



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO



**José Francisco Núñez Castro**

# Aprendizaje automático en fusión nuclear con Deep Learning

Informe Proyecto de Título de Ingeniero Civil Electrónico



**Escuela de Ingeniería Eléctrica  
Facultad de Ingeniería**

Valparaíso, 07 de diciembre de 2017



# Aprendizaje automático en fusión nuclear con Deep Learning

José Francisco Núñez Castro

Informe Final para optar al título de Ingeniero Civil Electrónico,  
aprobada por la comisión de la  
Escuela de Ingeniería Eléctrica de la  
Facultad de Ingeniería de la  
Pontificia Universidad Católica de Valparaíso  
conformada por

Sr. Gonzalo Farías Castro

Profesor Guía

Sr. Gabriel Hermosilla Vigneau

Segundo Revisor

Sr. Sebastián Fingerhuth Massmann

Secretario Académico

Valparaíso, 07 de diciembre de 2017

*A mis padres José Núñez y Marisol Castro.*

# Agradecimientos

En este trabajo de titulación quisiera agradecer a todos quienes me acompañaron en mis años universitarios, a quienes me conocieron y con quienes compartí, en especial a mis amigos de carrera con los cuales superé metas, dificultades y forme vínculos especiales, a mis profesores los cuales me enseñaron, guiaron y me motivaron a continuar día a día, a mis padres, hermano y a mi novia que pusieron en mí su confianza y comprensión para ayudar a superarme y llevar a término mi carrera. Finalmente, un agradecimiento especial a mi mascota Jack compañero de largos desvelos.

*Valparaíso, 7 de diciembre de 2017*

J. N.

# Resumen

El enfoque Deep Learning o aprendizaje profundo en español, es un tipo de aprendizaje automático que puede crear modelos de redes neuronales profundos, es decir, de varios niveles que nos permite obtener características abstractas dependiendo del número de capas ocultas con que se diseñe la red neuronal base que se desee utilizar.

Para este proyecto la red neuronal artificial estudiada fue la red neuronal recurrente, la cual puede manejar datos en secuencias, pero dependiendo de la cantidad de capas ocultas con las que se diseñe puede conllevar que las primeras capas no se entrenen lo suficiente para obtener “memoria” a largo plazo, ya que, al realizar propagación hacia atrás, el error que se tiene en la salida se va degradando en información haciendo que se pierda en las capas iniciales. Una opción a este problema fue el desarrollo de las redes neuronales recurrentes Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU), las que fueron diseñadas para evitar la problemática del gradiente desvaneciente. Su estructura básica está compuesta por tres capas, una capa de entrada, una capa oculta y una capa de salida.

Las redes LSTM y GRU son arquitecturas que nos permiten trabajar con datos en secuencias, la característica recurrente permite a estas redes que la información logre propagarse por el largo de la cadena de análisis, al ser algoritmos del enfoque Deep Learning podemos utilizarlas para analizar grandes cantidades de datos. Estas redes pueden ser utilizadas para extraer características de cualquier dato en secuencia temporal.

En este proyecto de titulación se utilizarán para extraer características que generen pronóstico de series temporales y clasificación de señales a partir de bases de datos provenientes de experimentos realizados en dispositivos experimentales de fusión nuclear para generar estrategias de evitación y mitigación de la problemática de la disrupción nuclear.

Palabras claves: Deep Learning, LSTM, GRU, pronóstico, clasificación, fusión nuclear, disrupción.

# Abstract

The Deep Learning approach is a type of machine learning that can create deep models, that is, of several levels that allows us to obtain abstract features depending on the number of hidden layers want to be used with which it is designed it can mean that the first layers are not trained enough to obtain "memory" in the long term, since when performing backward propagation, the error that is present in the output degrades into information causing it to be lost in the initial layers. One option to this problem was the development of the recurrent neural networks Deep Learning, the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU) both were designed to avoid the problem of the fading gradient. Its basic structure is composed of three neurons, an input layer, a hidden layer and an output layer.

The LSTM and GRU networks are architectures that allow us to work with data in sequences, the recurrent feature allows these networks that the information is able to propagate along the length of the chain of analysis, Deep Learning's architectures we can use them to analyze large amounts of data. These networks can be used to extract characteristics of any data in temporal sequence, in this university degree project they will be used to extract features to obtain forecast of time series of nuclear fusion and classification of nuclear fusion signals from databases from tests carried out in nuclear fusion experimental devices to generate avoidance and mitigation strategies for nuclear disruption.

Keywords: Deep Learning, LSTM, GRU, forecast, classification, nuclear fusion, disruption.

# Índice general

Introducción.....	1
Objetivos generales.....	3
Objetivos específicos .....	3
<b>1 Energía Nuclear.....</b>	<b>4</b>
1.1 El mundo necesita energía .....	4
1.2 Energía limpia .....	5
1.3 Fisión nuclear.....	5
1.4 Fusión nuclear.....	6
1.5 Dispositivos de fusión nuclear.....	7
1.5.1 TJ-II.....	7
1.5.2 Tokamak JET .....	7
1.6 Proyecto ITER.....	7
<b>2 Problemática y motivación.....</b>	<b>9</b>
2.1 Problemática en la fusion nuclear.....	9
2.2 Disrupción nuclear .....	9
2.2.1 Mitigación y evitación.....	10
2.3 Datos de información del plasma.....	10
2.3.1 Base de datos JET.....	11
2.3.2 Base de datos KB5V .....	12
2.3.3 Base de datos TJ-II.....	13
2.4 Motivación en base a la inteligencia artificial .....	14
2.4.1 Series de tiempo .....	14
2.4.2 Reconocimiento de patrones .....	15
2.4.3 Pronóstico de series temporales .....	15
2.5 Redes neuronales artificiales .....	16
2.5.1 Backpropagation .....	17
2.5.2 Aprendizaje supervisado.....	17
2.5.3 Aprendizaje no supervisado .....	18
2.6 Redes Neuronales Recurrentes.....	18
2.7 Deep Learning.....	19

---

2.8 Redes recurrentes Profundas .....	20
2.8.1 Problema de dependencia a largo plazo .....	20
2.8.2 Gradiente Desvaneciente.....	20
2.9 Long short-term memory .....	20
2.9.1 Funcionamiento .....	21
2.10 Gated recurrent unit .....	23
2.10.1 Funcionamiento .....	23
<b>3 Implementación de algoritmos.....</b>	<b>25</b>
3.1 Pronóstico de series temporales.....	25
3.1.1 Rendimiento del pronóstico.....	26
3.2 Clasificación de señales.....	26
3.2.1 Cálculo de centroides.....	27
3.2.2 Análisis de peaking factor .....	27
3.2.3 Alarmas para mitigación y evitación .....	28
3.3 Entorno de trabajo .....	28
3.3.1 Tensorflow y Keras .....	28
3.4 Diseño y construcción de la red para pronóstico .....	29
3.4.1 Diseño e implementación de la red .....	30
3.5 Diseño y construcción de la red para clasificación.....	30
3.5.1 Diseño e implementación de la red .....	31
<b>4 Resultados Obtenidos .....</b>	<b>33</b>
4.1 Pronóstico base JET .....	34
4.2 Pronóstico base TJ-II .....	37
4.3 Clasificación de series temporales .....	43
4.3.1 Cálculo de centroides.....	43
4.3.2 Análisis de Peaking Factor .....	45
4.3.3 Resultados obtenidos centroides y peaking factor .....	46
4.4 Clasificación de señales disruptivas .....	46
<b>Discusión y conclusiones.....</b>	<b>49</b>
<b>Bibliografía .....</b>	<b>52</b>



# Introducción

El mundo actual se encuentra en busca de nuevas formas de energía, los recursos no renovables, tales como combustibles fósiles, han marcado el desarrollo tecnológico en siglos pasados pero con consecuencias negativas para el planeta, como lo es el calentamiento global, que ha perjudicado a todos los ecosistemas y seres vivos que habitan este mundo, los combustibles fósiles son recursos limitados y altamente contaminantes, debido al constante crecimiento de ciudades y tecnologías es necesario contar con energías verdes que logren sustentar la demanda energética y no añadan más daño a nuestro mundo. Se espera que la energía nuclear de fusión sea el camino crucial en el ámbito de energías verdes. La energía nuclear de fusión es una energía no renovable pero que cuenta con grandes recursos disponibles a diferencia de la energía nuclear de fisión que utiliza uranio para desintegrar y genera residuos radiactivos que demoran años en degradarse, la fusión nuclear utiliza hidrógeno, abundante en la tierra y genera residuos de helio, no contaminante ni radioactivo. En este escenario la energía de fusión nuclear nace como opción, ya que es sustentable y altamente abundante en producción de energía.

El primer capítulo de este texto está dedicado a la explicación de cómo la fusión nuclear nace del deseo de poder manejar energías comparables a las que se presentan en las estrellas y como se generan los datos con que se trabajaron en este proyecto, desde 1950 se han ideado y construido dispositivos que hagan realizable llevar la energía del sol a un laboratorio. Dispositivos experimentales tales como los diseños Stellarator y Tokamak, han logrado grandes avances en la manipulación de plasma y se sigue trabajando actualmente en dispositivos que lleven aún más allá este concepto [1]. Uno de ellos es el proyecto ITER, que busca demostrar que la producción de energía de fusión es posible y segura.

Al ser un nuevo tipo de energía también presenta grandes desafíos para su producción y entre ellos está la disrupción nuclear, una disrupción se genera cuando el confinamiento del plasma en el reactor se pierde y esta choca con las paredes generando daños a la estructura, este problema que no ha podido ser evitado y se busca constantemente sistemas que logren la evitación y mitigación de esta. Cada experimento probado en el dispositivo corre con el riesgo de provocar una disrupción nuclear, es por eso que de las distintas pruebas se han obtenido cantidades inmensas de información del plasma, estos datos son de gran utilidad para conocer la dinámica física del plasma.

La cantidad de datos generados están grande que es imposible analizar todos los datos manualmente para un operario, cada experimento obtiene un sin número de señales que entregan información relevante del plasma en el momento en que ocurre una disrupción o no. Este trabajo busca apoyar las distintas técnicas que existen para pronosticar la dinámica del plasma a cierto horizonte de tiempo y clasificar una señal como disruptiva o no disruptiva, usando el aprendizaje automático y el aprendizaje profundo (Deep Learning).

En el segundo capítulo se explica el concepto de aprendizaje automático y como el uso de la arquitectura de una red neuronal biológica es vital para el aprendizaje de la máquina, esta neurona ha sido replicada en la llamada red neuronal artificial que imita los procesos producidos en la neurona biología a términos matemáticos. Este capítulo contempla el objetivo principal de este trabajo, el cual es implementar las redes neuronales recurrentes a nivel profundo, para ello fueron desarrolladas algoritmos basados en las redes Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU) que superan esta dificultad y pueden ser entrenadas con grandes cantidades de datos.

Un tercer capítulo explica de modo concreto la aplicación de Deep Learning, se describe el uso de dos tipos de arquitectura de red neuronal recurrente, la red LSTM y la red GRU, de esta forma se genera un enfoque empírico al uso de estas redes. En este capítulo se detalla los procedimientos realizados para cada algoritmo y experimento en primer lugar para el pronóstico y luego para la clasificación, se explican la adquisición de datos, creación de bases de datos e implementación de las redes neuronales recurrentes LSTM y GRU.

Finalmente, el cuarto capítulo recopila todos los resultados obtenidos a partir de la implementación de los algoritmos de pronóstico de series temporales y clasificación de señales, en este apartado se comentarán los resultados para analizar los mejores casos junto a graficas que muestren el desempeño de cada una de las pruebas. Dependiendo del algoritmo utilizado se ocuparán cierto medidas de desempeño para lograr un óptimo resultado, ya sea exactitud y precisión, distancias euclídeas y porcentaje de seguridad del modelo, ya que para el proyecto internacional ITER se deben generar modelos de aprendizaje automático con tasas de acierto sobre el 95% y tasas de falsas alarmas bajo el 5%, siendo estos el criterio general de validación de los modelos de clasificación.

Se plantean entonces los siguientes objetivos, detallando cada uno de ellos, se clasificarán en objetivos generales y específicos que serán el motor principal de investigación para este proyecto de titulación.

## **Objetivos generales**

Diseñar e implementar algoritmos de aprendizaje automático en dispositivos de fusión nuclear con Deep Learning.

Con este objetivo principal en mente se esperarán lograr los siguientes objetivos específicos a lo largo de todo el desarrollo del proyecto:

## **Objetivos específicos**

- Estudiar el enfoque Deep Learning en aprendizaje automático.
- Estudiar los principales problemas asociados a la fusión nuclear.
- Implementar un sistema de aprendizaje automático con Deep Learning en un problema seleccionado para un dispositivo de fusión nuclear real.

# 1 Energía Nuclear

## 1.1 El mundo necesita energía

La predicción realizada por las Organización de Naciones Unidas (ONU), indica que la población mundial crecerá de los 7.3 billones actuales a 9.2 billones para el 2040 [1], esto quiere decir que la demanda energética debe incrementarse substancialmente para ese periodo. El crecimiento de la población y el aumento de la esperanza de vida en países en vías de desarrollo causará una fuerte demanda de energía.

La demanda de energía aumenta cada año, especialmente la energía eléctrica que constituye el 42% de la energía producida [1]. Energías renovables como la solar y eólica producen energías limpias, pero actualmente son costosas por unidad producida e intermitentes en los sistemas de distribución de energía. Todas las organizaciones internacionales sugieren que la energía nuclear debe ser la energía del futuro debido a su característica de ser amigable con el medioambiente y su capacidad de abastecer sistemas a gran escala. El mundo necesitará en el futuro próximo aumentar el abastecimiento de energías, especialmente la electricidad producida por energías limpias.

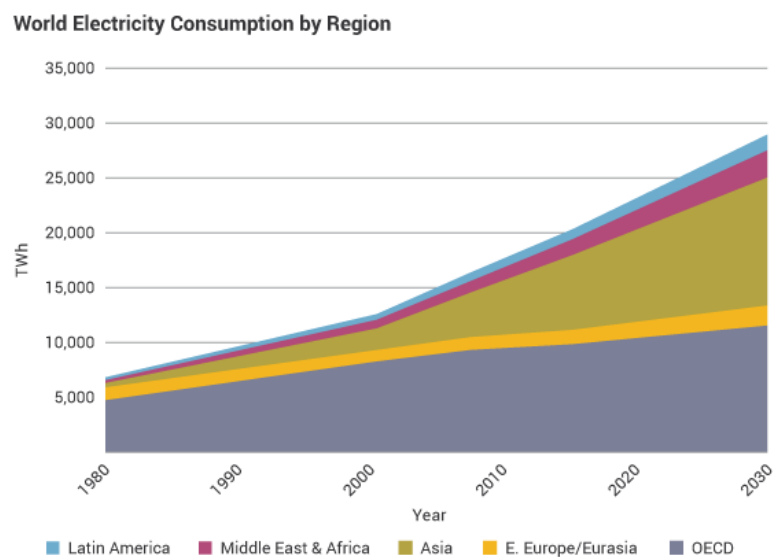


Figura 1-1: Consumo de energía mundial por región (fuente: <http://www.world-nuclear.org>)

## 1.2 Energía limpia

Existen diferentes tipos de recursos para la generación de energía tales como el carbón, el gas natural y el petróleo. El carbón no es escaso a nivel mundial, pero debe ser transportado en grandes cantidades desde donde abunda hacia donde se necesita, generando implicaciones económicas y medioambientales (emisión de carbono a la atmósfera), el gas natural es abundante y se comercializa cada vez a más largas distancias, y su obtención aumenta debido al incremento de las tecnologías para extraerlo, y finalmente el petróleo es limitado pero su uso en el área automotriz lo hace poseer un valor único.

El panel intergubernamental de la ONU sobre el Cambio Climático (IPCC) ha examinado exhaustivamente el calentamiento global y ha llegado a un consenso de que el fenómeno es real y representa una amenaza ambiental significativa para el próximo siglo si el uso de combustibles fósiles continúa en los niveles actuales y la energía nuclear tiene un rol importante que jugar en la reducción de los gases de invernadero, ya que, representa un 11% total de la generación de electricidad a nivel mundial [1].

## 1.3 Fisión nuclear

La energía de fisión nuclear es el proceso principal producido por la energía nuclear, la fisión es un tipo de reacción en la cual un neutrón de cierta energía colisiona con un núcleo pesado, es decir, un elemento de alto número atómico como el Uranio. Esto tiene por consecuencia la formación de un núcleo compuesto muy excitado e inestable, el cual se dividirá en dos núcleos más pequeños de tamaño parecido y que a su vez liberará neutrones de radiación Beta, Gamma y Neutrinos. Esta reacción libera una gran cantidad de energía, cercana a los 200 MeV, que es transformada en energía térmica y luego en energía eléctrica [2].

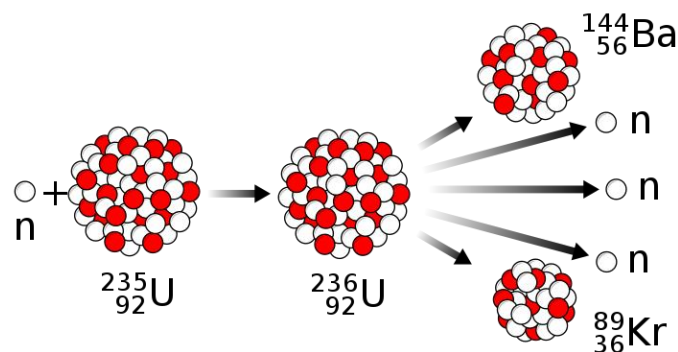


Figura 1-2: Esquema simplificado de la fisión nuclear (fuente: <https://commons.wikimedia.org>)

Hoy en día, la fisión es usada principalmente para la generación de electricidad en reactores nucleares de potencia, en donde la reacción en cadena es controlada, gracias a elementos llamados Moderadores, por ejemplo, Agua liviana (H<sub>2</sub>O), Agua pesada (D<sub>2</sub>O), Grafito o Sodio Metálico, los cuales se encargan de disminuir la velocidad de los neutrones eyectados junto con

elementos absorbedores de neutrones como el Boro, Cadmio o Gadolinio. El combustible más usado para la generación de energía eléctrica es el uranio enriquecido al 3,5% (Uranio-235), el cual genera residuos nucleares que demoran años en degradarse debido a su vida media prolongada.

### 1.4 Fusión nuclear

La fusión nuclear es otro tipo de reacción nuclear en la cual energía extra es liberada cuando se fusionan núcleos ligeros. Este tipo de reacción se produce en nuestro Sol y en las demás estrellas. Esta reacción ocupa isótopos de Hidrógeno, como el Deuterio y Tritio, los cuales se fusionan para formar Helio, este proceso libera neutrones, Helio y radiación. A diferencia del proceso de fisión nuclear, se necesitan temperaturas y presiones extremas para iniciar y sostener la reacción de fusión, lo que la convierte en un desafío pues no se puede fácilmente alcanzar temperatura y presión elevada que existe en el interior del Sol en la Tierra, aunque la investigación y el desarrollo dirigidos a lograr una fusión controlada ha dado lugar a avances significativos en la reciente década y se han conseguido métodos que son capaces de vencer la repulsión de los elementos generando así grandes cantidades de energía (4-20MeV) por fusión [2].

