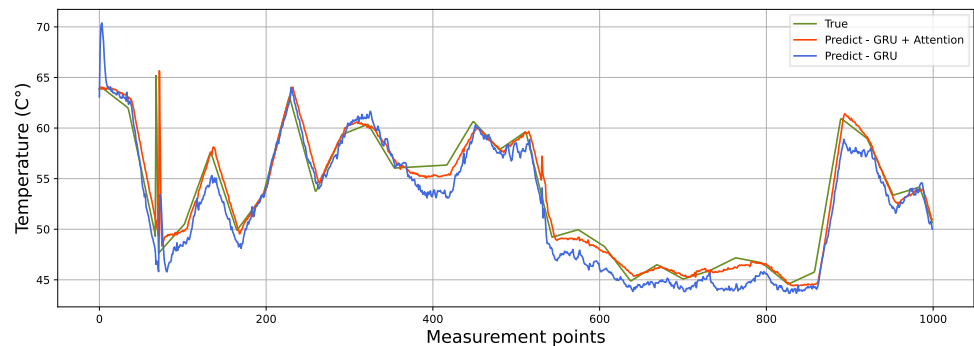


Figure 8. Predictive versus true behavior of the GRU and GRU + attention models in the test set in one of the output thermocouples.

Additionally, in Table 2, the individual comparison of the RMSE error of each one of the thermocouples for each model used is presented. Here, it can be observed how some thermocouples have small prediction errors and others very large, which is averaged and leads to obtaining the Average RMSE of the total forecast.

Table 2. RMSE results in the 16 thermocouples for the train and test sets versus the different deep learning models for a 6 h forecast.

Thermocouple (T)	GRU		GRU + ATT		LSTM		LSTM + ATT	
	Train	Test	Train	Test	Train	Test	Train	Test
T1	2.11	3.62	2.07	3.38	1.69	3.77	2.13	3.65
T2	2.68	3.12	2.76	3.18	2.18	3.51	2.91	3.15
T3	2.05	2.40	2.31	2.37	1.56	2.61	2.32	2.56
T4	1.39	1.12	1.46	1.15	1.22	1.29	1.47	1.15
T5	3.22	3.87	3.59	4.13	2.88	4.35	3.59	4.16
T6	3.69	4.60	3.76	4.23	3.08	5.02	3.80	4.22
T7	2.33	3.46	2.65	2.94	1.83	3.82	2.66	3.03
T8	1.58	1.23	1.62	1.25	1.38	1.48	1.66	1.24
T9	2.21	2.38	2.43	2.44	1.98	2.44	2.46	2.48
T10	2.49	2.26	2.65	2.46	2.10	2.52	2.65	2.74
T11	2.29	2.70	2.57	2.89	1.81	3.42	2.62	2.78
T12	1.56	1.40	1.65	1.33	1.38	1.62	1.69	1.41
T13	7.31	8.22	7.26	8.03	6.54	5.83	7.44	8.05
T14	6.63	6.65	6.71	6.50	6.35	7.32	6.78	6.55
T15	4.16	5.14	4.25	4.43	3.72	5.66	4.29	4.47
T16	2.86	3.29	2.72	2.63	2.57	2.99	2.83	3.11

5.2. Parameter Exploration

To evaluate the influence of some parameters in the Average RMSE results, an exploration procedure was executed. The changing of three different parameters was evaluated. These parameters were the optimizer, the number of cells in the GRU and LSTM models and finally the number of training epochs in the GRU model.

5.2.1. Changing of Optimizer

Four different optimizers were evaluated in order to compare their influence on the Average RMSE obtained by the GRU model. The four compared optimizers were RMSprop, Adam, Adamax and Nadam. As shown in Table 3, the best optimizer was Adam, obtaining an RMSE value in the train set of 3.01.

Table 3. Average RMSE for GRU model with attention mechanisms against optimizer variance.

SET	Optimizer			
	RMSprop	Adam	Adamax	Nadam
Train RMSE	3.08	3.01	3.13	3.03
Test RMSE	3.62	3.32	3.37	3.44

5.2.2. Change of GRU Cell Number

The variation in the number of GRU cells was studied by changing this number from 50 to 175, as shown in Table 4. The results indicate that, as the number of cells increases, the RMSE in the training set decreases, which does not mean that it is a good result because, in this way, the model is over-fitting with the training data, which means that, as the number of cells increases, the RMSE of the test set becomes worse because the model is so adjusted to the training data that when new and unknown data arrive in the model input, it is more difficult to make an adequate prediction.

Table 4. Average RMSE for GRU model with attention mechanisms against GRU unit variance.