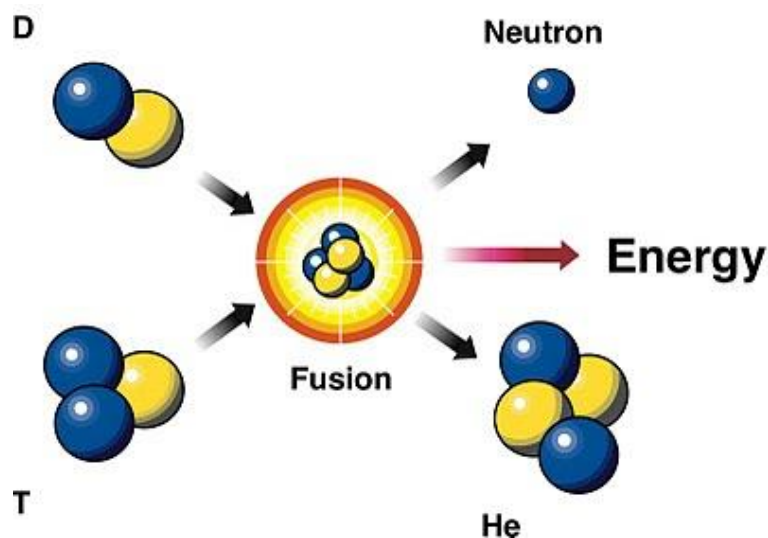


Figura 1-3: Esquema simplificado de la fusión nuclear (fuente: <http://interestingenergyfacts.blogspot.cl>)

Debido a que la fusión nuclear toma lugar solo a temperaturas comparables a las que se producen en el centro del sol, implementar un reactor de fusión nuclear involucra el desarrollo de técnicas para crear y confinar "plasma". Una opción para generar fusión nuclear es a través del Confinamiento Magnético, en el cual el plasma se confina mediante campos magnéticos generados en el reactor con tal de formar una barrera entre las partículas cargadas del plasma y la pared de la cámara que lo contiene, actualmente se encuentran en desarrollo reactores de fusión nuclear por confinamiento magnético, siendo el más importante el proyecto ITER, un proyecto multinacional que busca la construcción y puesta en marcha de un reactor termonuclear de fusión, hoy en día la reacción por fusión no es un proceso viable, es decir, se necesita más energía en producir el proceso de fusión que la que se obtiene por este proceso.

## 1.5 Dispositivos de fusión nuclear

Desde mediados de siglo XX se ha investigado y desarrollado tecnología para obtener una reacción controlada, debido a la enorme dificultad de soportar la enorme presión y temperatura que se requiere para obtener fusión nuclear. Actualmente existen diferentes tipos de dispositivos experimentales de fusión por confinamiento magnético.

### 1.5.1 TJ-II

Es un Stellarator flexible de tipo heliac de tamaño medio y es el segundo más grande en operación en Europa [3]. Un dispositivo Stellarator es un diseño americano utilizado para confinar plasma dentro de campos magnéticos. Es uno de los primeros diseños de confinamiento de magnético, inventado por Lyman Spitzer y construido en 1950 en la Universidad de Princeton. Su estructura poloidal genera un campo, gracias a las bobinas externas a la cámara de forma similar a la como se genera la componente toroidal. Su principal problema en los primeros modelos fue que el plasma se filtraba entre las líneas del campo. Su nombre se debe a la posibilidad de generar energía como en las estrellas (“stella”) y al principio físico de generar energía (“generator”).

El Stellarator TJ-II fue diseñado en España con colaboración de EEUU [3] y el primer plasma se produjo en el año 1997, posee un radio mayor de 1,5m, un radio menor de promedio de 0,2m y un campo magnético de 1T. Las altas temperaturas se obtienen mediante la aplicación de 800kW de radiofrecuencia (53GHz) y 1,6MW de inyección de Hidrogeno neutro acelerado a 40keV. TJ-II cuenta con 92 ventanas de acceso en las que se sitúan sistemas de medida para la realización de los diferentes experimentos.

### 1.5.2 Tokamak JET

Es un reactor termonuclear de modelo ruso por confinamiento magnético que tiene forma de cámara toroidal. Su nombre proviene de un acrónimo ruso, que en español se traduciría como (cámara toroidal con bobinas magnéticas). Fue ideado por los físicos soviéticos Ígor Tam y Andréi Sájarov en 1950 [4]. Similar al Stellarator la componente toroidal del campo magnético se genera mediante bobinas externas, haciendo que el embobinado primario induzca el campo sobre el plasma y utilizándolo este mismo como un embobinado secundario.

El mayor reactor de este tipo es JET (toro europeo conjunto) y ha conseguido condiciones de fusión nuclear con factor  $Q > 0.7$  [4], es decir que la energía generada por fusión y energía requerida para sostener la reacción es 0.7 (para que la auto reacción se mantenga debe alcanzar un factor de  $Q > 1$ ).

## 1.6 Proyecto ITER

ITER- “el camino” en latín- será la próxima gran instalación de tipo tokamak es uno de los proyectos más ambiciosos en el mundo hoy en día.

ITER será el primer dispositivo de fusión nuclear en mantener la fusión por largos periodos de tiempo, con un factor ( $Q > 10$ ), además de ser el primer dispositivo en testear tecnologías,

materiales y régimen físicos necesarios para la producción comercial de energía eléctrica basada en fusión. Sus principales objetivos [5] son producir 500 MW de poder de fusión, demostrar el funcionamiento integrado de las tecnologías para una planta de energía de fusión, alcanzar un nivel de plasma, deuterio-tritio, que se mantenga mediante calentamiento interno, probar la capacidad de producir tritio y demostrar las características de seguridad de un dispositivo de fusión.

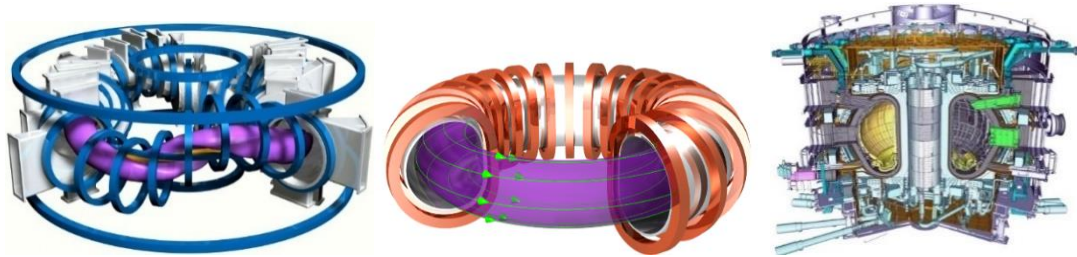


Figura 1-4: Estructuras dispositivos experimentales de fusión nuclear

Se observa en la Figura 1-4 se observan las estructura de estos dispositivos, a la izquierda Stellarator TJ-II (fuente: <http://fusionsites.ciemat.es>), al centro el Tokamak JET (fuente: <http://www.ideen2020.de>) y a la derecha la estructura de ITER (fuente: <https://www.iter.org>), en las imágenes se visualizan sus tipos de bobinas y flujos magneticos



## 2 Problemática y motivación

### 2.1 Problemática en la fusión nuclear

El objetivo principal de este proyecto es comprender la factibilidad de usar el enfoque Deep Learning en las problemáticas que tiene esta nueva energía, para ello se espera lograr un análisis abstracto de las señales obtenidas de los dispositivos experimentales de fusión nuclear bajo el uso de algoritmos de red neuronal recurrente, con este objetivo en mente se espera obtener una comprensión del comportamiento de las señales puede ser de gran utilidad para los investigadores, científicos y físicos que trabajan continúa y exhaustivamente en encontrar precursores en las señales muestreadas provenientes de los sensores que recorren completamente el toroide del dispositivo experimental de fusión nuclear. Son tantas las señales que se estima que solo el 10% de ellas es analizada, por lo tanto, poder realizar un trabajo de análisis y posterior pronóstico de anomalías o características de estas señales puede facilitar la tarea de sobre llevar la principal problemática de la fusión que es la denominada disrupción nuclear.

### 2.2 Disrupción nuclear

Como se mencionó anteriormente la energía de fusión nuclear es la energía que se busca para el futuro, con grandes ventajas corre como favorita entre las energías denominadas verdes, pero también está asociada a grandes problemas como lo son las disrupciones o pérdida de estabilidad en el confinamiento magnético. Es un fenómeno complejo que involucra la estabilidad magnetohidrodinámica (dinámica de fluidos conductores de electricidad en presencia de campos eléctricos y magnéticos) del plasma lo cual resulta en una rápida pérdida de calor y en la culminación de una descarga. El control del plasma puede perderse desencadenando un evento de desplazamiento vertical en el cual el dispositivo puede resultar dañado, en otras palabras, se entiende como una pérdida de confinamiento del plasma en el reactor, haciendo que toda la energía se dirija hacia las paredes internas del dispositivo generando así daños irreparables, que conllevan tiempo y dinero.

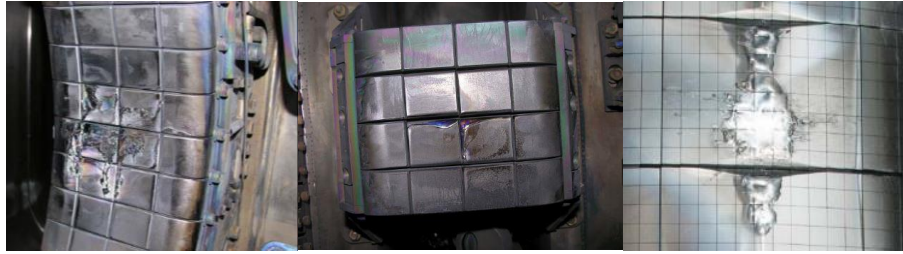


Figura 2-1: Daños en la pared del dispositivo (fuente: <http://www.euro-fusionscipub.org>)

En la Figura 2-1 se observan los daños producidos por la disrupción de plasma, en ellas se ve cuan perjudicial es el daño físico en las paredes del dispositivo.

La disrupción es un fenómeno extremadamente complejo. Hasta la fecha la ocurrencia de disrupciones ha demostrado ser un aspecto difícilmente evitable en los grandes dispositivos experimentales, es de alta prioridad desarrollar sistemas que predigan disrupciones, para obtener estrategias que eviten y mitiguen los daños que conlleva una disrupción.

### 2.2.1 Mitigación y evitación

Se necesitan entonces estrategias que logren la Evitación y Mitigación de la disrupción [6], en ella se busca desarrollar, validar y desplegar estrategias que actúen antes de la primera descarga del plasma, que verifiquen la MHD y estabilidad de control crítico, los últimos programas han hecho esfuerzos por concentrarse mayormente en Mitigación.

La Evitación requiere estudio adicional y se lleva a cabo al inicio de la disrupción, las opciones que se tiene son: reparar y continuar con el experimento, retroceso y apagado (proceso normalmente utilizado) del dispositivo y “aterrizaje suave” (parada rápida) del experimento. Se debe tener en cuenta las diferencias entre los tiempos de recuperación y reinicio para poder realizar cualquier tipo de evitación, además existe la posibilidad de inyección lenta de gas o de pellets.

La Mitigación es necesaria cuándo la evitación falla y conlleva complejas consecuencias. El método candidato para la mitigación es la inyección masiva de gas, que puede incluso re-evitar la disrupción. Una de las desventajas notables de los métodos mencionados son el agotamiento del sistema tras el impacto del gas y la acumulación de polvo para estos inyectables condensables.

### 2.3 Datos de información del plasma

Para el desarrollo de este proyecto se obtuvieron datos de dispositivos experimentales de fusión nuclear, estos datos corresponden a campañas de experimentación, cada campaña posee distintas descargas realizadas en el dispositivo, en la cual se obtienen una gran cantidad de señales que registran el comportamiento del plasma. Estas descargas están agrupadas en dos categorías, descargas disruptivas y descargas no disruptivas, las descargas disruptivas son aquellas descargas que durante la realización del experimento se produjo un evento disruptivo, haciendo que todas las señales registradas en la descargas se etiqueten como señales con comportamiento disruptivo. De igual manera las descargas no disruptivas son aquellas descargas que durante la realización del experimento no se produjo ningún evento disruptivo, haciendo que

todas las señales registradas en la descarga se etiqueten como señales sin comportamiento disruptivo. Cada descarga posee señales de interés que generan información del plasma que se produce en el toroide, estas señales son registradas en términos de tiempo y magnitud, y duran hasta que se produce un evento disruptivo, denominado tiempo de disrupción o hasta que termine el experimento en termino de descargas seguras.

### 2.3.1 Base de datos JET

De las campañas realizadas en el Tokamak JET y facilitada por el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), se obtuvo una base de datos con descargas disruptivas y descargas no disruptivas, con 9 señales de interés.

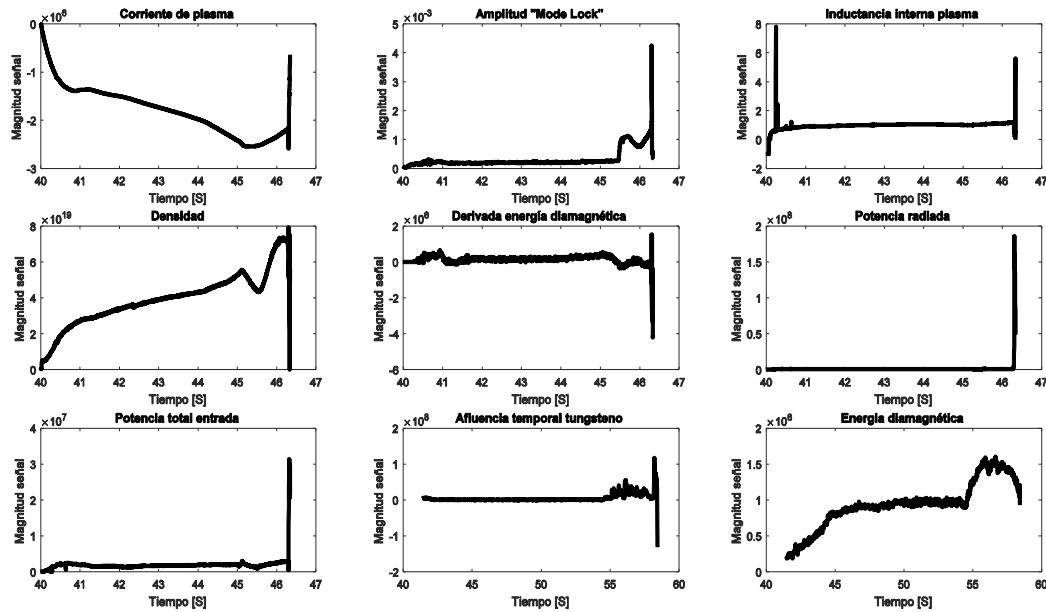


Figura 2-2: Señales de interés JET

En la Figura 2-2 se observan las señales de interés y en la Tabla 2-1 se presentan los nombres de cada señal.

Tabla 2-1: Señales Base de datos JET

Nombre de la señal	Descripción
1 Plasma current	Intensidad de corriente de plasma
2 Mode Lock	Densidad de flujo magnético por Ampere
3 Plasma internal inductance	Inductancia propia del plasma
4 Plasma electronic density	Densidad electrónica del plasma
5 Diamagnetic energy derivative	Derivada energía diamagnética
6 Radiated power	Potencia radiada
7 Total input power	Potencia total de entrada
8 Temporal evolution of W influx level	Evolución temporal de la afluencia del W
9 Diamagnetic energy	Energía diamagnética

Esta base está compuesta por dos campañas, las señales de la 1 a la 7 corresponden a la campaña C2830 con 201 descargas disruptivas y 1036 descargas no disruptivas, las señales de la 8 a 9 corresponden a la campaña C36 con 151 descargas disruptivas y 995 descargas no disruptivas. Estas señales fueron muestreadas a un 1ms con duración en promedio de 20s.

### 2.3.2 Base de datos KB5V

De las campañas realizadas en el Tokamak JET y facilitada por el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), se obtuvo una base de datos con señales correspondientes a una de las cámaras que registran las señales bolométricas del plasma, la KB5V es la cámara vertical que observa el plasma, esta cámara genera 24 señales, de derecha a izquierda se encuentran las señales de la número 1 a 8, en el centro de derecha a izquierda se encuentran las señales de la número 9 a 15 y finalmente en el lado izquierdo de derecha a izquierda se encuentran las señales de la número 17 a 24.

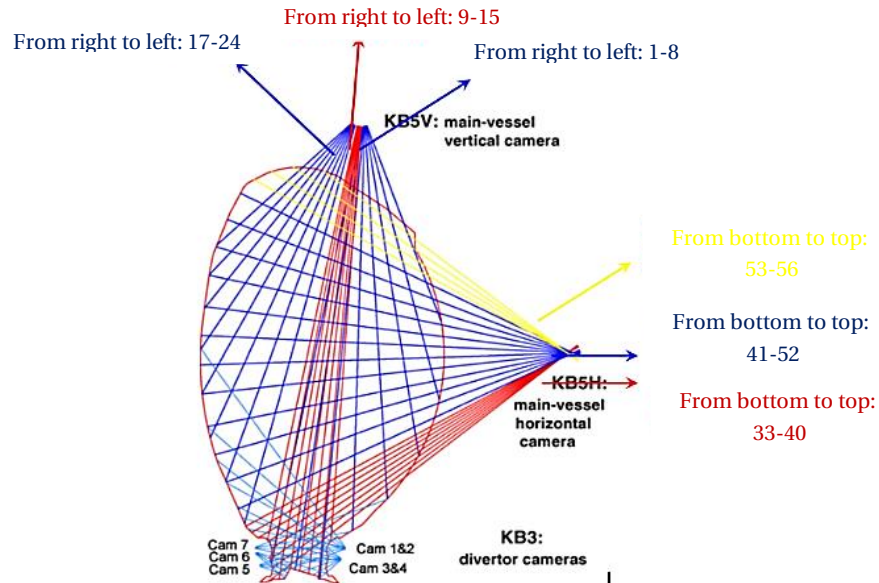


Figura 2-3: Cámara KB5V (fuente: <http://www.euro-fusionscipub.org>)

En la Figura 2-3 se observan la disposición de las señales de interés en la cámara KB5V

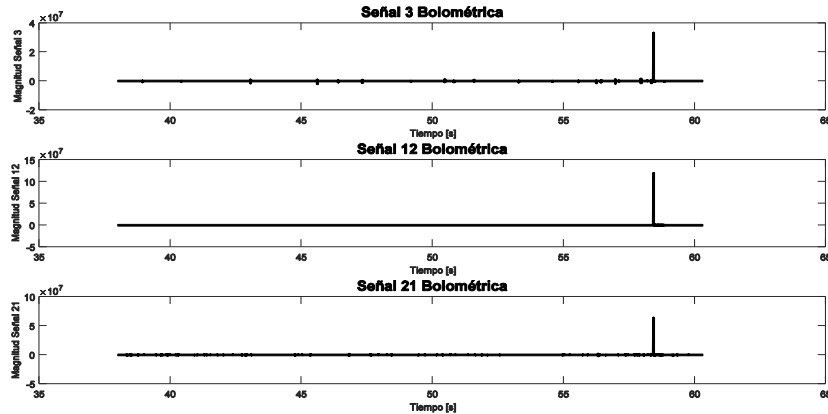


Figura 2-4: Señales de interés de la cámara KB5V

De ellas se seleccionaron 3 señales, una del eje lateral derecho, otra del eje central y una del eje lateral izquierda, como se muestra en la Figura 2-4. y en la Tabla 2-2 se presentan los nombres de cada señal.

Tabla 2-2: Señales de Cámara KB5V

Nombre de la señal	Descripción
Señal 3	Señal eje derecho
Señal 12	Señal línea central
Señal 21	Señal eje izquierda

Esta base de datos está compuesta por descargas que corresponden a la campaña C36 con 151 descargas disruptivas y 995 descargas no disruptivas. Estas señales fueron muestreadas a un 0.2 ms con duración en promedio de 12 s.

### 2.3.3 Base de datos TJ-II

De las campañas realizadas en el Stellarator TJ-II y facilitada por el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), se obtuvo una base de datos con 5 señales de interés.

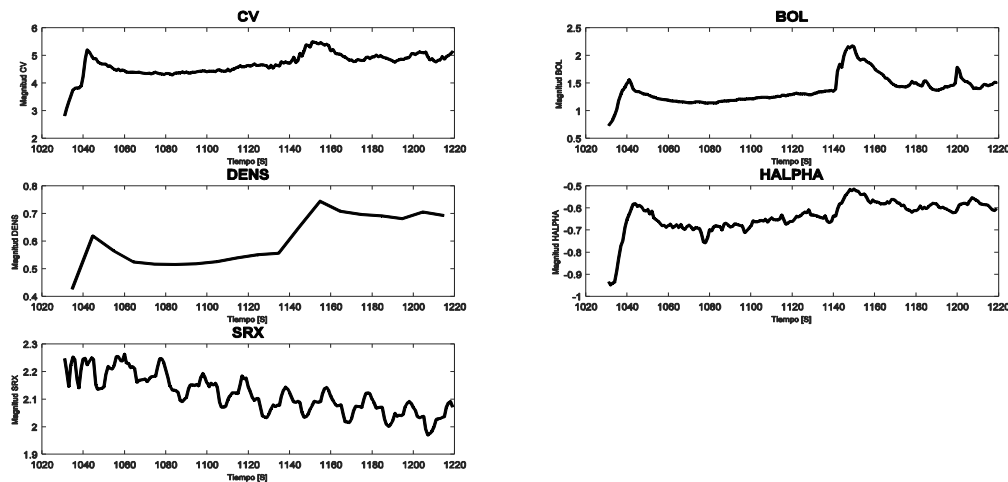


Figura 2-5: Señales de interés de TJ-II

En la Figura 2-5 se observan las señales de interés y en la Tabla 2-3 se presentan los nombres de cada señal.

Tabla 2-3: Señales Base de datos TJ-II

Nombre de la señal	Descripción
1 CV	Señal estroboscópica
2 BOL	Señal bolométrica
3 DENS	Línea media de densidad electrónica
4 HALPHA	Línea de intensidad $H\alpha$
5 SXR	Rayos X suaves

Esta base de datos está compuesta por 68 señales que fueron muestreadas a 0,01ms con un promedio de duración de 20 s.

### 2.4 Motivación en base a la inteligencia artificial

Los dispositivos experimentales de fusión nuclear generan una gran cantidad de datos, como se mencionó anteriormente las bases de datos obtenidas están compuestas por gran cantidad de descargas que a su vez poseen un número relativo de señales que a su vez poseen una cantidad importante de datos, generando bases de datos de Gigas de información relativa al comportamiento del plasma, este gran volumen de información no puede ser totalmente analizado por una persona y se necesitan sistemas automatizados que puedan manejar tal información.

Desde la antigüedad se han imaginado máquinas que piensan y actúan de forma autónoma, ejemplos de ellos son las figuras mitológicas de Galatea, Talos y Pandora, actualmente también se busca el mismo concepto a través de computadores programables, lograr crear una inteligencia artificial que ayude a la investigación científica.

La inteligencia artificial es un campo que crece rápidamente y que ha demostrado resolver rápidamente problemas difíciles para los humanos, la gran complejidad está en lograr que la inteligencia artificial aprenda a desarrollar y describir problemas de forma intuitiva y automáticamente [7]. El aprendizaje automático, es una rama de la inteligencia artificial que permite a las computadoras aprender, en este contexto se refiere a identificar patrones complejos en gran cantidad de datos, y esto lo hace a través de algoritmos que revisan los datos, de esta forma la computadora es capaz de predecir comportamientos futuros. Estos algoritmos pueden descubrir cómo realizar tareas generalizando a partir de ejemplos, esto es muy atractivo en contraste a la construcción de estos programas manualmente, haciéndolo factible y rentable donde la programación manual no lo es, mientras más datos se posean más problemas se pueden abordar y en la última década el uso de aprendizaje automático se ha esparcido rápidamente más allá de la ciencia computacional.

En el tiempo actual disponemos de gigantescas bases de datos con gran cantidad de señales de dispositivos de fusión que pueden ser totalmente analizados por algoritmos de inteligencia artificial, estos pueden hacer uso de estos datos para lograr pronosticar varias series temporales en un periodo determinado de tiempo, gracias a la posibilidad de abstracción de estas inteligencias es posible encontrar patrones, características o comportamientos que pueden describir la dinámica de las señales durante una interrupción para lograr estrategias de evitación y mitigación.

#### 2.4.1 Series de tiempo

Los datos provenientes de los distintos experimentos realizados en los dispositivos de fusión nuclear son una serie de datos listados en un orden temporal, esta serie temporal esta muestreada a un tiempo de muestreo igual para cada dato, es decir, son una secuencia de datos de tiempo discreto. Las series de tiempo son comúnmente usadas en estadísticas, matemáticas financieras,

ingeniería de control, ingeniería de telecomunicaciones, reconocimiento de patrones, procesamiento de señales y en distintos ámbitos de la ciencia donde se involucre mediciones de tiempo.

### 2.4.2 Reconocimiento de patrones

El reconocimiento de patrones un proceso que detecta e identifica estructuras ordenadas en datos, los patrones se obtienen a partir de procesos de segmentación y extracción de características, con el objetivo de obtener información que permita establecer un conjunto de reglas que describan a los datos analizados. El proceso de un sistema de reconocimiento de patrones consta de varias etapas, como se observa en la Figura 2-6, la primera etapa consta de un sensor que obtiene información y datos para representar algún elemento físico, luego la extracción de característica se encarga, a partir de un patrón de representación, de extraer información importante y eliminando aquella que considere irrelevante y finalmente el clasificador que se encargar los patrones de clase desconocida a una categoría.

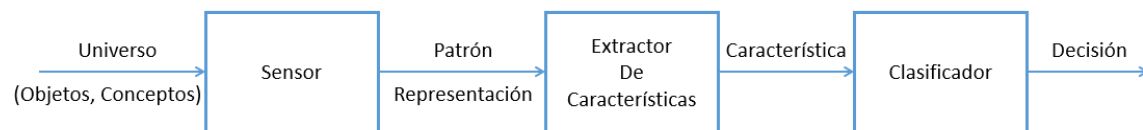


Figura 2-6: Proceso reconocimiento de patrones

Para lograr el modelo más óptimo se debe analizar un cierto número de características para poder clasificar de manera satisfactoria los datos de entrada, es necesario que el modelo se cree a partir de secuencias de datos de entrenamiento o vectores de características, estos sistemas deben tener en cuenta la variabilidad de los datos, como lo son el ruido, cambio de escalas o modificaciones realizadas a las secuencias de entrada.

### 2.4.3 Pronóstico de series temporales

El pronóstico de series temporales es tipo de análisis que genera un modelo que puede predecir valores futuros a partir de valores previamente observados. Los métodos de pronóstico hacen predicciones basadas solamente en patrones históricos que puedan presentar los datos, por lo tanto, realizar pronósticos a futuro de la respuesta de un sistema requiere de modelos que puedan comprender la dinámica de este en un instante de tiempo en particular. Los modelos pueden pronosticar a cierto horizonte de tiempo  $n$  (en el futuro), usando los valores pasados y actuales.

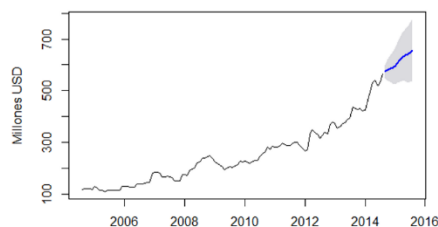


Figura 2-7: Pronóstico para una cartera comercial de un banco (fuente: <http://blog.espol.edu.ec>)

Se aprecia en la Figura 2-7 un ejemplo del pronóstico de series temporales, en azul el pronóstico para el año 2016 para una cartera comercial de un banco.

## 2.5 Redes neuronales artificiales

Las redes neuronales se han utilizado ampliamente como un método de clasificación y de pronóstico en series temporales [7], debido a que las redes neuronales artificiales son una forma de conexionismo, son sistemas computacionales inspirados en la arquitectura biológica de una neurona (Figura 2-8), las redes neuronales artificiales deben reproducir los componentes primarios de las redes neuronales biológicas, es decir, neuronas, axones, etc. En la actualidad se crean estas redes mediante programación en computadoras, esto permite que el uso de computadoras para este campo sea cada vez más potente.



Figura 2-8: Comparación sinapsis neurona biología y artificial (fuente: <https://puentesdigitales.com>)

Manteniendo la esencia de los sistemas neuronales biológicos, una neurona artificial se define de la siguiente manera:

- Posee cierto número de *entradas* (ya sean estos datos originales o bien salidas de otras neuronas de la red). Cada entrada proviene de una *conexión* que tiene cierta fuerza (o peso). Estos pesos se corresponden con la *eficacia sináptica* de una neurona biológica, cada neurona tiene también un cierto *valor umbral*. En el interior de la neurona se suman entradas, y a esta suma se le resta el valor umbral propio de la neurona. Esto genera la *activación de la neurona*, también conocida como *potencial post-sináptico* o *PSP*.
- La señal de activación se procesa mediante una *función de activación*, también conocida como *función de transferencia*, que da lugar a la salida de la neurona.

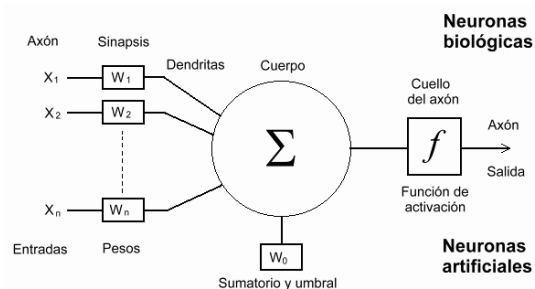


Figura 2-9: Esquema comparación neurona biología y artificial (fuente: <http://www.um.es>)



Típicamente, las neuronas se organizan en capas. Diferentes capas pueden realizar diferentes tipos de transformaciones en sus entradas donde las señales viajan desde la primera (entrada) hasta la última (salida) capa, después de atravesar las capas muchas veces. En las neuronas artificiales con múltiples capas ocultas, las capas iniciales pueden detectar generalizaciones simples, y sus salidas alimentan a capas más profundas que realizan generalizaciones más abstractas.

El objetivo original de la red neuronal artificial es resolver problemas de la misma forma en que lo haría un cerebro humano. A través del tiempo, la atención se ha concentrado en igualar habilidades mentales específicas lo que lleva a desviaciones biológicas, como la propagación inversa (backpropagation) o la transmisión de información en sentido inverso y el ajuste a la red para reflejar esa información.

### 2.5.1 Backpropagation

Del español propagación hacia atrás o inversa, es un algoritmo que se usa para entrenar redes neuronales artificiales, este algoritmo emplea un ciclo de propagación, al momento de ingresar un patrón de entrada a la red, este se propaga desde la primera capa a través de las capas siguientes de la red hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas, estas se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de capa oculta que contribuyan directamente a la salida, estas solo reciben una pequeña fracción de la señal total del error, la cual se basa aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite en todas las capas, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que a medida que se entrena la red completa, las neuronas de las capas ocultas se organizan a sí mismas para aprender a reconocer distintas características del patrón total de entrada. Luego del entrenamiento, cuando se ingrese un patrón ruidoso o incompleto, las neuronas de la capa oculta responderán con una salida activa si la entrada contiene un patrón semejante a alguna característica que las neuronas individuales hayan aprendido durante su entrenamiento.

### 2.5.2 Aprendizaje supervisado

En este algoritmo se usan datos que contienen características a través de un supervisor, y cada una asociada a una etiqueta. Esto hace que el algoritmo pueda aprender a clasificar diferentes problemas en base a esas características. Las tareas que caen dentro del aprendizaje supervisado son el reconocimiento de patrones (clasificación) y la regresión (aproximación de funciones), además se aplica a datos secuenciales. El aprendizaje supervisado se puede considerar aprendizaje con un “maestro”, en el sentido que una función proporciona retroalimentación continua sobre la calidad de las soluciones obtenidas hasta el momento.

### 2.5.3 Aprendizaje no supervisado

En este algoritmo no existe un supervisor que controle el proceso de aprendizaje, se tiene mucha información que contiene muchas características y se aprende usando todas las propiedades de la base de datos, para luego aprender a clasificarlas.

## 2.6 Redes Neuronales Recurrentes

Es una clase de red neuronal artificial donde las conexiones entre las unidades forman un ciclo dirigido (Figura 2-10). La idea detrás de las redes neuronales recurrentes, es hacer uso de información secuencial, en una red neuronal tradicional se supone que todas las entradas (y salidas) son independientes entre sí. Pero para muchas tareas es una mala idea, por ejemplo, si se desea predecir la siguiente palabra dentro de una oración es mejor saber que palabras le precedieron.

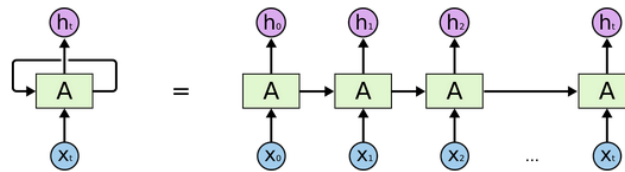


Figura 2-10: Esquema red neuronal recurrente (fuente: <http://colah.github.io>)

Las redes neuronales recurrentes (RNN) se llaman recurrentes porque realizan la misma tarea para cada elemento de la secuencia [8], dependiendo de la salida de los cálculos anteriores. Otra forma de pensar en las redes neuronales recurrentes es que tiene una “memoria” que captura la información sobre lo que se ha calculado anteriormente. En teoría las redes neuronales recurrentes pueden hacer uso de la información en secuencias arbitrariamente largas, pero en la práctica se limitan a mirar hacia atrás unos pocos pasos.

Una red neuronal recurrente es una extensión de una red convencional retroalimentada la cual es capaz de manejar una secuencia de longitud variable en su entrada. La red neuronal recurrente maneja secuencias de longitud variable teniendo un estado oculto recurrente cuya activación depende de estados previos.

Formalmente, dada la secuencia  $\mathbf{x} = (x_1, x_2, \dots, x_T)$ , la RNN actualiza el estado oculto recurrente  $h_t$  cuando

$$\mathbf{h}_t = \begin{cases} 0, & t = 0 \\ \phi(\mathbf{h}_{t-1}, \mathbf{x}_t), & e. o. c \end{cases} \quad (2-1)$$

Donde  $\phi$  es una función no lineal como la composición lógica Sigmoide con una transformación afín. Opcionalmente, la RNN puede tener una salida  $\mathbf{y} = (y_1, y_2, \dots, y_T)$ , la cual puede ser de longitud variable.

Tradicionalmente, la actualización del estado oculto recurrente es implementado como:

$$\mathbf{h}_t = g(W\mathbf{x}_t + U\mathbf{h}_{t-1}), \quad (2-2)$$

Donde  $g$  es una función suave y delimitada, como una función lógica Sigmoide o una función tangente hiperbólica.

Una RNN genera una distribución de probabilidad sobre el siguiente elemento de la secuencia, dado un flujo de estado  $h_t$ , y esto genera un modelo que puede capturar una distribución sobre secuencias de longitud variable y usando un símbolo de salida especial para representar el final de la secuencia.

$$p(x_1, \dots, x_T) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \dots p(x_T|x_1, \dots, x_{T-1}),$$

Donde el último elemento es un valor especial al final de la secuencia.

## 2.7 Deep Learning

En español, Aprendizaje Profundo, es una forma de aprendizaje automático (machine learning) que permite a las computadoras aprender de la experiencia y entender el mundo en términos de conceptos jerárquicos. Debido a que la computadora reúne conocimiento desde la experiencia, no hay necesidad que un operador humano le especifique formalmente todo el conocimiento que ella necesita. Esta jerarquía de conceptos permite a la computadora aprender conceptos complicados al construirlos de conceptos más simples. En una computadora los modelos de aprendizaje profundo pueden alcanzar una precisión de última generación, incluso superando el rendimiento a nivel humano [9]. Estos modelos de aprendizaje automático son entrenados usando gran conjunto de datos etiquetados y arquitecturas de Redes Neuronales que contienen muchas capas. Deep Learning es un algoritmo de aprendizaje automático que usa una estructura en cascada de múltiples capas de unidades de procesamiento no lineal para la extracción de características y transformación. Cada capa sucesiva usa la salida de la capa previa como entrada, este aprendizaje puede ser supervisado o no supervisado y contiene muchos niveles de representación dependiendo de los diferentes niveles de abstracción (niveles de conceptos jerarquizados).

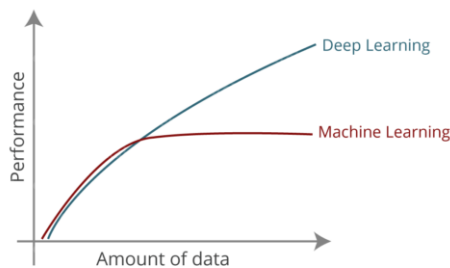


Figura 2-11: Rendimiento Deep Learning (fuente: [www.edureka.co](http://www.edureka.co))

La Figura 2-11 muestra como los algoritmos de Deep learning poseen un mejor rendimiento al aumentar la cantidad de datos que manejan en relación a otros tipos de algoritmos basados en arquitecturas no profundas.

## 2.8 Redes recurrentes Profundas

El proceso en la mayoría de las redes neuronales recurrentes puede ser descompuesto en tres simples bloques de parámetros y correspondiente transformación:

1. Desde la *entrada* al *estado oculto*
2. Desde el *estado oculto* al siguiente *estado oculto*, y
3. Desde el *estado oculto* a la *salida*

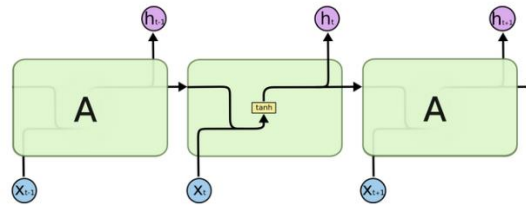


Figura 2-12: Simple red neuronal recurrente (fuente: <http://colah.github.io>)

### 2.8.1 Problema de dependencia a largo plazo

Como las redes neuronales recurrentes hacen uso de información secuencial se supone que todas las entradas (y salidas) son independientes entre sí. Pero para muchas tareas no se obtienen buenos resultados [10] ya que se pierde información a medida que avanza. Otra forma de pensar en las redes neuronales recurrentes es que tiene una “memoria” que captura la información sobre lo que se ha calculado anteriormente. En teoría las redes neuronales recurrentes pueden hacer uso de la información en secuencias arbitrariamente largas, pero en la práctica se limitan a mirar hacia atrás unos pocos pasos perdiendo el contexto de la predicción.

### 2.8.2 Gradiente Desvaneciente

Introducido por Bengio [10] el problema de los gradientes de explosión, se refiere al gran aumento en la norma del gradiente durante el entrenamiento. Tales eventos se deben a la explosión de los componentes a largo plazo, los cuales pueden crecer más exponencialmente que los términos a corto plazo. El problema de gradiente desvaneciente se refiere a lo opuesto del comportamiento descrito, cuando los componentes a largo plazo crecen exponencialmente rápido a la norma 0, haciendo imposible para el modelo aprender entre eventos temporalmente distantes.