SET	GRU UNITS					
	50	75	100	125	150	175
Train RMSE	3.11	3.08	3.06	3.05	3.04	3.03
Test RMSE	3.41	3.26	3.29	3.31	3.35	3.39

5.2.3. Change of LSTM Cell Number

Three different cell numbers were compared in the LSTM model. In this case, they were 32, 64 and 96, as shown in Table 5. The results show favorable behavior for the variation of 64 cells; as in the GRU model, more units does not lead to better results, due again to phenomena such as over-fitting.

Table 5. Average RMSE for LSTM model with attention mechanisms against LSTM unit variance.

SET	LSTM UNITS		
	32	64	96
Train RMSE	3.25	3.10	2.80
Test RMSE	3.44	3.41	3.81

5.2.4. Changing of the Loss Behavior of the GRU Model through the Epochs

Figure 9 shows the loss behavior as the number of training epochs increases. From the results, it is evident that the first seven epochs are crucial in the decrease in loss, while, from epoch 7 onwards, the decrease in loss is scarce.

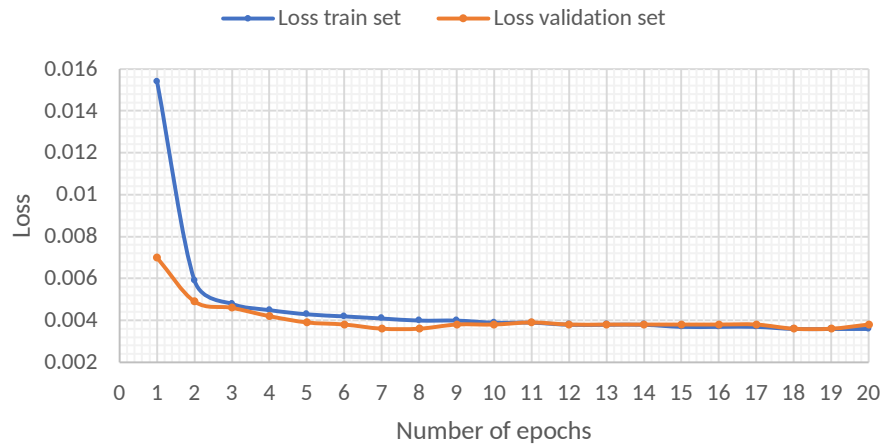


Figure 9. Evaluation of the loss behavior of the GRU model through the epochs in the train set and test set.

5.3. Time Series Cross-Validation

Different cross-validation procedures have been developed to evaluate the behavior of a time series forecasting model [33]. In this study, three different approaches to perform time series cross-validation were used. These three approaches were (a) 7-fold moving origin, (b) Blocking Time Series Split and (c) Blocking Time Series Split with a static test set. Below, these three approaches are described and discussed.

5.3.1. Seven-Fold Moving Origin Time Series Split Cross-Validation

The first approach for the time series cross-validation was the 7-fold moving origin. This procedure involves cumulative training data from October 1 of 2020 to September 1 of 2021. Figure 10 illustrates the results at the top and details the data division in the bottom section. Seven different folds were evaluated; the first is the least in the training set, and as the folds increase, the size of the training data also increases. The size of the test set remains constant in each fold. The size of this test set is 4000 data instances.

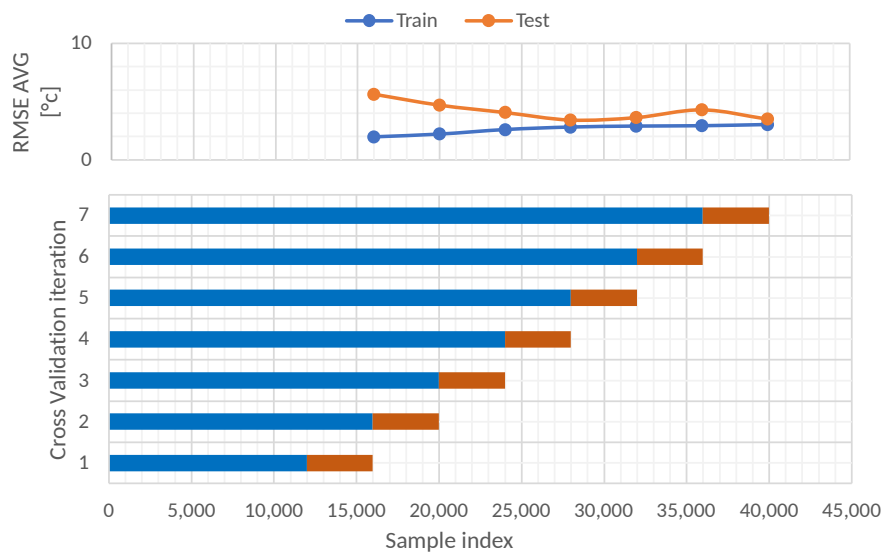


Figure 10. Time series split cross-validation data partitions.

Table 6 shows the RMSE results for the train and test sets in each fold. As can be seen, the train RMSE increases as the number of folds increases. The opposite happens with

the behavior of the RMSE test set; this indicates that it is better to train with numerous data because, with more data, the model can learn different scenarios that are presented in the furnace.

Table 6. Average RMSE for training and test sets at each iteration with time series split.

# Iteration	Train RMSE	Test RMSE
1	1.99	5.63
2	2.22	4.70
3	2.60	4.07
4	2.81	3.43
5	2.89	3.64
6	2.92	4.30
7	3.02	3.51

5.3.2. Blocking Time Series Split

A second study using a blocking time series split cross-validation was performed. This validation approach consists of setting a fixed size of the train and test sets and moving across the entire dataset in several folds. In this case, 11 folds were used, and the train test had a size of 36,000 instances, whereas the test size had a size of 4000 instances. The shift between each fold was 140 days. Figure 11 illustrates the 11 folds and every train set in blue and test set in orange. In total, 177312 instances of the dataset were used; these data began on September 9th of 2016 and ended on September 30th of 2021.

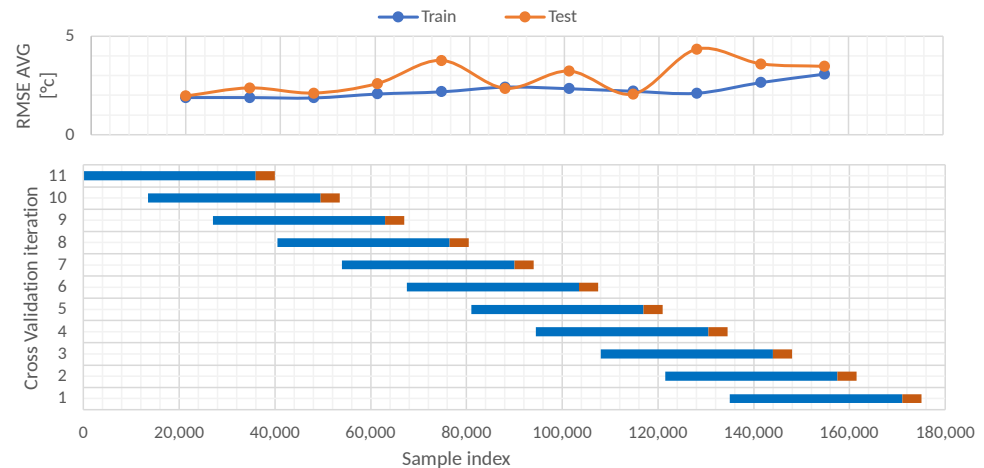


Figure 11. Blocking time series split cross-validation data partitions.

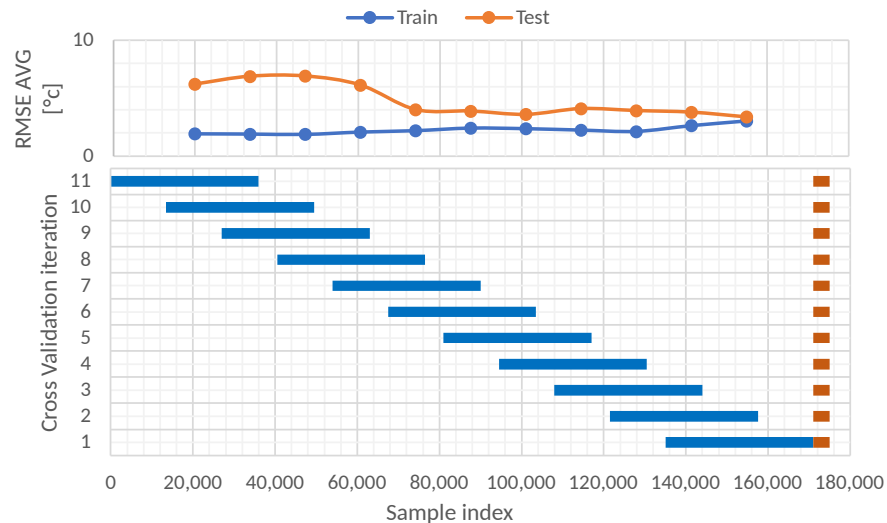
The results after performing the 11-fold blocking time series split cross-validation are shown in Table 7. From these results, it is evident that the best results of RMSE in the train set (1.89) and test set (1.96) were reached by the oldest fold—in this case, the 11th fold. Furthermore, a decreasing behavior of the RMSE through the folds is evident for the train set. In contrast, the behavior of the test set is oscillatory decreasing. Considering the 11 folds, the average RMSE was 2.24 for the train set and 2.89 for the test set.