## 2.9 Long short-term memory

Las redes de memoria de largo a corto plazo o Long Short-Term Memory (LSTM) es un sistema de aprendizaje profundo y son un tipo especial de red neuronal recurrente que es capaz de lidiar con el problema de la dependencia a largo plazo, estas redes fueron explícitamente diseñadas para recordar información por largos periodos de tiempo. Este tipo de red es ideal para trabajar con datos que vengan en secuencias, ya sea datos discretos o series temporales.

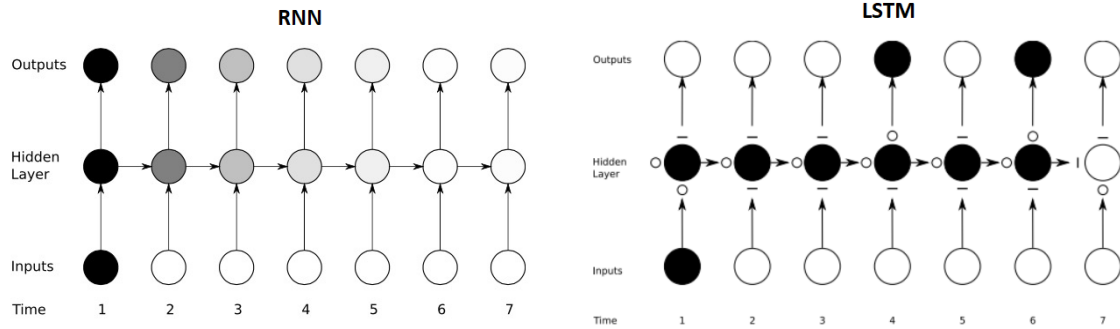


Figura 2-13: Problema gradiente desvaneciente (fuente: <http://eric-yuan.me>)

La unidad de Larga Memoria a Corto Plazo o Long Short-Term Memory (LSTM) fue inicialmente propuesta por Hochreiter y Schmidhuber [11]. Desde entonces, unos mínimos de modificaciones fueron hechas a la unidad original de LSTM.

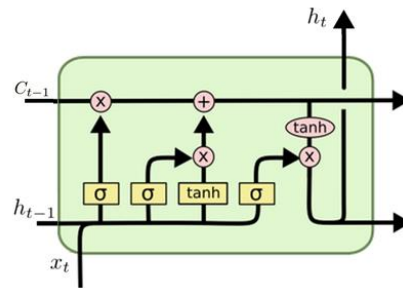


Figura 2-14: Unidad LSTM (fuente: <http://colah.github.io>)

### 2.9.1 Funcionamiento

El primer paso de la red LSTM es decidir qué información va a desechar de la memoria, esta decisión es tomada por la compuerta “Forget Gate”.

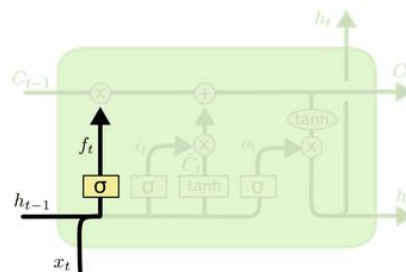


Figura 2-15: Ingreso de datos (fuente: <http://colah.github.io>)

Donde  $h_{t-1}$  y  $x_t$  se concatenan y el resultado será la entrada a la red que conforma la *Forget Gate*  $f_t^j$ , para cada  $j$  – *esimo* elemento de la unidad LSTM, este resultado servirá para decidir si el estado

de la memoria se dejará o se alterará eliminando algún elemento. El alcance en que se olvida la memoria existente se modula mediante la compuerta de olvido  $f_t^j$ .

$$f_t^j = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + V_f \mathbf{c}_{t-1})^j, \quad (2-3)$$

El siguiente paso que realiza la LSTM es decidir qué información nueva almacenará en la memoria, primero en una compuerta llamada "Input Gate"  $i_t$  decide que valores actualizar, luego una pequeña red con  $\tanh$  crea un vector de valores nuevos candidatos,  $\tilde{C}_t$ , y luego se combinan los dos resultados para crear una actualización del estado. El grado en el cual el nuevo contenido de memoria es agregado a la celda de memoria es modulada por la compuerta de entrada  $i_t^j$ . Donde  $V_f$  y  $V_i$  son matrices diagonales.

$$i_t^j = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + V_i \mathbf{c}_{t-1})^j \quad (2-4)$$

$$\tilde{C}_t^j = \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1})^j \quad (2-5)$$

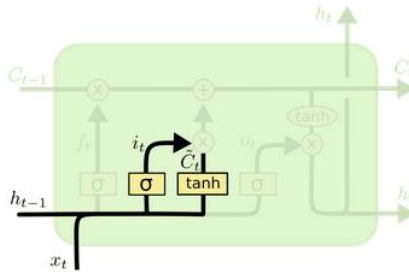


Figura 2-16: Almacenamiento de datos (fuente: <http://colah.github.io>)

Luego es posible actualizar el estado de la memoria cambiando  $C_{t-1}^j$  por el nuevo estado  $\tilde{C}_t^j$ , debido a que las decisiones ya fueron realizadas solo se aplican las operaciones correspondientes, es decir, se multiplica el estado anterior por  $f_t^j$  olvidando una cierta cantidad de información en función de  $f_t^j$ , se suma  $i_t^j * \tilde{C}_t^j$  a la memoria, actualizando esta con nuevos valores que pueden servir a futuro.

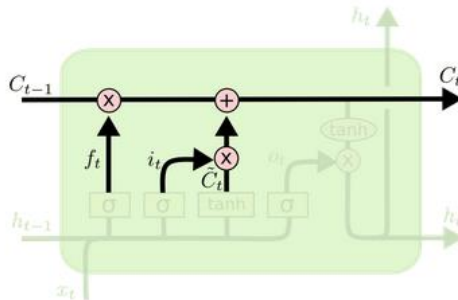


Figura 2-17: Actualización de memoria (fuente: <http://colah.github.io>)

$$C_t^j = f_t^j C_{t-1}^j + i_t^j \tilde{C}_t^j \quad (2-6)$$

Finalmente se decide la salida de la celda LSTM mediante la salida de la red con el sigmoide, que servirá para decidir qué elementos de la memoria se combinarán y un segundo elemento será filtrado de datos desde la memoria  $\tanh$  para “empujar” los valores entre -1 y 1. Estos dos elementos se multiplicarán dando como resultado la nueva salida de la celda.

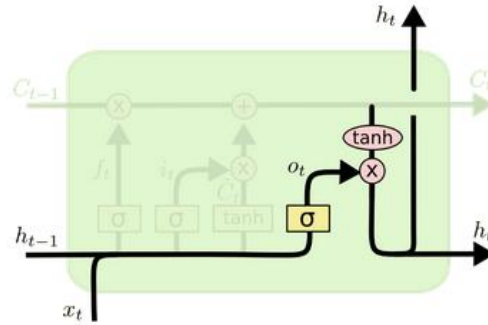


Figura 2-18: Salida a siguiente neurona (fuente: <http://colah.github.io>)

A diferencia de la unidad recurrente que simplemente calcula una suma ponderada de la señal de entrada y aplica una función no lineal, cada  $j$  –esimo elemento de la unidad LSTM mantiene una memoria  $C_t^j$  al tiempo  $t$ . Donde  $o_t^j$  es una compuerta de salida que modula la cantidad de contenido de la memoria que es expuesta. La salida  $h_t^j$ , o activación, de la unidad LSTM es entonces

$$o_t^j = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t)^j \quad (2-7)$$

$$h_t^j = o_t^j \tanh(c_t^j), \quad (2-8)$$

Donde  $\sigma$  es una función lógica Sigmoide y  $V_o$  es una matriz diagonal.

## 2.10 Gated recurrent unit

Las Unidades Recurrentes Bloqueadas o Gated Recurren Unit (GRU) fueron propuestas por Cho [12] para hacer que cada unidad recurrente capture de forma adaptativa dependencias de diferentes escalas de tiempo. De forma similar a la unidad LSTM, la GRU tiene unidades de compuerta que modulan el flujo de información dentro de la unidad. Estas tienen menos parámetros que la LSTM, y que carecen de una puerta de salida.

### 2.10.1 Funcionamiento

El primer paso que realiza la red GRU es la activación  $h_t^j$  de la GRU al tiempo  $t$ , esta es una interpolación lineal entre la activación previa  $h_{t-1}^j$  y el candidato de activación  $\tilde{h}_t^j$ :

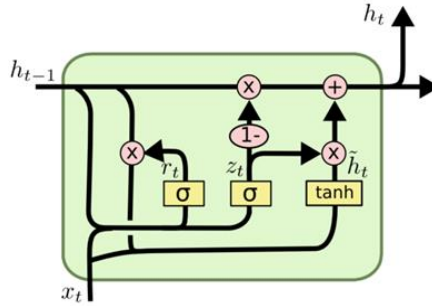


Figura 2-19: Unidad GRU (fuente: <http://colah.github.io>)

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j \quad (2-9)$$

Donde la actualización de la compuerta  $z_t^j$  decide cuanto actualiza la unidad de activación. La actualización de la compuerta es calculada por:

$$z_t^j = \sigma(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1})^j \quad (2-10)$$

Este procedimiento de tomar una suma lineal entre los estados existentes y el nuevo estado calculado es similar al de la unidad LSTM. La GRU, en cambio, no tiene ningún mecanismo para controlar el grado al cual es expuesto el estado, pero expone todo el estado a la vez.

El candidato de activación  $\tilde{h}_t^j$  es calculado de manera similar a la unidad recurrente tradicional

$$\tilde{h}_t^j = \tanh(W \mathbf{x}_t + U(\mathbf{r}_t \odot \mathbf{h}_{t-1}))^j \quad (2-11)$$

Donde  $\mathbf{r}_t$  es un conjunto de puertas de reinicio y  $\odot$  es una multiplicación de elementos. Cuando está apagado ( $r_t^j$  cercano a 0) la compuerta de reinicio efectivamente hace que la unidad actúe como si estuviera leyendo el primer símbolo de la secuencia de entrada, permitiéndole que olvide la computación del estado anterior. La compuerta de reinicio  $r_t^j$  es calculada de manera similar a la compuerta de actualización:

$$r_t^j = \sigma(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1})^j \quad (2-12)$$



## 3 Implementación de algoritmos

En el presente capítulo, se detalla el proceso de construcción de las distintas redes neuronales recurrentes utilizadas en este proyecto. En primer lugar, cómo funciona el proceso del pronóstico de series temporales, las medidas de error y exactitud y la explicación de la construcción de la red neuronal recurrente tipo LSTM y tipo GRU, en segundo lugar, se explica cómo se diseñó algoritmos para clasificación de descargas disruptivas y no disruptivas.

### 3.1 Pronóstico de series temporales

Los problemas de pronóstico de series de tiempo son un tipo difícil de problema de modelado predictivo, las series de tiempo agregan la complejidad de la dependencia del tiempo en la secuencia en la variable de entrada, es por ello que las redes neuronales recurrentes fueron propuestas para este proyecto ya que pueden manejar la dependencia en la secuencia.

Las redes neuronales recurrentes tienen conexiones que poseen ciclos repetitivos, agregando realimentación y memoria a la red durante el tiempo. Esta memoria permite a este tipo de red aprender y generalizar a través de secuencias de entradas en lugar de patrones individuales. Un poderoso tipo de redes neuronales recurrentes son las llamadas Long Short-Term Memory Network y estas han mostrado ser particularmente efectivas cuando son apiladas en una configuración profunda, logrando resultados de vanguardia en una amplia gama de problemas.

La principal característica de las redes neuronales recurrentes de tipo LSTM o GRU es que la dependencia en los datos de entrada puede ser aprendido, y además no es necesario especificar un conjunto fijo de observaciones retardadas, junto con esto queda implícito, para la red, que se puede aprender una dependencia temporal que varía según las circunstancias, como lo son las dinámicas presenten en las señales de fusión nuclear. Gracias a las capacidades descritas de la red neuronal recurrente tipo LSTM y GRU, se ha decidido probar esta red para pronosticar el comportamiento futuro de los datos de una señal. Se desea entonces conocer si la red es capaz de predecir el comportamiento de cada una de las señales necesarias para el pronóstico a distintos horizontes de tiempo.

### 3.1.1 Rendimiento del pronóstico

Para la medición del rendimiento del pronóstico se utilizarán dos criterios, el promedio del error absoluto porcentual (MAPE) y la precisión. El error MAPE es un criterio de exactitud, es decir, apunta a medir cuan cercanos son los valores pronosticados con respecto a sus valores reales y la precisión apunta a cuan repetible y reproducible es el pronóstico. La Figura 3-2 muestra con un ejemplo didáctico la diferencia entre exactitud y precisión.

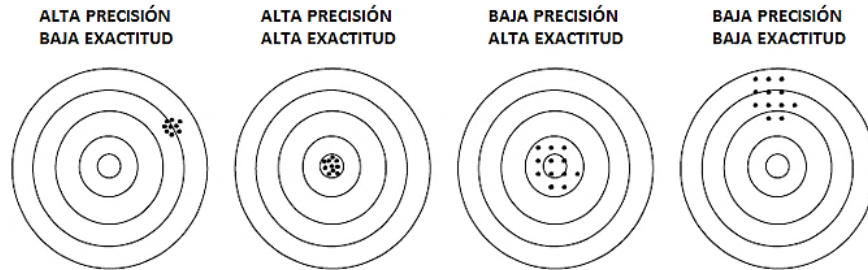


Figura 3-1: Diferencia entre exactitud y precisión (fuente: <https://ingenieriaelectronica.org>)

La ecuación 3-1 y la ecuación 3-3 muestra cómo se realiza el cálculo del error MAPE y la precisión respectivamente, donde  $n$  es el número de muestras de los datos de la serie de tiempo,  $y_i$  denota el valor real de la señal analizada e  $\hat{y}_i$  denota el valor pronosticado por la red.

$$MAPE\% = 100 * \frac{1}{n} * \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3-1)$$

$$Exactitud\% = 100 - MAPE\% \quad (3-2)$$

$$\sigma\% = 100 * \sqrt{\frac{1}{n} * \sum_{i=1}^n \left( \left| \frac{y_i - \hat{y}_i}{y_i} \right| - MAPE \right)^2} \quad (3-3)$$

$$Precisión\% = 100 - \sigma\% \quad (3-4)$$

### 3.2 Clasificación de señales

La clasificación de señales disruptivas y señales no disruptivas es un problema complejo de resolver, la dificultad reside en a la aleatoriedad de las señales provenientes de las campañas realizadas, la clasificación de series temporales tiene su base en encontrar patrones en la secuencia de datos de entrada, para ello una serie temporal debe ser analizada ventana a ventana durante un periodo tiempo que dure la secuencia, el objetivo es clasificar una señal como disruptiva o no disruptiva si se encuentran patrones precursores que indiquen que a cierto tiempo  $t$  ocurrirá un evento disruptivo en el dispositivo. Debido a la capacidad de las redes neuronales recurrentes ya que pueden manejar la dependencia en la secuencia y, de ser posible, encontrar tiempos con suficientemente antelación para generar técnicas de mitigación y

evitación que logren ayudar a los nuevos dispositivos de fusión nuclear en los cuales se sigue investigando cada día.

### 3.2.1 Cálculo de centroides

Como las señales vienen en secuencia de datos estas están divididas en ventanas de tiempo, llamaremos centroide a la característica compuesta por un dato impar (ventana impar) y un dato par (ventana par) a partir del valor de la secuencia donde se encuentre el tiempo disruptivo, además se posee un criterio para la construcción de centroides para señales disruptivas, en él se discute que la ventana impar de cualquier señal disruptiva debe ser mayor (en magnitud) que la ventana par que la antecede, con esto se generarán centroides por cada ventana impar-par de la señal y en particular se calculará un centroide promedio de todos los centroides, para señales disruptivas como para señales no disruptivas. De esta forma será posible calcular distancias euclídeas entre los centroides compuestos por las ventanas impar-par y el centroide promedio disruptivo y centroide promedio no disruptivo. Se observará el comportamiento cada dos ventanas (impar-par) y se verificará su distancia euclídea con respecto a ambos centroides promedio y con ello se generará una alarma cuando las ventanas analizadas posean una distancia euclídea más cercana al centroide promedio disruptivo que al centroide promedio no disruptivo.

La Distancia euclídea es la distancia “ordinaria” entre dos puntos de un espacio euclídeo  $n$  – *dimensional*. Donde Sean  $P = (p_1 + p_2 + \dots + p_n)$  y  $Q = (q_1 + q_2 + \dots + q_n)$

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3-5)$$

### 3.2.2 Análisis de peaking factor

Entenderemos este factor a aquel que se obtiene al dividir una señal completa, dato a dato, por otra, generando así una forma de onda nueva, la hipótesis del uso de este factor es obtener cerca del tiempo de la disrupción una tendencia de puntos con pendiente positiva, este precursor sería de mucha ayuda para lograr una estrategia de evitación. Para el desarrollo de peaking factor se han utilizado las señales bolométricas y con ella el uso del valor RMS, que es una medida estadística de la magnitud de una variable, esta variable puede tomar valores positivos y negativos, en nuestro caso estamos interesados en obtener un promedio que no recoja los efectos del signo de la variable, para ello se elevan al cuadrado las observaciones para obtener su media aritmética y en extraer la raíz cuadrada de nuestras variables para volver a la unidad de medida original, con ese valor se obtendrán de igual manera centroides (ver sección 3.2.1). Por lo tanto, la media cuadrática para una colección de  $N$  valores  $\{x_1, x_2, \dots, x_N\}$  de una variable discreta  $x$  viene dada por la siguiente fórmula:

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_N^2}{N}} \quad (3-6)$$

### 3.2.3 Alarmas para mitigación y evitación

Los criterios que se ocuparán para validar cada resultado obtenido se harán a partir de alarmas, es decir, el tiempo donde el clasificador etiqueta un dato de la secuencia de la serie temporal como disruptiva o no disruptiva. Para activar cada tipo de alarma, es necesario un tiempo que varía según el tiempo de activación de la alarma, tendremos entonces distintos tipos de alarmas las cuales son las siguientes: Alarma Válida (VA) la cual tendrá valores sobre 30 ms y bajo 1 s, Alarma Prematura (PA) la cual tendrá valores sobre 1 s, Alarma Tardía (TD) la cual tendrá valores mayores de 0 s y menores a 30 ms, la suma de VA, PA y TA nos proporcionara la Tasa de Acierto (SR) y junto con Alarma Perdida (MS) aquellas no detectadas, obtendremos el porcentaje total de elementos testeados.

$$VA = 30 \text{ ms} < WT < 1 \text{ ms} \quad (3-7)$$

$$PA = WT > 1 \text{ s} \quad (3-8)$$

$$TD = 0 < WT < 30 \text{ ms} \quad (3-9)$$

$$SR = VA + PA + TD \quad (3-10)$$

$$MS = 1 - SR = WT \leq 0 \text{ s} \quad (3-11)$$

También a partir de los tipos de alarma en el testeo de descargas no disruptivas tendremos las alarmas Falsa Alarma (FA) cuando se detecte una señal no disruptiva como disruptiva.

$$FA = \text{Señal no disruptiva detectada como Disruptiva} \quad (3-12)$$

## 3.3 Entorno de trabajo

Para el diseño y construcción del modelo de red neuronal recurrente de tipo LSTM y tipo GRU se ha utilizado el lenguaje de programación Python junto con la biblioteca TensorFlow y la Interfaz de programación de aplicaciones (API) Keras, se detalla a continuación el por qué el uso de estos.