Table 7. Average RMSE for training and test sets at each iteration with blocking time series split.

# Iteration	Train RMSE	Test RMSE
1	3.08	3.46
2	2.65	3.59
3	2.11	4.33
4	2.21	2.06
5	2.34	3.22
6	2.42	2.35
7	2.18	3.76
8	2.07	2.59
9	1.87	2.11
10	1.89	2.37
11	1.89	1.96
Average RMSE	2.24	2.89

5.3.3. Blocking Time Series Split with Static Test Set

The last study for the time series cross-validation model was the blocking time series split with the static test set. In this case, 11 folds were also evaluated, but the test set remained the same for every fold. This test set was created with the most recent 4000 instances. Different training sets were tested. The shift between each training fold was 140 days. The size of each training set was 36,000 instances. Figure 12 illustrates the blocking time series split with static test set approach.

**Figure 12.** Blocking time series split with static test set cross-validation data partitions.

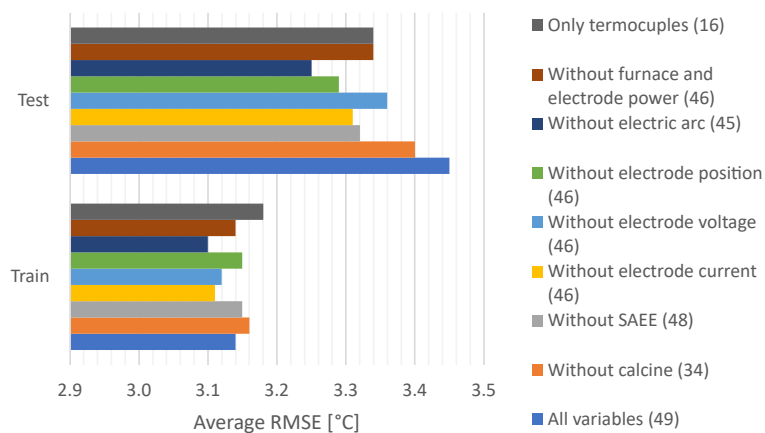
The RMSE results of the blocking time series split with static test set approach are shown in Table 8. The results indicate that it is preferable to perform training with recent data since the RMSE increases as the training data move away from the test data. The RMSE in the test set changes from 3.37 for the first fold to 6.21 in the 11th fold, which represents an increase of 84.27%. Therefore, it is advisable to train the model every certain period to avoid obvious increases in the RMSE.

Table 8. Average RMSE for training and test sets at each iteration, with blocking time series split with static test set.

# Iteration	Train RMSE	Test RMSEr
1	3.02	3.37
2	2.62	3.78
3	2.13	3.92
4	2.24	4.09
5	2.36	3.59
6	2.41	3.88
7	2.19	4.01
8	2.07	6.13
9	1.87	6.91
10	1.90	6.89
11	1.92	6.21
Average RMSE	2.24	4.79

5.4. Variable Importance Study

A study of the selected variables used to train and test the GRU model was performed to evaluate their influence on the RMSE values. This study was performed with the data of the 7th fold in the time series split cross-validation data partitions shown in Figure 10. Therefore, 36,000 instances composed the training set, whereas 4000 instances constituted the test set. Seven different scenarios were selected to train and test, removing the original number of variables as follows: (a) only with the 16 thermocouples to predict, (b) without furnace and electrode electric power, (c) without an electric arc, (d) without electrode position, (e) without electrode voltage, (f) without electrode current, (g) without the automatic control of electric power in the furnace (SAEE) mode, (h) without calcine chemistry and (i) using all 49 variables. The results of the RMSE comparison are shown in the bar chart of Figure 13. From the results in the bar chart, one can observe the difference between the RMSE results in the train and test sets, the latter being the one with the largest RMSE values. In particular, the worst results in the test set were obtained by the (i) all variables' configuration, causing the error to reach the highest value of 3.45 in the test set. Consequently, when removing different groups of variables, the RMSE value improved. The lowest RMSE value of the test set of 3.22 was reached when the group (c) without electric arc was removed. Thus, it is better to remove the group of variables (c) related to the electric arc to develop the GRU model.

**Figure 13.** Variable influence in the Average RMSE of the GRU model.

5.5. Study of Increasing the Number of Predicted Thermocouples to 76

Based on a request made by the CMSA engineering team in which they preferred to concentrate on the monitoring of the plate coolers in the lower row, the number of thermocouples to be monitored and predicted was increased. A study of the increase in the number of thermocouples was carried out, progressively evaluating their impact on the Average RMSE of the predictions.

The number of thermocouples was progressively increased from 16 until reaching 76 thermocouples. The RMSE values of the training set and the test set were measured for each increase; these values are shown in Table 9. It can be seen that the relationship between the increase in thermocouples and the Average RMSE of the predictions is directly proportional since, in Figure 14, the increasing trend of this evaluation metric can be observed due to the increase in the number of variables to be predicted.

The increase that occurs in the Average RMSE is small compared with the increase in the number of thermocouples. There was an increase of 4.75 times in the number of thermocouples, while the RMSE in the training set remained approximately constant because there was more information that the model could use to obtain better relationships between the variables. On the other hand, for the test set, the RMSE increased only 1.1 times; this is because the number of variables and data that the model must predict is greater, but it is still a good prediction result.

From the results depicted in Figure 14, it can be seen that the attention GRU model improved the RMSE value when 24 thermocouples were predicted. Therefore, for a few thermocouples, the attention GRU model is better, whereas, for numerous thermocouples, it is recommended to use the GRU model without the attention mechanism.

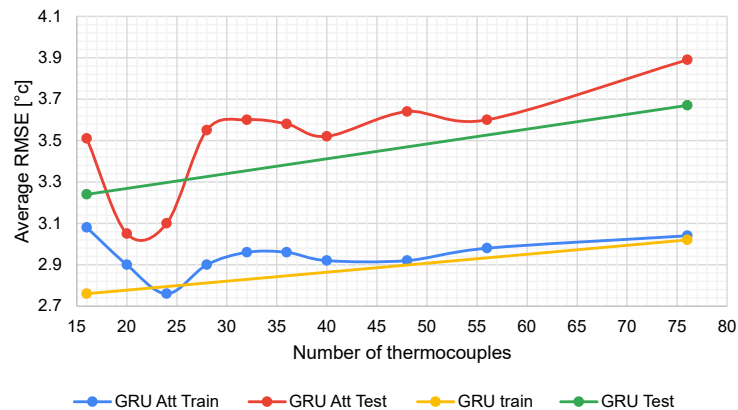


Figure 14. Average RMSE behavior when increasing the output thermocouples to predict.

Table 9. Average RMSE results when increasing the number of thermocouples to predict in the GRU attention model.

Number of Thermocouples	GRU Att Train	GRU Att Test
16	3.08	3.51
20	2.90	3.05
24	2.76	3.10
28	2.90	3.55
32	2.96	3.60
36	2.96	3.58
40	2.92	3.52
48	2.92	3.64
56	2.98	3.60
76	3.04	3.89

5.6. Root Mean Squared Error Distribution by Each Thermocouple in the Test Set

Figure 15 illustrates the boxplot of the RMSE obtained by each thermocouple in the test set. In particular, the four quadrants of the furnace are separated. These quadrants are named as follows: northwest (section 18), southwest (section 19), southeast (section 20) and northeast (section 21). From the results, the southeast quadrant presents the lower error, while the southwest quadrant presents the highest. In general, the mean value of RMSE for each thermocouple is near to 0.4, reaching a maximum value of 1.75.

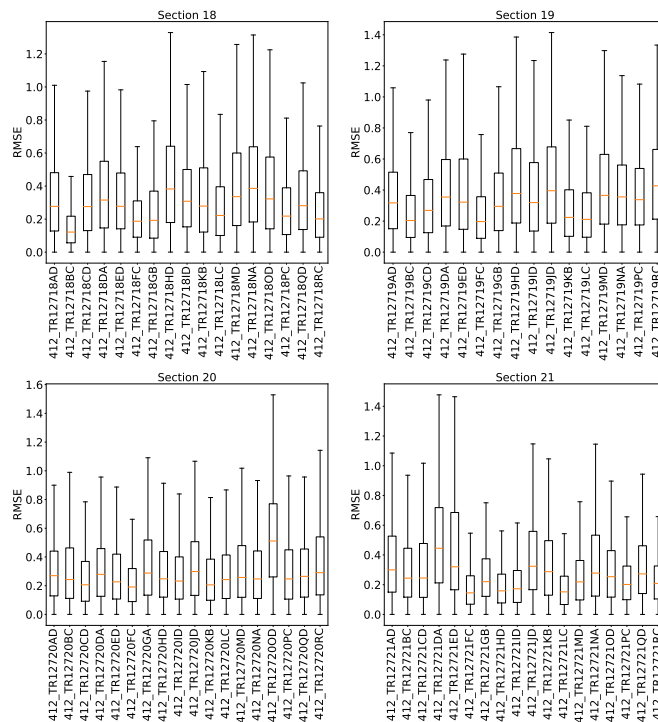


Figure 15. Absolute error analysis by each thermocouple in the test set.