### 3.3.1 Tensorflow y Keras

Es una biblioteca de código abierto, desarrollado y liberado por Google, que se basa en un sistema de redes neuronales. Esto significa que puede relacionar varios datos en una red simultáneamente, de la misma forma que lo hace un cerebro humano. Por ejemplo, puede reconocer varias palabras del alfabeto porque relaciona letras y fonemas. Otro caso es el de imágenes y textos que se pueden relacionar entre si rápidamente gracias a la capacidad de asociación del sistema de redes neuronales (recurrentes). En el programa, se almacenan todas las pruebas y experimentos que se realizaron para el desarrollo de programas y aplicaciones.

La API Keras es una librería de Aprendizaje Profundo (Deep Learning) de Python, es una interfaz de programación de aplicaciones de alto nivel, escrita en Python y capaz de ejecutarse en TensorFlow, CNTK o Theano. Nos proporciona la posibilidad de trabajar con Redes Neuronales Recurrentes gracias a sus librerías, donde el modelo LSTM y modelo GRU ya vienen

implementados, permitiendo al usuario ajustar los parámetros acordes a sus necesidades. Keras fue desarrollado con un enfoque que permite la experimentación rápida.



Figura 3-2: Keras interfaz simplificada de TensorFlow (fuente: <https://blog.keras.io>)

### 3.4 Diseño y construcción de la red para pronóstico

Para el diseño de una red neuronal recurrente de tipo LSTM/GRU se utilizó el siguiente procedimiento:

1. Selección de datos de entrada
2. Preprocesamiento de los datos de entrada
3. Creación base de datos de entrenamiento y testeo
4. Diseño y construcción de red neuronal
5. Implementación de la red neuronal
6. Resultados del pronóstico
7. Criterios de evaluación

El proceso es el siguiente, la primera instancia es la adquisición de datos, donde se tiene dos bases de datos, la base de datos JET con 201 descargas disruptivas y 1036 descargas no disruptivas con 7 señales cada una (señales 1 a 7), además de 2 señales con 151 descargas disruptivas y 995 señales no disruptivas con dos señales cada una (señales 8 a 9), como lo muestra la Tabla 2-1 en el Capítulo 2, junta a ella se tiene la base de datos TJ-II con 68 descargas, cada una con 5 señales a analizar como lo muestra la Tabla 2-3 del Capítulo 2.

La base de datos de JET está compuesta por 7 señales por descarga y están muestreadas cada una a 0.001 ms por lo que fueron remuestreadas a 32 ms utilizando el valor promedio de las ventanas de tiempo, la base de datos de JET que está compuesta por 2 señales por descarga están muestreadas cada una a 0.025 ms por lo que fueron remuestreadas a 32 ms utilizando el valor promedio de las ventanas de tiempo.

La base de datos de TJ-II está compuesta de 5 señales por descarga y están muestreadas a 0.01 ms, por lo que fueron remuestreadas a 1 s utilizando el valor promedio de las ventanas de tiempo.

Todas estas señales de cada base de datos fueron normalizadas (fueron representadas sus datos entre valores de 0 a 1) antes de ingresar a la red neuronal. Todas las bases de datos mencionadas anteriormente se dividieron en dos tipos: base de datos entrenamiento con el 60% de la señales y base de datos de testeo con el 40% restante, estas bases de datos para poder ser ingresada a una red neuronal recurrente deben tener retardo, el cual fue un retardo de 2 s para la base de datos de JET y un retardo de 2 s para la base de datos de TJ-II.

### 3.4.1 Diseño e implementación de la red

El diseño de la red neuronal tipo LSTM y tipo GRU para la base de datos JET se concibió de la siguiente manera: 7 neuronas de entrada, 20 neuronas de estado oculto, 1 neurona de salida, 200 épocas de entrenamiento, función de activación sigmoide (por defecto), dropout recurrente de 0.1, una batch size de 2, error de red RMS y optimizador RMSProp. Para el pronóstico en base de datos TJ-II diseño la red neuronal tipo LSTM y tipo GRU con: 5 neuronas de entrada, 20 neuronas de estado oculto, 1 neurona de salida, 200 épocas de entrenamiento, función de activación sigmoide (por defecto), dropout recurrente de 0.1, una batch size de 2, error de red RMS y optimizador RMSProp como lo muestra el Listado 3-1.

Listado 3-1: Implementación red LSTM/GRU Pronóstico

```

1  From keras.layers import LSTM, GRU
2  model_LSTM = Sequential ()
3  model_LSTM.add (LSTM (20, recurrent_dropout=0.1, input_dim=lag)) (inputs)
4  model_LSTM.add (Dense (1))
5  model_LSTM.compile(loss='mean_squared_error', optimizer='RMSProp')
6  model_LSTM.fit (trainX, trainY, nb_epoch=200, batch_size=2)
7  model_GRU = Sequential ()
8  model_GRU.add (GRU (20, recurrent_dropout=0.1, input_dim=look_back))
9  model_GRU.add (Dense (1))
10 model_GRU.compile(loss='mean_squared_error', optimizer='RMSProp')
11 model_GRU.fit (trainX, trainY, nb_epoch=200, batch_size=2)
12 #Pronóstico
13 testPredict_LSTM = model_LSTM.predict(testX)
14 testPredict_GRU = model_GRU.predict(testX)
15
16

```

En el Listado 3-1, tenemos la importación del modelo a partir de `keras.layers`, el modelo LSTM (`model_LSTM.add` y `model_LSTM.fit`), y el modelo GRU (`model_GRU.add` y `model_GRU.fit`) y los pronósticos en para LSTM (`model_LSTM.predict`) y GRU (`model_GRU.predict`).

Se implementa la red descrita y se obtienen los resultados del pronóstico que son evaluados según los criterios descritos en la sección 3.1.1, los cuales serán mostrados y analizados en el Capítulo 4.

## 3.5 Diseño y construcción de la red para clasificación

Para el diseño de una red neuronal recurrente de tipo LSTM/GRU se utilizó el siguiente procedimiento:

1. Selección de datos de entrada
2. Preprocesamiento de los datos de entrada
3. Creación base de datos de entrenamiento y testeo
4. Selección de etiquetas para señales
5. Diseño y construcción de red neuronal
6. Implementación de la red neuronal
7. Resultados de la clasificación
8. Verificación de alarmas
9. Tasas de acierto

El proceso es el siguiente, la primera instancia es la adquisición de datos, donde se tiene tres bases de datos, la base de datos JET con 201 descargas disruptivas y 1036 descargas no disruptivas con 7 señales cada una (señales 1 a 7 de la Tabla 2-1) de la cual solo se utilizará la señal 2 “Mode Lock”, la base de datos JET con 2 señales con 151 descargas disruptivas y 995 señales no disruptivas con 2 señales cada una (señales 8 a 9 de la Tabla 2-1) y la base de datos de las señales bolométricas con 151 descargas disruptivas y 995 señales no disruptivas con 3 señales cada una (ver Tabla 2-2), como se muestra en el Capítulo 2.

De la base de datos de JET que está compuesta por 7 señales por descarga, solo se utilizará la señal “Mode Lock”, la cual esta muestreada a 0.001 ms por lo que fue remuestreada a 32 ms utilizando el valor promedio de las ventanas de tiempo. De la base de datos de JET que está compuesta por 2 señales por descarga, están muestreadas a 0.025 ms por lo que fueron remuestreadas a 32 ms utilizando el valor promedio de las ventanas de tiempo. De la base de datos de JET con señales bolométricas que están muestreadas a 0.2 ms fueron remuestreadas 32 ms utilizando el valor promedio de las ventanas de tiempo.

Todas estas señales de cada base de datos fueron normalizadas (fueron representadas sus datos entre valores de 0 a 1) antes de ingresar a la red neuronal. Para el cálculo de centroides y análisis de peaking factor se utilizaron la señal Mode Lock, la señal evolución temporal de la afluencia del tungsteno y la señal energía diamagnética, para este algoritmo se dividió la base de datos en 50% de los datos para entrenamiento y 50% de los datos para testeo.

Las bases de datos para clasificación se dividieron en dos tipos: entrenamiento con el 70% de la base de datos y testeo con el 30% restante de la base de datos, las bases de datos para poder ser ingresadas a una red neuronal recurrente deben tener retardo, el cual fue un retardo de 2 s para la base de datos de JET. La etiqueta para señales se realizó de la siguiente forma. para las descargas disruptivas se seleccionó aquellas que están etiquetadas como descargas disruptivas no intencionadas (etiqueta = 1) y para las descargas no disruptivas aquellas etiquetas como seguras (etiqueta = 0).

Para realizar la clasificación se implementó la siguiente etiqueta, para las señales de descargas disruptivas, en donde se busca el tiempo donde se generó la disrupción (tiempo disruptivo) y las 50 ventanas anterior (aproximadamente 1.6 s) a este se etiquetan con etiqueta “1” y las ventanas restantes a estas 50 ventanas son etiquetadas con etiqueta “0” (total de la descarga). Para las señales de descargas no disruptivas son etiquetadas en todas sus ventanas con etiqueta “0” ya que no poseen comportamiento disruptivo.

#### 3.5.1 Diseño e implementación de la red

El diseño de la red neuronal tipo LSTM y tipo GRU se concibió de la siguiente manera: los datos de entrenamiento con retardo ingresan a la red LSTM/GRU, con las siguientes características: Neuronas de Entrada: variable dependiendo de la señal a clasificar, Neuronas Ocultas: 128, Neuronas de Salida: 2, Error: Mean squared error, Optimizer: RMSProp, Dropout recurrente: 0,1, Épocas: 100 y Batch: variable dependiendo de la señal a clasificar.

Listado 3-2: Implementación red LSTM/GRU Clasificación

```

1  From keras.layers import LSTM, GRU
2  inputs, epochs = Input([None, 1]), 100
3  model_lstm = LSTM(128, return_sequences=True)(inputs)
4  model_lstm = LSTM(128, recurrent_dropout=.1)(model_lstm)
5  out_lstm = Dense(2, activation='softmax')
6  model_gru = GRU(128, return_sequences=True)(inputs)
7  model_gru = GRU(128, recurrent_dropout=.1)(model_gru)
8  out_gru = Dense(2, activation='softmax')(model_out_gru)
9  model_lstm = Model(inputs, out_lstm)
10 model_gru = Model(inputs, out_gru)
11 model.compile(optimizer='RMSProp', loss='categorical_crossentropy')
12 model.fit(Train, Label, batch_size=np.max([16, len(Label)//3]), ...
13         epochs=epoch+1, initial_epoch=epoch)
14
15 Predict_test_d[id_test], Predict_test_s[id_test]=model_lstm.predict(Test), ...
16 model_gru.predict(Test)

```

En el Listado 3-2 se observan los modelos de LSTM y GRU (`model_lstm` y `model_gru`) además de la predicción de clasificación para LSTM y GRU (`model_lstm.predict` y `model_gru.predict`).

De esta forma el aprendizaje de todas las señales disruptivas conlleva a la validación por señal, es decir, se realiza verificando ventana a ventana de la señal de validación, en teoría la red aprende ventana a ventana como es el comportamiento de una señal disruptiva y lo verifica con la señal de testeo ventana a ventana hasta que se encuentre 95% de que la ventana es disruptiva. De igual manera se verifican las señales no disruptivas para obtener falsos positivos. Se implementa la red descrita y se obtienen los resultados del pronóstico que son evaluados según los criterios descritos en la sección 3.2.1, sección 3.2.2 y la sección 3.2.3, los cuales serán mostrados y analizados en el Capítulo 4.



## 4 Resultados Obtenidos

En el presente capítulo se muestran y comentan los resultados obtenidos a partir de la implementación de las redes neuronales recurrentes de arquitectura Long-Short Term Memory (LSTM) y Gated Recurrent Unit (GRU). Siendo este el capítulo más importante de este proyecto de titulación, este capítulo recopila todo el trabajo realizado durante Seminario de Proyecto y Proyecto de Titulación, los resultados obtenidos están constituidos de la siguiente manera, se presentan tablas de resultados, gráficas de resultados y comentarios.

La primera parte de los resultados conlleva las primeras pruebas realizadas para el pronóstico de series temporales provenientes de las bases de datos de fusión nuclear. En primer lugar, se comentarán los resultados obtenidos en la base de datos JET para el pronóstico de series temporales a una ventana adelante, estos resultados nos proporcionarán la motivación para utilizar otra base de datos. Una segunda parte está constituida por los resultados obtenidos para el pronóstico a una y cinco ventanas para la base de datos proveniente de TJ-II, principal objetivo de este proyecto.

En tercer lugar, se mostrarán los resultados obtenidos de las pruebas para encontrar características en la base de datos de JET en base al Cálculo de Centroides y Peaking Factor, para finalmente gracias a estas características utilizar las redes LSTM y GRU para realizar Clasificación de señales disruptivas y no disruptivas gracias a la abstracción de las redes neuronales.

Todos los comentarios y conclusiones para este trabajo se recopilan en el último capítulo de este informe.

## 4.1 Pronóstico base JET

Para el pronóstico de series de tiempo a una ventana de tiempo realizado en la base de datos de JET se obtuvieron los siguientes resultados:

Tabla 4-1: Tabla de resultados pronóstico a una ventana de tiempo red LSTM.

Nombre de la señal	Disruptiva Exactitud %	No disruptiva Exactitud %
1 corriente de plasma	99,21	99,98
2 amplitud de “Mode Lock”	98,48	98,56
3 inductancia interna del plasma	99,38	99,01
4 densidad	99,40	99,11
5 derivada de la energía diamagnética	43,04	76,33
6 potencia radiada	98,10	99,06
7 potencia total de entrada	98,85	99,26

En la Tabla 4-1 se muestran los resultados de la Exactitud obtenida a partir del error MAPE, para la red neuronal recurrente tipo LSTM a una ventana de pronóstico, en dos tipos de señales, para señales disruptivas y para señales no disruptivas, en esta tabla se observa que para descargas con señales disruptivas el mejor resultado se obtuvo en la señal de densidad con un 99,40% de exactitud con respecto a su valor real, y el peor resultado se obtuvo en la señal derivada de la energía diamagnética con un 43,04% de exactitud con respecto a su valor real. En las señales de descargas no disruptivas el mejor resultado se obtuvo en la señal corriente de plasma con un de exactitud de 99,98% con respecto a su valor y el peor resultado lo obtuvo la señal derivada de la energía diamagnética con una de exactitud de 76,33% con respecto a su valor real.

Tabla 4-2: Tabla de resultados pronóstico a una ventana de tiempo red GRU.

Nombre de la señal	Disruptiva Exactitud %	No disruptiva Exactitud %
1 corriente de plasma	99,91	99,99
2 amplitud de “Mode Lock”	98,03	99,83
3 inductancia interna del plasma	98,25	99,53
4 densidad	99,64	99,71
5 derivada de la energía diamagnética	14,54	87,07
6 potencia radiada	98,92	98,88
7 potencia total de entrada	99,76	99,27

En la Tabla 4-2 se muestran los resultados de la Exactitud obtenida a partir del error MAPE, para la red neuronal recurrente tipo GRU a una ventana de pronóstico, en dos tipos de señales, para señales disruptivas y para señales no disruptivas, en esta tabla se observa que para descargas con señales disruptivas el mejor resultado se obtuvo en la señal corriente de plasma con un 99,91% de exactitud con respecto a su valor real, y el peor resultado se obtuvo en la señal derivada de la energía diamagnética con un 14,54% de exactitud con respecto a su valor real. En las señales de descargas no disruptivas el mejor resultado se obtuvo en la señal corriente de plasma con un

99,99% de exactitud con respecto a su valor y el peor resultado lo obtuvo la señal derivada de la energía diamagnética con un 87,07% de exactitud con respecto a su valor real.

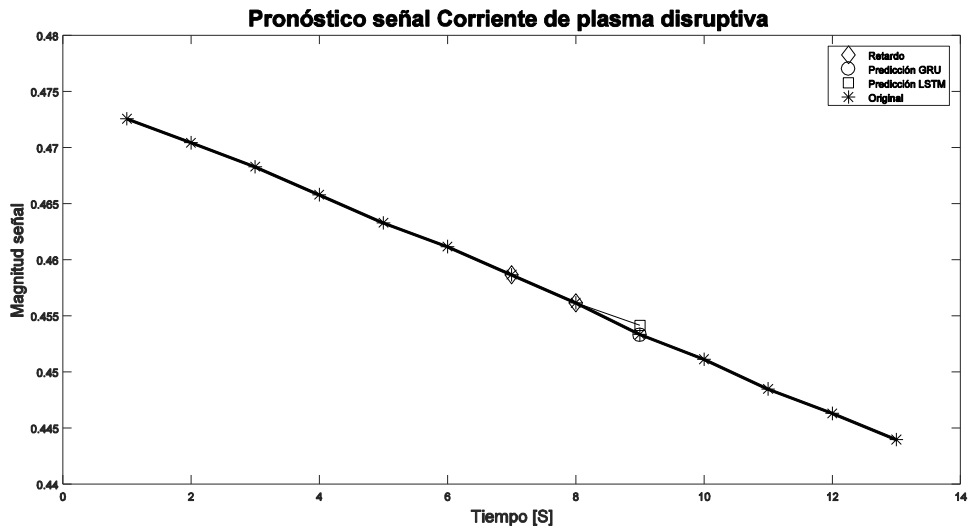


Figura 4-1: Señal Corriente de plasma disruptiva a una ventana de predicción

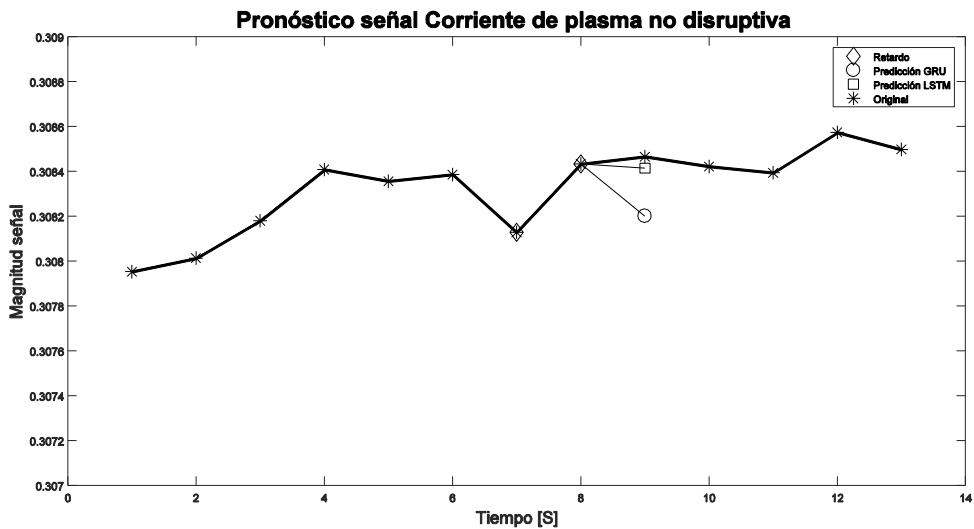


Figura 4-2: Señal Corriente de plasma no disruptiva a una ventana de predicción

Se muestra en la Figura 4-1 el pronóstico a una ventana deslizante para la señal Corriente de plasma disruptiva, se puede observar como en la señal disruptiva el valor pronosticado se acerca bastante al valor real, en la red GRU, no obstante, en la Figura 4-2 la señal Corriente de plasma no disruptiva el valor pronosticado por la red LSTM tiene el menor error con respecto a su valor real.