6. Conclusions

This work has shown the development of a multivariate time series deep learning model to predict the temperature behavior of a lining furnace. The developed model is based on an attention mechanism in the encoder–decoder approach of a recurrent neural network. The validation of the model was performed using data acquired in an industrial ferronickel furnace over a period of 5 years. The model considered the historical behavior of 49 variables involved in ferronickel production. Among these variables were the electrode current, voltage, power and position, besides the electric arc, the chemical composition and the temperature measured by the thermocouples themselves. These results were validated by a study carried out in terms of the Average RMSE calculated in 76 different thermocouples located in the furnace lining side-wall at four different heights.

The principal conclusions of this work are as follows:

- The temperature of the lining furnace at different heights of the wall and in different sectors was satisfactorily predicted using the developed deep learning model.
- The results showed that the prediction time influenced the obtained Average RMSE, which was better when predicted in a time window of 1 h in the future when the attention mechanism was used. RMSE values increased as the time window increased.
- A comparison between four different approaches using GRU, LSTM, and their attention-based variants was performed. The best RMSE results were obtained using the GRU attention-based model.

- Three different time series cross-validation procedures were used: the 7-fold moving origin time series split, the Blocking Time Series Split and the Blocking Time Series Split with static test set. The results showed that, over time, the model lost its ability to correctly predict temperatures. Therefore, it is recommended to retrain the model every year to maintain an RMSE value of around 4 °C.
- A study increasing the number of thermocouples to predict from 16 to 76 was carried out. The results showed that the Average RMSE was maintained at a value near to 4 °C, which is allowed in the furnace operation due to the normal operation conditions.

As general conclusions, we can highlight that this work aimed to provide and validate a forecast temperature methodology that is applied to an electric arc furnace. The validation was performed by using real data from a furnace of the Cerro Matoso S.A. and results were validated by staff from the same company. Although the methodology was implemented in this furnace, the paper presents the steps to apply it to any multivariable process to predict the behavior of a variable.

As future works, the following ideas will be explored:

- An online learning-based stream data approach will be developed to evaluate damages in the refractory walls using the developed model. Moreover, the concept of drift detection and treatment in these variables will be studied.
- The methodology will be adapted to forecast other important variables in this furnace, such as the thickness of the refractory wall by predicting, among others, the flow heat in these walls. Since thickness can be measured directly by the operational conditions, it can be obtained by a model that uses forecasted variables.

Author Contributions: All authors contributed to the development of this work; their specific contributions are as follows: conceptualization, D.A.T., C.P., B.R., J.R. and W.V.; data organization and preprocessing, D.F.G.-R., J.X.L.-M., D.A.T. and F.P.; methodology, J.X.L.-M., D.F.G.-R., D.A.T. and C.P.; validation, J.X.L.-M., D.A.T., C.P., F.P., W.V., J.R. and B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the Colombian Ministry of Science through grant number 786, “Convocatoria para el registro de proyectos que aspiran a obtener beneficios tributarios por inversión en CTel”. This work has been partially funded by the Spanish Agencia Estatal de Investigación (AEI)—Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER) through the research project DPI2017-82930-C2-1-R, and by the Generalitat de Catalunya through the research project 2017-SGR-388.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors express their gratitude to Luis Bonilla for providing the dataset and some very useful information about the furnace operation. In the same manner, we thank Janneth Ruiz, Cindy Lopez and Carlos Galeano Urueña for their support throughout the development of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RMSE	Root mean square error
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
SHM	Structural health monitoring
EAF	Electric arc furnace
CNN	Convolutional neural network
CMSA	Cerro Matoso S.A.

References

- Anaya, M. Design and Validation of a Structural Health Monitoring System Based on Bio-Inspired Algorithms. Ph.D. Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2016.
- Tibaduiza Burgos, D.A.; Gomez Vargas, R.C.; Pedraza, C.; Agis, D.; Pozo, F. Damage Identification in Structural Health Monitoring: A Brief Review from its Implementation to the Use of Data-Driven Applications. *Sensors* **2020**, *20*, 733. [[CrossRef](#)]
- Diamanti, K.; Soutis, C. Structural health monitoring techniques for aircraft composite structures. *Prog. Aerosp. Sci.* **2010**, *46*, 342–352. [[CrossRef](#)]
- Tibaduiza, D. Design and Validation of a Structural Health Monitoring System for Aeronautical Structures. Ph.D. Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2013.
- Senyurek, V. Detection of cuts and impact damage at the aircraft wing slat by using Lamb wave method. *Measurement* **2015**, *67*, 10–23. [[CrossRef](#)]
- Rytter, A. Vibrational Based Inspection of Civil Engineering Structures. Ph.D. Thesis, Department of Building Technology and Structural Engineering, Aalborg University, Aalborg, Denmark, 1993.
- Kaloo, M.R.; Hu, J.W. Damage Identification and Performance Assessment of Regular and Irregular Buildings Using Wavelet Transform Energy. *Adv. Mater. Sci. Eng.* **2016**, *2016*, 11. [[CrossRef](#)]
- Yamamoto, K.; Miyamoto, R.; Takahashi, Y.; Okada, Y. Experimental Study about the Applicability of Traffic-induced Vibration for Bridge Monitoring. *Eng. Lett.* **2018**, *26*, pp. 276–280.
- Fan, X.P.; Lu, D.G. Reliability prediction of bridges based on monitored data and Bayesian dynamic models. In *Key Engineering Materials*; Trans Tech Publ.: Freienbach, Switzerland, 2014; Volume 574, pp. 77–84.
- Endsley, A.; Brooks, C.; Harris, D.; Ahlborn, T.; Vaghefi, K. Decision support system for integrating remote sensing in bridge condition assessment and preservation. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2012*; International Society for Optics and Photonics: San Diego, CA, USA, 2012; Volume 8345, p. 834548.
- Tibaduiza, D.A.; Leon-Medina, J.X.; Gomez, R.; Ricardo, J.; Rueda, B.; Zurita, O.; Forero, J.C. Structural Health Monitoring System for Furnace Refractory Wall Thickness Measurements at Cerro Matoso SA. In *European Workshop on Structural Health Monitoring*; Rizzo, P., Milazzo, A., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 414–423.
- Voermann, N.; Gerritsen, T.; Candy, I.; Stober, F.; Matyas, A. Furnace technology for ferro-nickel production—an update. In *Proceedings of the International Laterite Nickel Symposium*, Charlotte, NC, USA, 14–18 March 2004.
- Jiao, K.; Zhang, J.; Hou, Q.; Liu, Z.; Wang, G. Analysis of the relationship between productivity and hearth wall temperature of a commercial blast furnace and model prediction. *Steel Res. Int.* **2017**, *88*, 1600475. [[CrossRef](#)]
- Leon-Medina, J.X.; Camacho-Olarte, J.; Rueda, B.; Vargas, W.; Bonilla, L.; Ruiz, J.; Sofrony, J.; Guerra-Gomez, J.A.; Restrepo-Calle, F.; Tibaduiza, D.A. Monitoring of the refractory lining in a shielded electric arc furnace: An online multitarget regression trees approach. *Struct. Control. Health Monit.* **2022**, *29*, e2885. [[CrossRef](#)]
- Klimas, M.; Grabowski, D. Application of Long Short-Term Memory Neural Networks for Electric Arc Furnace Modelling. In *Intelligent Data Engineering and Automated Learning—IDEAL 2021*; Yin, H., Camacho, D., Tino, P., Allmendinger, R., Tallón-Ballesteros, A.J., Tang, K., Cho, S.B., Novais, P., Nascimento, S., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 166–175.
- Leon-Medina, J.X.; Camacho, J.; Gutierrez-Osorio, C.; Salomón, J.E.; Rueda, B.; Vargas, W.; Sofrony, J.; Restrepo-Calle, F.; Pedraza, C.; Tibaduiza, D. Temperature Prediction Using Multivariate Time Series Deep Learning in the Lining of an Electric Arc Furnace for Ferronickel Production. *Sensors* **2021**, *21*, 6894. [[CrossRef](#)] [[PubMed](#)]
- McCoy, J.T.; Auret, L. Machine learning applications in minerals processing: A review. *Miner. Eng.* **2019**, *132*, 95–109. [[CrossRef](#)]
- Yang, Q.; Shen, D. Learning Damage Representations with Sequence-to-Sequence Models. *Sensors* **2022**, *22*, 452. [[CrossRef](#)] [[PubMed](#)]
- Leon-Medina, J.X.; Vargas, R.C.G.; Gutierrez-Osorio, C.; Jimenez, D.A.G.; Cardenas, D.A.V.; Torres, J.E.S.; Camacho-Olarte, J.; Rueda, B.; Vargas, W.; Esmeral, J.S.; et al. Deep Learning for the Prediction of Temperature Time Series in the Lining of an Electric Arc Furnace for Structural Health Monitoring at Cerro Matoso (CMSA). *Eng. Proc.* **2020**, *2*, 23.
- Dong, M.; Grumbach, L. A hybrid distribution feeder long-term load forecasting method based on sequence prediction. *IEEE Trans. Smart Grid* **2019**, *11*, 470–482. [[CrossRef](#)]
- Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [[CrossRef](#)] [[PubMed](#)]
- Zhao, H.; Wang, Y.; Duan, J.; Huang, C.; Cao, D.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; Zhang, Q. Multivariate Time-series Anomaly Detection via Graph Attention Network. In *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy, 17–20 November 2020.
- Barić, D.; Fumić, P.; Horvatić, D.; Lipic, T. Benchmarking attention-based interpretability of deep learning in multivariate time series predictions. *Entropy* **2021**, *23*, 1–23. [[CrossRef](#)] [[PubMed](#)]
- Gangopadhyay, T.; Tan, S.Y.; Jiang, Z.; Meng, R.; Sarkar, S. Spatiotemporal Attention for Multivariate Time Series Prediction and Interpretation. In *Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 6–11 June 2021.