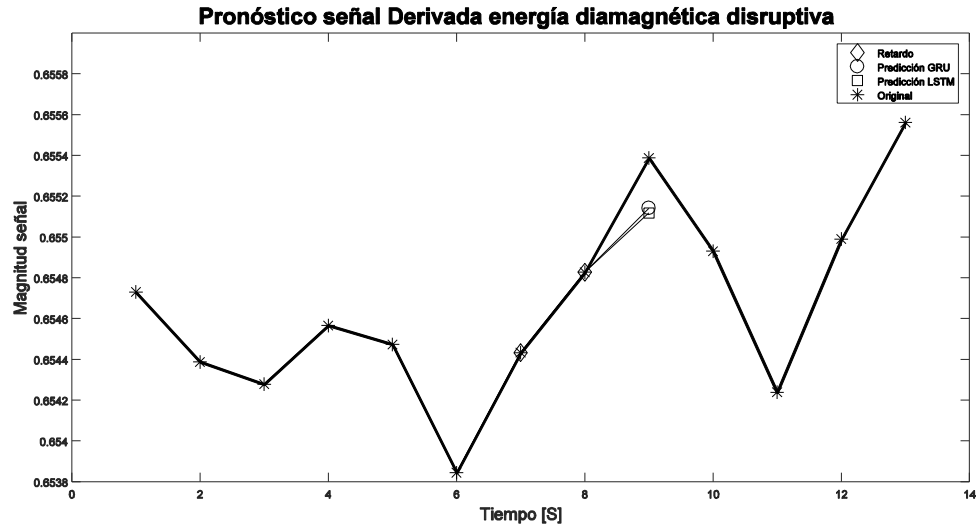


Figura 4-3: Señal Derivada de la energía diamagnética disruptiva a una ventana de predicción

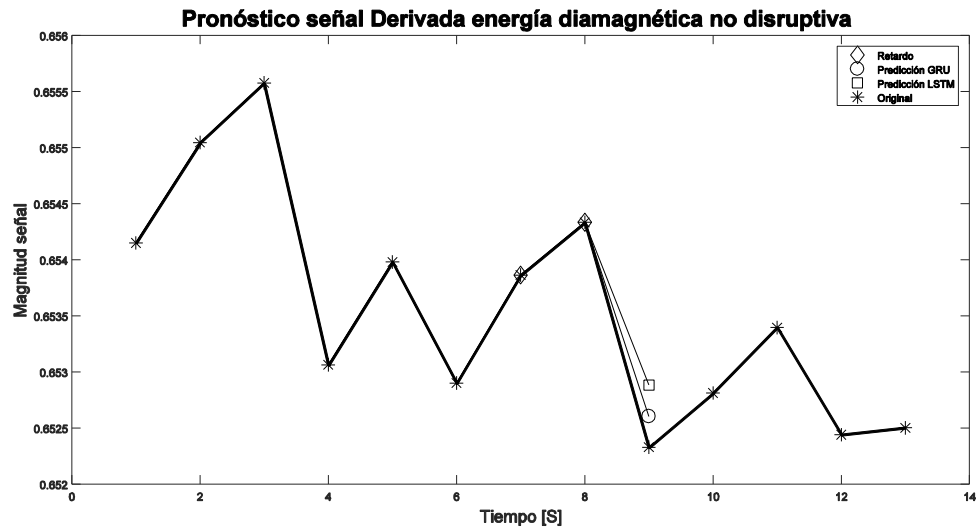


Figura 4-4: Señal Derivada de la energía diamagnética no disruptiva a una ventana de predicción

Se muestra en la Figura 4-3 el pronóstico a una ventana deslizante para la señal Derivada de la energía diamagnética disruptiva, se puede observar como en la señal disruptiva el valor pronosticado por la red GRU se acerca bastante al valor real, igualmente con la Figura 4-4 donde la señal Derivada de la energía diamagnética no disruptiva el valor pronosticado por la red GRU tiene el menor error con respecto a su valor real.

Con este resultado y desempeño se utilizó estas arquitecturas de la red neuronal recurrente para ser aplicada a la base de datos correspondientes al dispositivo experimental de fusión nuclear TJ-II con el objetivo de realizar pronóstico en sus cinco señales, ya que estas señales son de interés para verificar cómo se comporta el modelo desarrollado anteriormente y obtener resultados comparables u óptimos en comparación a esta base de datos.

## 4.2 Pronóstico base TJ-II

Para el pronóstico de series de tiempo con la base de datos TJ-II, el cual es el principal objetivo de este proyecto [13]. Los resultados de la Tabla 4-3 y Tabla 4-4 muestran el desempeño de la red LSTM y red GRU en base a los dos criterios descritos en el capítulo 3, Exactitud y Precisión, a diferencia de los resultados comentados anteriormente, estos resultados poseen pronóstico a +1, +3 y +5 ventanas adelante. A continuación, se muestran los resultados obtenidos:

Tabla 4-3: Tabla de resultados pronóstico a una, tres y cinco ventanas de tiempo red LSTM

Nombre señal	Exactitud GRU %			Precisión GRU %		
	+1	+3	+5	+1	+3	+5
1 CV	96,2	96,6	96,5	95,1	94,6	94,5
2 BOL	96,9	97,0	96,8	96,2	95,5	95,4
3 DENS	96,5	96,0	96,8	96,5	95,5	94,6
4 HALPHA	96,9	96,8	96,7	96,6	95,6	95,7
5 SXR	97,3	97,3	97,3	97,1	96,5	96,2

En la Tabla 4-3 se muestran los resultados de la Exactitud y Precisión obtenida a partir del error MAPE, para la red neuronal recurrente tipo LSTM a diferentes ventanas de pronóstico, en esta tabla se observa que el mejor resultado en exactitud para +1 ventana adelante se obtuvo en la señal SXR con un 97,3%, el mejor resultado en exactitud para +3 ventanas adelante se obtuvo en la señal SXR con un 97,3% y finalmente el mejor resultado en exactitud para +5 ventanas adelante se obtuvo en la señal SXR con un 97,3%. También en esta tabla se observa que el mejor resultado en precisión para +1 ventana adelante se obtuvo en la señal SXR con un 97,1%, el mejor resultado en precisión para +3 ventanas adelante se obtuvo en la señal SXR con un 96,5% y finalmente el mejor resultado en precisión para +5 ventanas adelante se obtuvo en la señal SXR con un 96,2%.

Tabla 4-4: Tabla de resultados pronóstico a una, tres y cinco ventanas de tiempo red GRU.

Nombre señal	Exactitud GRU %			Precisión GRU %		
	+1	+3	+5	+1	+3	+5
1 CV	96,3	96,6	96,6	96,3	95,6	95,6
2 BOL	96,9	97,1	96,8	96,5	95,8	95,7
3 DENS	96,9	96,5	96,0	97,3	96,4	95,7
4 HALPHA	96,8	96,8	96,7	97,0	96,5	95,9
5 SXR	97,4	97,5	97,5	97,3	96,7	96,7

En la Tabla 4-4 se muestran los resultados de la Exactitud y Precisión obtenida a partir del error MAPE, para la red neuronal recurrente tipo LSTM a diferentes ventanas de pronóstico, en esta tabla se observa que el mejor resultado en exactitud para +1 ventana adelante se obtuvo en la señal SXR con un 97,4%, el mejor resultado en exactitud para +3 ventanas adelante se obtuvo en la señal SXR con un 97,5% y finalmente el mejor resultado en exactitud para +5 ventanas adelante se obtuvo en la señal SXR con un 97,5%. También en esta tabla se observa que el mejor resultado

en precisión para +1 ventana adelante se obtuvo en la señal DENS con un 97,3% y la señal SXR con un 97,3%, el mejor resultado en precisión para +3 ventanas adelante se obtuvo en la señal SXR con un 96,7% y finalmente el mejor resultado en precisión para +5 ventanas adelante se obtuvo en la señal SXR con un 96,7%.

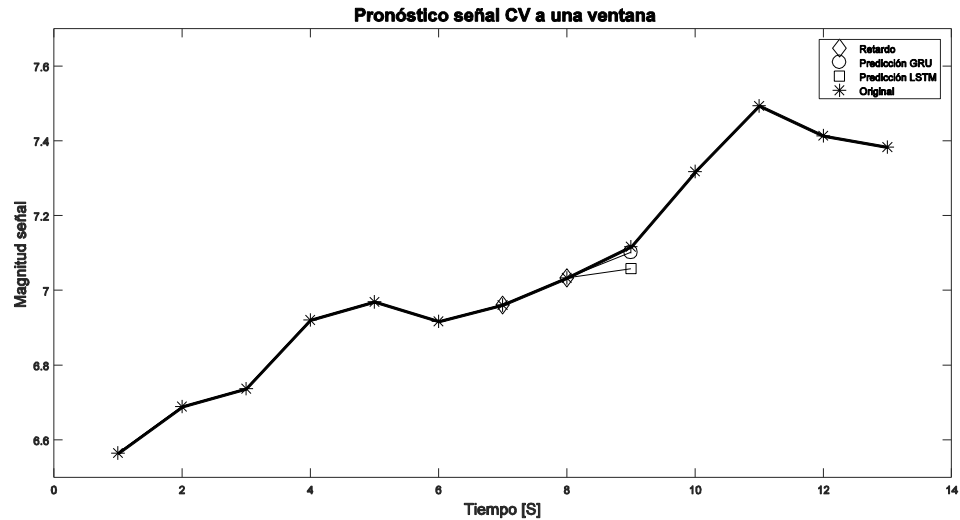


Figura 4-5: Pronostico a una ventana para la señal CV

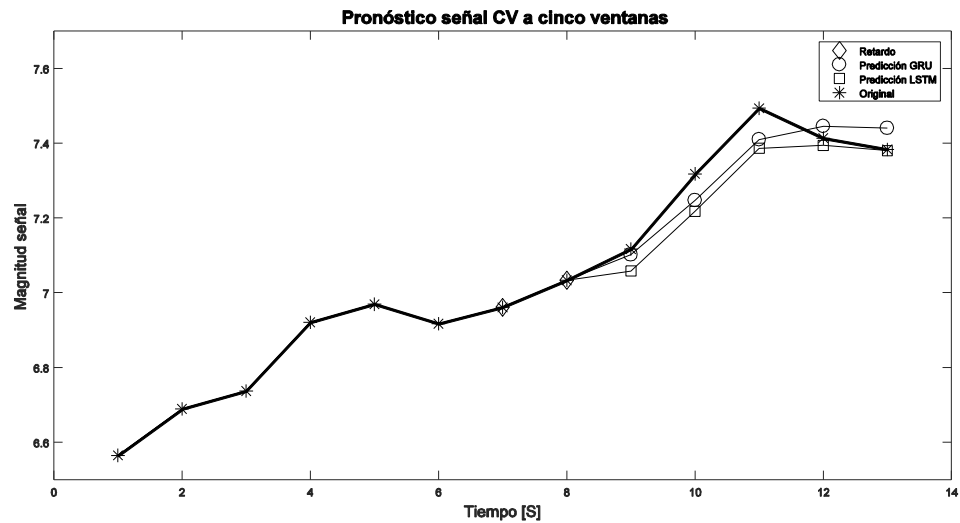


Figura 4-6: Pronostico a cinco ventanas para la señal CV

Se muestran en la Figura 4-5 y Figura 4-6, las gráficas obtenidas para el pronóstico a una y cinco ventanas futuras, para la señal CV, se muestran en la gráfica ambos pronósticos de la red, en la red LSTM y red GRU. Como se aprecia el primer pronóstico, la red GRU es la que se acerca más al valor real, pero a medida que se avanza en el pronóstico a cinco ventanas, tanto la red GRU como la red LSTM siguen el comportamiento ascendente de la señal, pero cerca del ultimo valor la red LSTM es quien tiene la menor desviación con respecto al valor real.

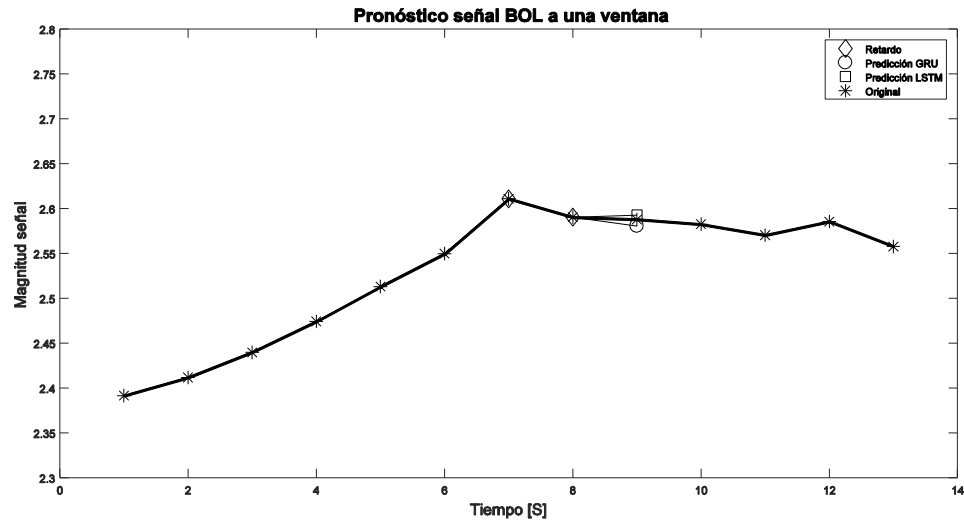


Figura 4-7: Pronóstico a una ventana para la señal BOL

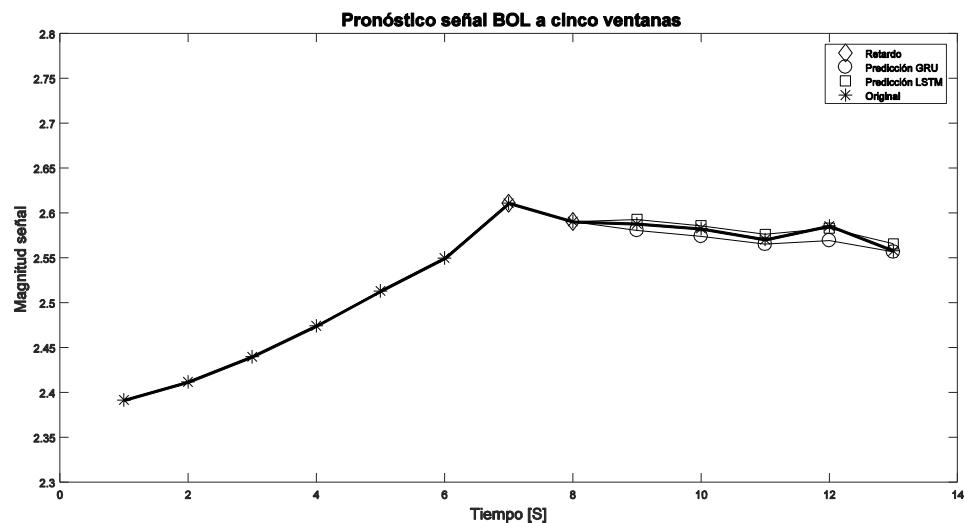


Figura 4-8: Pronóstico a cinco ventanas para la señal BOL

Se muestran en la Figura 4-7 y Figura 4-8, las gráficas obtenidas para el pronóstico a una y cinco ventanas futuras, para la señal BOL, se muestran en la gráfica ambos pronósticos de la red, en la red LSTM y red GRU. Como se aprecia el primer pronóstico, ambas redes pronostican un valor semejante bastante al comportamiento de la señal y a medida que se avanza en el pronóstico a cinco ventanas, ambas siguen el comportamiento de la señal incluido el peak cerca del final de la muestra y es la red GRU quien presenta el menor error con respecto al valor final.

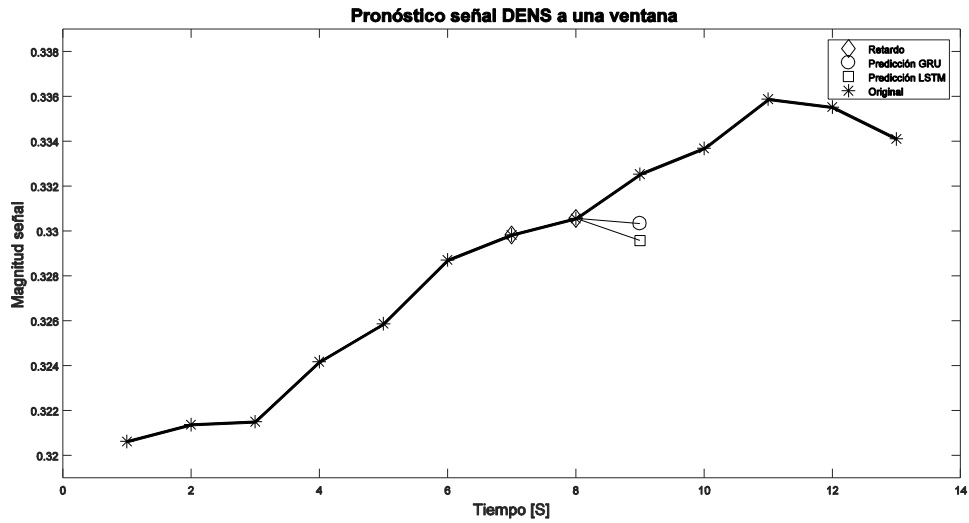


Figura 4-9: Pronostico a una ventana para la señal DENS

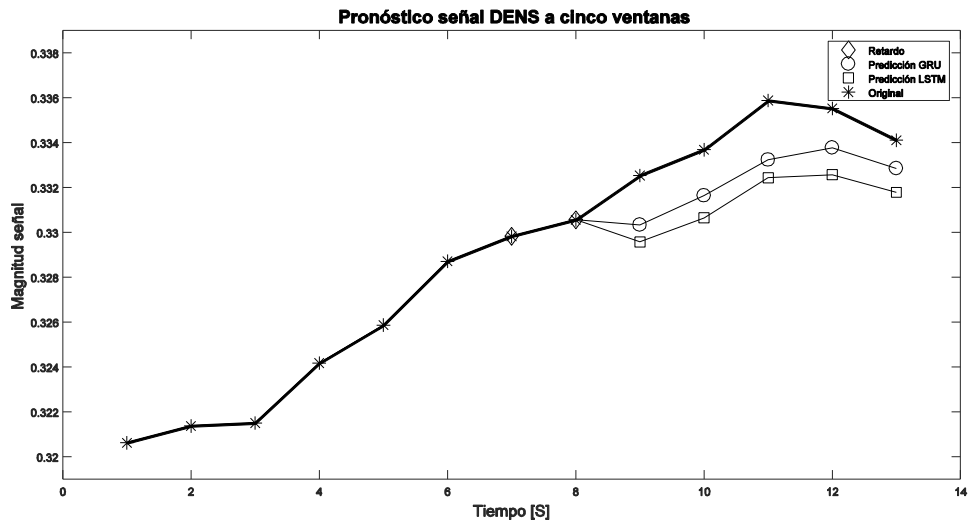


Figura 4-10: Pronostico a cinco ventanas para la señal DENS

Se muestran en la Figura 4-9 y Figura 4-10, las gráficas obtenidas para el pronóstico a una y cinco ventanas futuras, para la señal DENS, se muestran en la gráfica ambos pronósticos de la red, en la red LSTM y red GRU. Como se aprecia el primer pronóstico ambos pronósticos se alejan bastante del comportamiento de la señal, dando una tendencia de continuidad, pero a medida que se avanza en el pronóstico a cinco ventanas, ambas logran seguir el comportamiento ascendente de la señal, pero ambas redes teniendo una desviación mayor con respecto al valor final pronosticado.