25. Janzen, J.; Gerritsen, T.; Voermann, N.; Veloza, E.; Delgado, R. Integrated furnace controls: Implementation on a covered-arc (shielded arc) furnace at cerro matoso. In Proceedings of the 10th International Ferroalloys Congress, Cape Town, South Africa, 1–4 February 2004; Volume 1, p. 4.
26. Voermann, N.; Gerritsen, T.; Candy, I.; Stober, F.; Matyas, A. Developments in furnace technology for ferronickel production. In Proceedings of the 10th International Ferroalloys Congress, Cape Town, South Africa, 1–4 February 2004, p. 455.
27. Mills, T.C. *Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting*; Academic Press: Cambridge, MA, USA, 2019.
28. Bontempi, G.; Ben Taieb, S.; Le Borgne, Y.A., Machine Learning Strategies for Time Series Forecasting. In *Tutorial Lectures, Proceedings of the Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, 15–21 July 2012*; Aufaure, M.A., Zimányi, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 62–77. [[CrossRef](#)]
29. Du, S.; Li, T.; Yang, Y.; Horng, S.J. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing* **2020**, *388*, 269–279. [[CrossRef](#)]
30. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025.
31. Rémy, P. Keras Attention Mechanism. 2021. Available online: <https://github.com/philipperemy/keras-attention-mechanism> (accessed on 24 September 2021).
32. Camacho-Olarte, J.; Torres, J.E.S.; Jimenez, D.A.G.; Medina, J.X.L.; Vargas, R.C.G.; Cardenas, D.A.V.; Gutierrez-Osorio, C.; Rueda, B.; Vargas, W.; Burgos, D.A.T.; et al. A Data Cleaning Approach for a Structural Health Monitoring System in a 75 MW Electric Arc Ferronickel Furnace. *Eng. Proc.* **2020**, *2*, 21. [[CrossRef](#)]
33. Schnaubelt, M. A comparison of machine learning model validation schemes for non-stationary time series data. In *Technical Report, FAU Discussion Papers in Economics, No. 11/2019*; Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics: Nürnberg, Germany, 2019.

**B. Anexo: Acuerdo de trabajo conjunto
South 32 Cerromatoso S.A**

Bogotá D.C., 8 de febrero de 2022

Señores
Universidad Nacional de Colombia
Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia

Asunto: Carta de intención de apoyo a trabajo final de maestría

Estimados señores:

Reciban un cordial saludo.

En mi calidad de líder de Innovación y mejoramiento de Cerro Matoso S.A. informo que la propuesta de trabajo final de maestría ***“Aprendizaje profundo para la predicción de temperatura en las paredes refractarias de un horno de arco eléctrico”*** del estudiante **DIEGO FERNANDO GODOY ROJAS**, identificado con C.C. 1031173820, quien es estudiante de Maestría en Automatización Industrial en la Universidad Nacional de Colombia, contará con apoyo de la compañía. Esto, en el marco del proyecto de innovación: **“Incremento de la capacidad operativa y eficiencia de producción de Cerro Matoso S.A.”**, vigente ante el Ministerio de Ciencia Tecnología e Innovación de Colombia.

El apoyo por parte de Cerro Matoso S.A. se basa en proveer datos de la operación de hornos eléctricos para la producción de ferróniquel con los cuales el estudiante desarrollará modelos matemáticos de estimación y predicción de temperatura en las paredes de los hornos que serán usados como prototipo para la validación de sus desarrollos. La información proporcionada por la compañía está cubierta por un acuerdo de confidencialidad vigente, razón por la cual cualquier publicación académica futura que contenga datos o información compartida debe tener aval de la compañía.

Cordialmente,



Janneth E. Ruiz A.

Líder de Innovación y Mejoramiento

Cerro Matoso S.A.

T +57 4 7623257

M +57 315 895 50 18

Janneth.A.Ruiz@south32.net