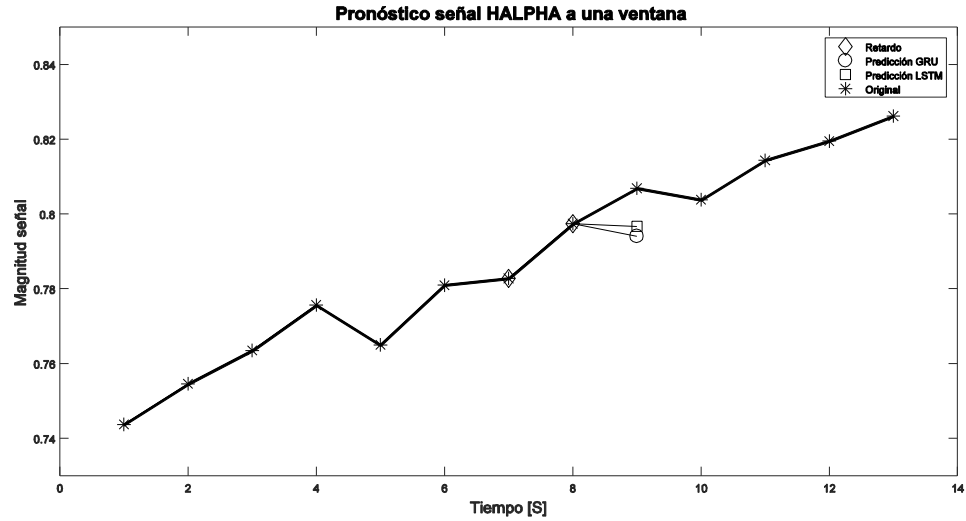


Figura 4-11: Pronostico a una ventana para la señal HALPHA

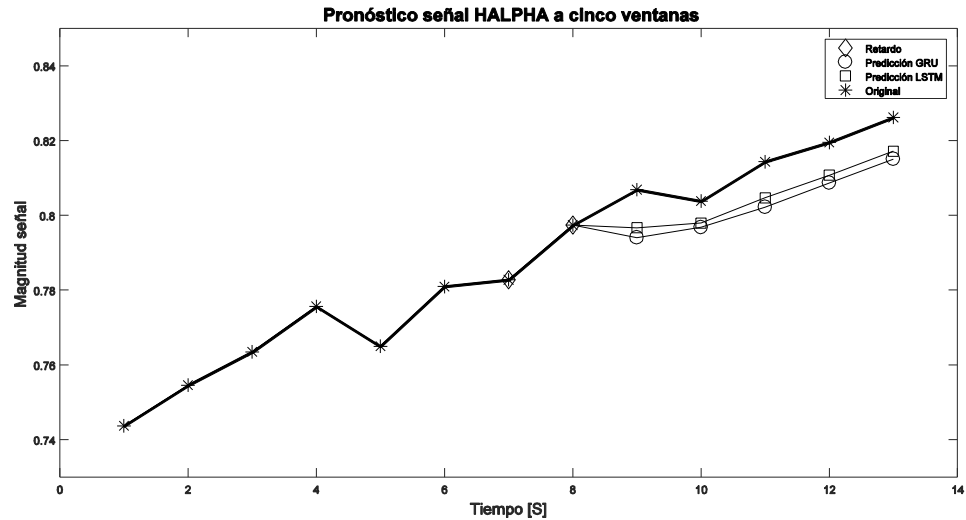


Figura 4-12: Pronostico a cinco ventanas para la señal HALPHA

Se muestran en la Figura 4-11 y Figura 4-12, las gráficas obtenidas para el pronóstico a una y cinco ventanas futuras, para la señal HALPHA, se muestran en la gráfica ambos pronósticos de la red, en la red LSTM y red GRU. Como se aprecia el primer pronóstico ambas redes siguen de cerca el comportamiento de la señal, y a medida que se avanza en el pronóstico a cinco ventanas logran seguir el comportamiento ascendente de la señal y ambas poseen pronósticos muy cercanos entre sí pero alejadas del valor real.

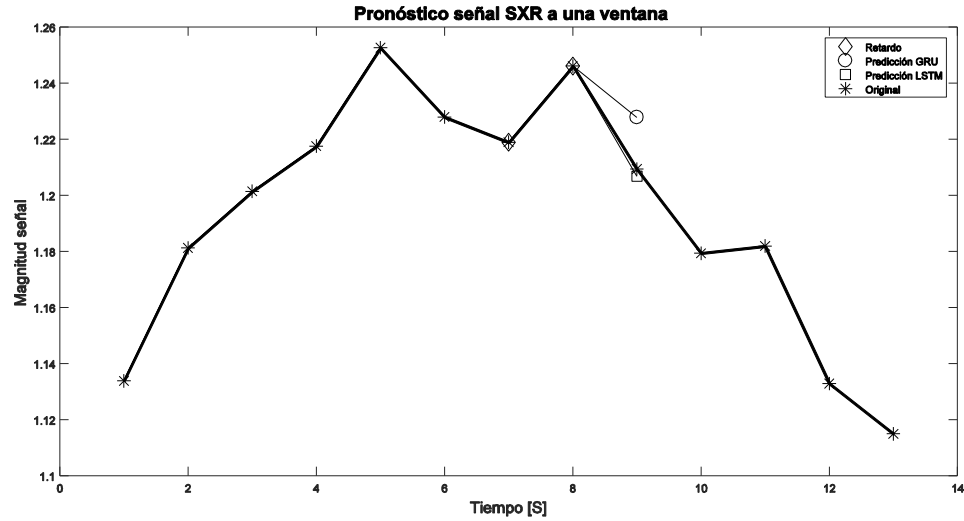


Figura 4-13: Pronostico a una ventana para la señal SXR

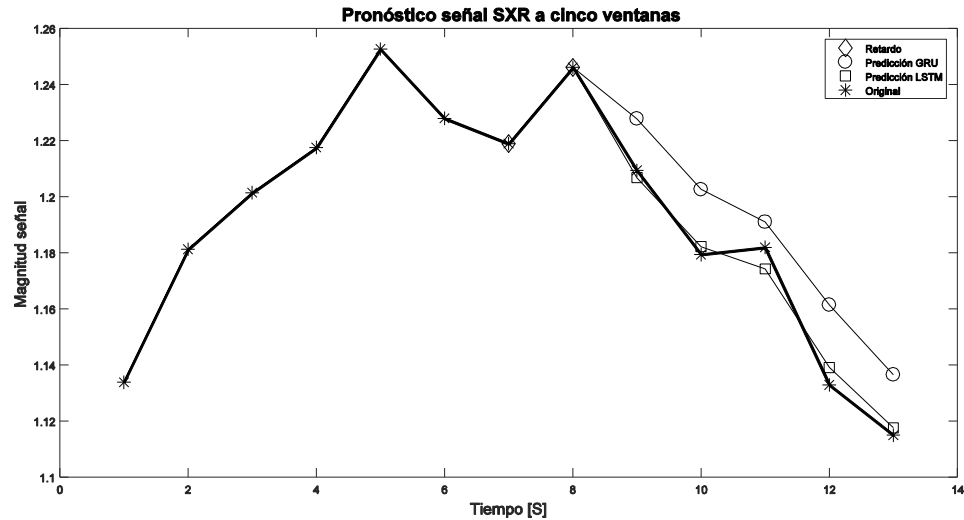


Figura 4-14: Pronostico a cinco ventanas para la señal SXR

Se muestran en la Figura 4-13 y Figura 4-14, las gráficas obtenidas para el pronóstico a una y cinco ventanas futuras, para la señal SXR, se muestran en la gráfica ambos pronósticos de la red, en la red LSTM y red GRU. Como se aprecia el primer pronóstico la red GRU se aleja del comportamiento de la señal, no así la red LSTM que a medida que se avanza en el pronóstico a cinco ventanas logra seguir el comportamiento descendente de la señal casi en su totalidad con el menor error y cerca del ultimo valor llega a tener una desviación con menor error con respecto al valor real.

### 4.3 Clasificación de series temporales

Como motivación adicional gracias a los buenos resultados obtenidos en la sección 4.1 se realizó un análisis para la clasificación de descargas disruptivas y descargas no disruptivas para ellos se efectuaron distintos tipos de experimentos, Cálculo de centroides, Análisis de Peaking Factor y Clasificación de señales. A continuación, muestran los resultados obtenidos en los distintos experimentos.

#### 4.3.1 Cálculo de centroides

A continuación, se muestran los centroides obtenidos clasificados según su distancia euclídea, en la Figura 4-15 se muestra el mejor resultado obtenido con la señal Mode Lock, la Figura 4-16 y la Figura 4-17 muestran los centroides calculados de la Evolución temporal de la afluencia del tungsteno y la Energía diamagnética respectivamente, para señales disruptivas que generan alarmas.

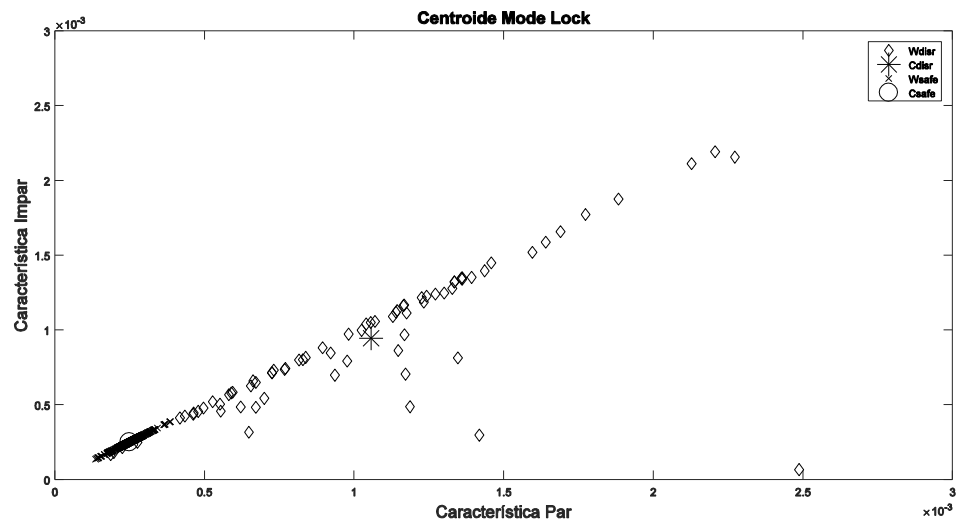


Figura 4-15: Centroide obtenido para la señal Mode Lock

En la Figura 4-15 se observa el primer análisis realizado para el cálculo de centroides, la señal Mode Lock tiene características que pueden determinar cuándo se producirá una interrupción pero al ser una señal que produce para este proceso, aun no se tiene consenso si es ideal ocuparla o no para realizar pronóstico, clasificación o cualquier otro proceso de resolución de problemática de la interrupción nuclear, pero se utiliza en este proyecto para validar el uso de técnicas que pueden ayudar a encontrar características.

En este caso agrupamiento de características en el centroide para la señal Mode Lock, en ella visualiza como las características pueden ser separadas por una línea oblicua dando la opción de aplicar esta técnica a otras señales con características similares.

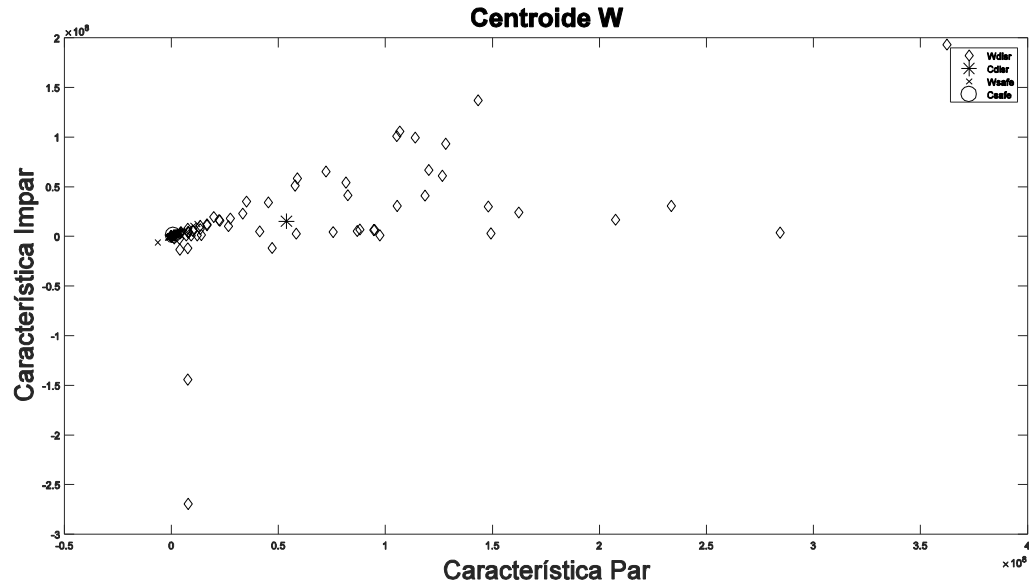


Figura 4-16: Centroide obtenido para la señal Evolución temporal de la afluencia del tungsteno

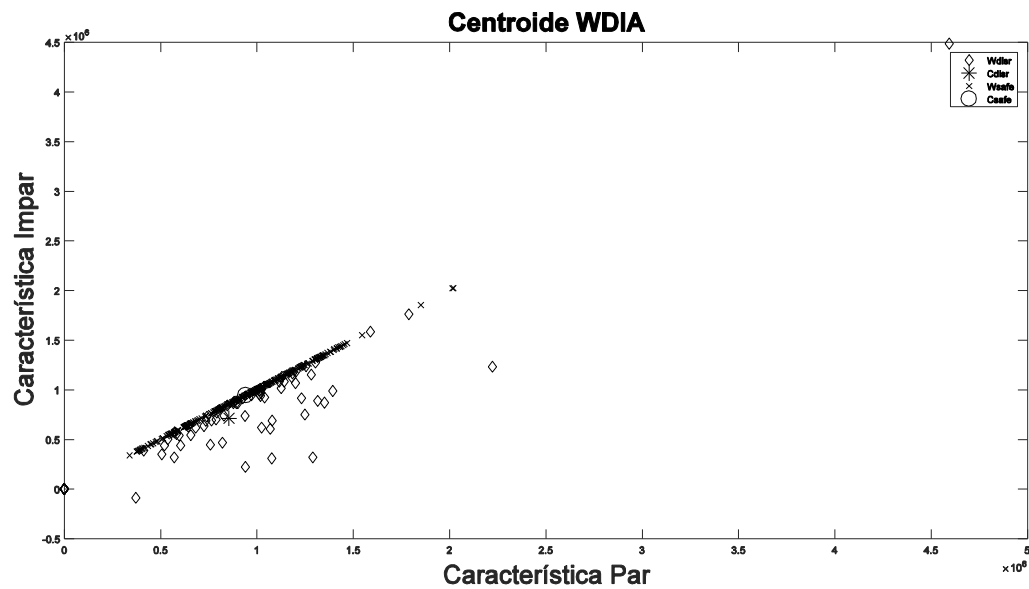


Figura 4-17: Centroide obtenido para la señal Energía diamagnética

la Figura 4-16 y Figura 4-17 que muestra el agrupamiento de características para Evolución temporal de la afluencia del tungsteno (W) y Energía diamagnética (WDIA), en donde se visualiza que no es posible separar las características para señales disruptivas y señales no disruptivas, siendo el caso más extremo en la Figura 4-17 donde es imposible separar señales disruptivas de no disruptivas obteniendo el mayor error.

### 4.3.2 Análisis de Peaking Factor

A continuación, se muestran las gráficas de peaking factor obtenidas junto con el valor RMS, en la Figura 4-18 se muestra el resultado obtenido por el peaking factor entre la señal 3 y la señal 12 y en la Figura 4-19 el resultado obtenido por el peaking factor entre la señal 21 y la señal 12, para señales disruptivas que generan alarmas.

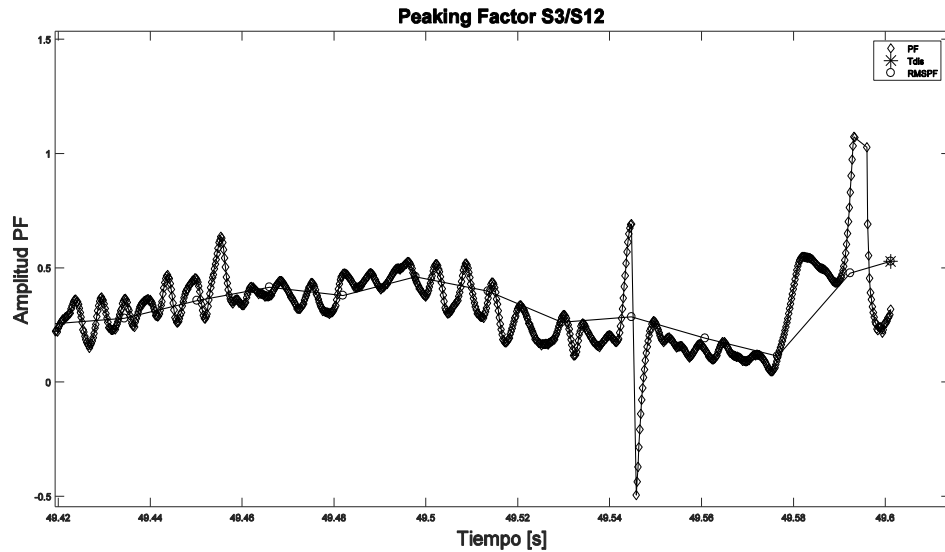


Figura 4-18: Peaking factor obtenido para la señal de canal 3 y la señal de canal 12

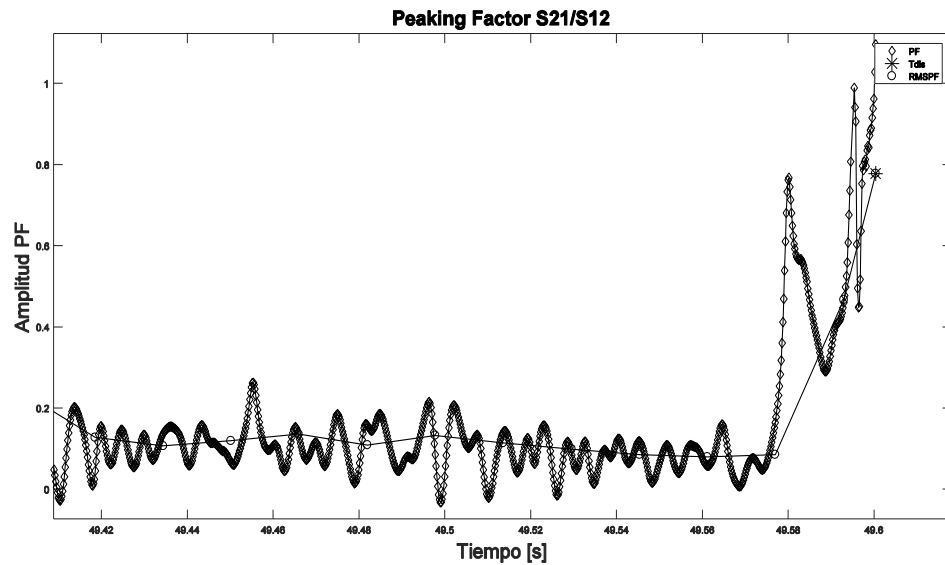


Figura 4-19: Peaking factor obtenido para la señal de canal 21 y la señal de canal 12

Se observa en la Figura 4-18 y Figura 4-19 como con el cálculo de peaking factor es posible obtener un tipo de característica, en este caso una pendiente positiva cerca del tiempo disruptivo que puede ser de ayuda al momento de utilizar este cálculo para realizar clasificación.

### 4.3.3 Resultados obtenidos centroides y peaking factor

A continuación, se muestran los resultados obtenidos con estas dos pruebas:

Tabla 4-5: Tabla de resultados Clasificación de señales criterios Centroides y Peaking Factor

Señal	SR %	VA %	PA %	TD %	MS %	FA %	Mean [ms]	STD [ms]
Mode Lock	92	68	5,33	18,67	8	2,41	209,3	420
W	62,67	28	29,33	5,33	37,33	24,07	1081,9	2204,2
WDIA	93,33	0	93,33	0	6,67	100	11683	52391
PF S3/S12	100	0	100	0	0	100	11411	50159
PF S21/S12	100	0	100	0	0	100	11396	5033

El desarrollo de continuas pruebas en el ámbito de poder encontrar precursores es importante para entrenar mejores modelos, el cálculo de centroide ha arrojado importantes resultados en términos visuales, gráficas de nubes de puntos y en términos de alarma, cada prueba realiza nos lleva a tratar de comprender la dinámica del plasma ya que es difícil saber su comportamiento, es necesario idear nuevas estrategias que logren obtener características que puedan ser usadas para clasificar descargas disruptivas y no disruptivas.

El mejor resultado se obtuvo con el análisis del Cálculo de Centroide de la Señal Mode Lock de la campaña C36, con SR de 92%, MS de 8% y FA de 2,41%, en comparación al Centroide W (Evolucion temporal de la afluencia del tungsteno) y Centroide WDIA (energía diamagnética).

El cálculo de Peaking Factor junto con el valor RMS y su correspondiente cálculo de Centroide, en ambos peaking factor realizados no muestran mejor clasificación con respecto a lo realizado anteriormente (cálculo de centroides por ventanas impar-par), debido a que las señales son en algunos canales (señales de canales 3-12-21) son demasiado ruidosas y la distancia entre ambos centroides (centroide promedio y centroides de ventanas para testeo) siguen siendo demasiado pequeñas para ser diferenciadas unas de otras, generando ambas alarmas con SR de 100% pero en términos de alarmas prematuras, que no son válidas a la hora de generar estrategias de evitación o mitigación.

## 4.4 Clasificación de señales disruptivas

Como motivación adicional debido a los buenos resultados obtenidos en el pronóstico de series temporales para señales de fusión nuclear, se implementó una red de tipo LSTM y tipo GRU para la clasificación de señales disruptivas y señales no disruptivas. Recordando los tiempos de alarmas del Capítulo 3, gracias a ellas podemos comprobar el rendimiento de nuestro modelo, en término de las alarmas validas que se generan.

El principal objetivo es obtener un gran porcentaje de alarmas válidas, considerando las especificaciones que nos pide ITER ( $TPR \geq 95\%$  y  $FPR \leq 5\%$ ). Con estas alarmas válidas es posible realizar cualquier tipo de estrategia de evitación o mitigación según se requiera, en cambio la

activación de otra alarma no es deseable ya que no presenta utilidad en las estrategias de evitación y mitigación.

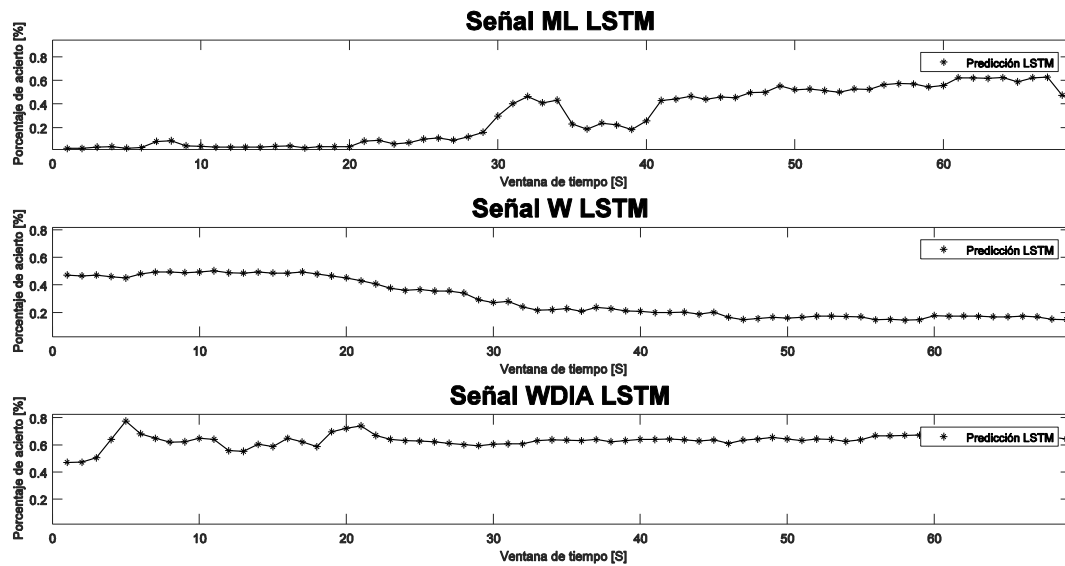


Figura 4-20: Predicción para la clasificación de señales con la red LSTM

En la Figura 4-20 se muestran los resultados de las predicciones para la red LSTM por ventana, en ella se muestra como ventana a ventana de tiempo la red considera que valores son disruptivos, en un rango de 0-100%, es decir, una ventana con 0% no es disruptiva y una ventana con 100% de acierto es una ventana disruptiva, la cual activa las correspondientes alarmas. A continuación, se muestran los resultados obtenidos para la clasificación de señales:

Tabla 4-6: Tabla de resultados Clasificación de señales en red LSTM.

Señal	LSTM							Mean [ms]	STD [ms]
	SR %	VA %	PA %	TD %	MS %	FA %			
Mode Lock	49,14	43,42	5,72	0	50,86	16,97	812,7	486,1	
W	44,74	0	44,74	0	55,26	0	369,4	2162	
WDIA	21,06	10,53	10,53	0	78,94	31,32	1190	704,6	

En la Tabla 4-6 se muestran los resultados obtenidos en la red LSTM diseñada, en ella se muestran los distintos porcentajes de alarmas obtenidas, se observa que el mejor SR se logró con la señal Mode Lock con un 49,14% y 43,42% de alarmas válidas, estas alarmas válidas poseen un 812,7 ms en promedio y una desviación estándar de 486,1 ms y además un 16,97% de falsas alarmas, el peor resultado se obtuvo en la señal energía diamagnética con un SR de 21,06%, alarmas válidas de 10,53%, estas alarmas válidas poseen un 1190 ms en promedio y una desviación estándar de 704,6 ms y además falsas alarmas de 31,32%.

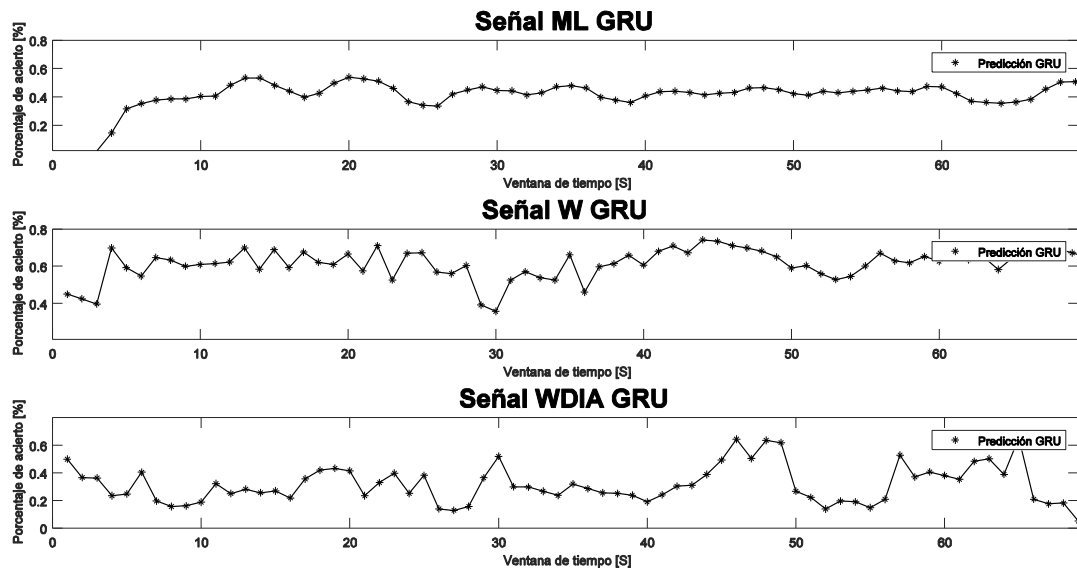


Figura 4-21: Predicción para la clasificación de señales con la red GRU

En la Figura 4-21 se muestran los resultados de las predicciones para la red GRU por ventana, en ella se muestra como ventana a ventana de tiempo la red considera que valores son disruptivos, en un rango de 0-100%, es decir, una ventana con 0% no es disruptiva y una ventana con 100% de acierto es una ventana disruptiva, la cual activa las correspondientes alarmas. A continuación, se muestran los resultados obtenidos para la clasificación de señales:

Tabla 4-7: Tabla de resultados Clasificación de señales en red GRU.

Señal	GRU							Mean [ms]	STD [ms]
	SR %	VA %	PA %	TD %	MS %	FA %			
Mode Lock	67,43	60	7,43	0	32,57	14,29	0,764	0,431	
W	4,57	0	4,57	0	95,43	1,06	6,04	3,99	
WDIA	0	0	0	0	100	100	-	-	

En la Tabla 4-7 se muestran los resultados obtenidos en la red LSTM diseñada, en ella se muestran los distintos porcentajes de alarmas obtenidas, se observa que el mejor SR se logró con la señal Mode Lock con un 67,43% y 60% de alarmas válidas, estas alarmas validas poseen un 0,764 ms en promedio y una desviación estándar de 0,431 ms y además un 14,29% de falsas alarmas, el peor resultado se obtuvo en la señal energía diamagnética con un SR de 0 %, teniendo ninguna alarma activada y falsas alarmas de 100 %.



## Discusión y conclusiones

Una de las principales motivaciones de este proyecto fue implementar una inteligencia artificial que pudiese analizar la inmensa cantidad de datos generados por los dispositivos experimentales de fusión nuclear, el estudio de redes neuronales nos llevó a utilizar la red neuronal recurrente en su arquitectura Deep Learning, la investigación nos condujo a analizar las señales secuenciales de las bases de datos proporcionadas para realizar pronóstico y clasificación de señales, para ello fueron seleccionadas dos arquitecturas conocidas como LSTM y GRU, las cuales debido a la reciente utilización en análisis de secuencias de datos nos proporcionó una base para poder utilizarlas para analizar series de tiempo de señales de fusión nuclear. Hasta el momento de la realización de este proyecto de titulación no había trabajos conocidos enfocados en nuestra problemática con el uso de estas redes, es así que se trabajó de forma muy ambiciosa y exhaustiva en lograr los mejores resultados posibles.

Unas de las principales con enfoques fue utilizar redes neuronales recurrentes con Deep Learning, donde se estudiaron las arquitecturas más conocidas con ellas se observó que presentan similitudes y diferencias, ya que tanto como LSTM y GRU poseen una mejora con respecto a la red neuronal recurrente.

La más prominente característica compartida entre estas unidades es la componente aditiva de su actualización de  $t$  a  $t + 1$ , que es la falta en la tradicional unidad recurrente. La unidad recurrente tradicional siempre reemplaza la activación, o el contenido de una unidad con un nuevo valor calculado desde la presente entrada y el estado oculto previo. Por otra parte, ambas unidades, LSTM y GRU, mantienen la existencia de contenido y agregan nuevo contenido al tope de este. Esta naturaleza de adición tiene dos ventajas. La primera, es fácil para cada unidad recordar la existencia de características específicas en el flujo de entrada para una larga serie de pasos. Cualquier característica importante, decidida por la compuerta de olvido de la LSTM o la compuerta de actualización de la GRU, no se sobrescribirá, sino que se mantendrá tal como está. Segundo, y quizás la más importante, esta adición crea efectivamente rutas de atajo que omiten múltiples pasos temporales. Estos atajos permiten al error ser retro propagado fácilmente sin ser rápidamente desvanecido (si la unidad de compuerta está casi saturada a 1) como resultado de pasar a través de múltiples, no linealidades acotadas, esto reduce la dificultad debido al gradiente desvaneciente.

Estas dos unidades poseen varias diferencias también, una característica de las unidades LSTM que falta en las unidades GRU es la exposición controlada del contenido de la memoria. En la unidad LSTM la cantidad de contenido de la memoria que es visto, o usado por las otras unidades en la red es controlada por la compuerta de salida. Por otra parte, la unidad GRU expone todo el contenido sin ser controlado. Otra diferencia es en la localización de la compuerta de entrada, o la correspondiente compuerta de reseteo. La unidad LSTM calcula el nuevo contenido de la memoria sin ningún control por separado de la cantidad de información que fluye desde los pasos de tiempos previos. Más bien, la unidad LSTM controla la cantidad de nuevo contenido de memoria que se agrega a la celda independientemente de la compuerta de olvido. Por otra parte, la unidad GRU controla la información que fluye desde pasos previos de activación mediante el cálculo de nuevos, candidatos de activación, pero no controla de forma independiente la cantidad de activación candidata que se agrega (el control está vinculado mediante la compuerta de actualización).

Desde estas similitudes y diferencias, se hace difícil concluir que tipo de compuerta funcionaria mejor en general, ya que estas dos unidades se comportaron de manera similar entre sí de acuerdo con los experimentos realizados, esto motiva al uso de estas unidades en problemáticas de pronóstico y clasificación señales para la evitación y mitigación de la disrupción nuclear a partir del aprendizaje automático.

Para el pronóstico de series temporales, en primera instancia se implementó el pronóstico para una ventana de tiempo con la base de datos JET, con la red LSTM y GRU logrando buenos resultados, el mejor resultado que se tuvo con la red LSTM fue de 99,40% con la señal densidad y el mejor resultado que se tuvo con la red GRU fue de 99,91% con la señal corriente de plasma, estos resultados motivaron la implementación a más horizontes de pronóstico donde se obtuvieron los siguientes resultados en la señal SXR obtuvo en el pronóstico a +1, +3 y +5 horizonte de tiempo, un 97,3% de exactitud y una precisión de 97,1%, 96,5% y un 96,2% respectivamente, siendo el mejor resultado para la red LSTM y también la señal SXR obtuvo en el pronóstico a +1, +3 y +5 horizonte de tiempo, una exactitud de 97,4%, 97,5% y un 97,5% respectivamente y una precisión de 97,3% compartida con la señal densidad, un 96,7% y un 96,7% respectivamente siendo estos los mejores resultados para la red GRU. Se concluye que para el pronóstico de series temporales las redes LSTM y GRU tienen un rendimiento óptimo al tener porcentajes de exactitud y precisión sobre el 96% lo que motivo a su implementación para clasificación de señales demostrando ser de gran precisión a la hora de predecir una ventana futura luego de ser entrenada con datos de ventanas previas. Esto es de gran ayuda para futuros trabajos donde se implemente un modelo que pueda, a partir de esta capacidad, predecir el comportamiento y bajo cierto umbral activar alguno tipo de alarma para generar evitación o mitigación.

Con los resultados obtenidos en el pronóstico de series temporales, se utilizó las mismas arquitecturas para desarrollar un clasificador de señales, el objetivo era lograr desarrollar un clasificador de descargar disruptivas para generar evitación y mitigación. El clasificador diseñado con una red LSTM logro con la señal Mode Lock un SR de 49,14% y un FA de 16,97% y el peor resultado fue con la señal energía diamagnética con un SR de 21,06% y un FA de 31,32%. El

clasificador con una red GRU obtuvo con la señal Mode Lock un SR de 67,43% y un FA de 14,29% además el peor resultado fue con la señal energía diamagnética con un SR de 0% y un FA de 100%. Concluyendo que aún falta mucho para estar en los estándares permitidos por ITER ( $TPR \geq 95\%$  y  $FPR \leq 5\%$ ) [14].

Finalmente se puede comentar y concluir que es necesario seguir trabajando en la construcción de algoritmos que permitan mejorar estos resultados, muchas pueden ser las opciones viables, pero según lo estudiado y desarrollado durante este largo y exhaustivo proceso se puede acotar que una de las opciones es obtener nuevas características por señales para ser ingresadas a la red, es decir, ocupar no solo el valor medio de la señal si no otras características temporales que puedan ayudar a obtener mejores precursores del comportamiento dinámico del plasma, realizar un filtro de señales ideales para ser analizadas, ya sea analizando una a una las señales o aplicar alguna media de entropía que pueda seleccionar las mejores señales, realizar filtraje para eliminar las señales muy ruidosas (ruido que no tiene conexión con el comportamiento del plasma), recopilar más y nuevas señales para ser analizadas, que permitan generar una gran base de datos que puedan ingresar a una red LSTM o GRU para ser analizada tratando de explotar al máximo las capacidades de una red neuronal recurrente Deep Learning.

# Bibliografía

- [1] W. N. Association, «World Nuclear Association,» Septiembre 2017. [En línea]. Available: [www.world-nuclear.org](http://www.world-nuclear.org). [Último acceso: 6 Diciembre 2017].
- [2] CienChile, «cienchile,» Centro de información sobre energía nuclear de Chile, [En línea]. Available: [www.cienchile.cl](http://www.cienchile.cl). [Último acceso: 06 Diciembre 2017].
- [3] CIEMAT, «Laboratorio Nacional de Fusión,» Laboratorio Nacional de Fusión, [En línea]. Available: [www.fusion.ciemat.es](http://www.fusion.ciemat.es). [Último acceso: 6 Diciembre 2017].
- [4] EUROfusion, «EUROfusion,» [En línea]. Available: [www.euro-fusion.org](http://www.euro-fusion.org). [Último acceso: 6 Diciembre 2017].
- [5] ITER, «iter,» [En línea]. Available: [www.iter.org](http://www.iter.org). [Último acceso: 6 Diciembre 2017].
- [6] J. Wesley, «Disruption Avoidance and Mitigation Challenges for ITER and Beyond,» 2010.
- [7] I. Goodfellow, Deep Learning, 2016.
- [8] Y. Cinar, E. Gaussier, P.Goswami y H.Mirisae, «Time Series Forecasting Using RNNs,» Grenoble, 2017.
- [9] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Springer, 2012.
- [10] Y. Bengio, P. Simard y P. Frasconi, «Learning Long-Term Dependencies with Gradient Descent is Difficult,» *IEEE TRANSACTIONS ON NEURAL NETWORK*, vol. 5, nº 2, 1994.
- [11] S. Hochreiter y J. Schmidhuber, «Long Short-Term Memory,» *MIT Press Cambridge*, pp. 1735-1780, 1997.

- [12] J. Chung, C. Gulcehre, K. Cho y Y. Bengio, «Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,» *arXiv e-prints*, 2014.
- [13] G. Farías, S. Dormido-Canto, J. Vega y N. Díaz, «Initial results with time series forecasting of TJ-II heliac waveforms,» *ELSEVIER*, 2015.
- [14] L. Alfaro, «Sistema de predicción de disrupción en base de datos de Fusión Nuclear,» 2016.