Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

ANÁLISIS WAVELET EN SEÑALES DEL TJ-II MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Autor: Alejandro Álvarez Montero

Tutor: Augusto Pereira González

2022

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

ANÁLISIS WAVELET EN SEÑALES DEL TJ-II MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Autor: Alejandro Álvarez Montero

Tutor: Augusto Pereira González

TRIBUNAL:

Presidente: Fco. Javier Acevedo Rodríguez

Vocal 1º: Saturnino Maldonado Bascón

Vocal 2º: Augusto Pereira González

FECHA: Septiembre 2022

Agradecimientos

En primer lugar, me gustaría agradecer a Augusto por su ayuda y comprensión, sin su paciencia y dedicación hubiese sido ardua la realización de este trabajo.

A mi familia, sobre todo a mis padres y mi hermana, por su confianza y sabiduría, sin ellos no podría haber alcanzado el lugar en el que me encuentro. Sus ánimos y alegrías por mis logros han sido claves en mis ganas de progresar.

Y, por último, gracias a todas esas personas que me he ido cruzando a lo largo de esta dura pero maravillosa etapa de mi vida, compartir con ellos esta experiencia ha sido un placer.

Análisis Wavelet en señales del TJ-II mediante técnicas de aprendizaje automático

Autor: Alejandro Álvarez Montero

Tutor: Augusto Pereira González

Departamento: Teoría de la Señal y Comunicaciones

Titulación: Grado en Ingeniería Electrónica y Automática Industrial

Palabras clave: Transformada Wavelet, análisis de datos, calentamiento NBI y programa de clasificación.

Resumen

El análisis de señales de evolución temporal mediante la transformada Wavelet permite obtener otras señales de menor dimensionalidad conservando sus principales características. Las técnicas de clasificación mediante aprendizaje automático permiten predecir la pertenencia de nuevas señales a diferentes grupos, a partir de un modelo entrenado previamente con otras señales iniciales conocidas. En este trabajo se propone el uso combinado de la transformada Wavelet y algoritmos de aprendizaje automático para recuperar y clasificar ondas similares a partir de un subconjunto de señales de la base de datos del TJ-II. En una primera etapa, el análisis Wavelet pre-procesará las señales de plasma para reducir su información y extraer sus principales características. En la siguiente etapa, y utilizando las señales suavizadas producidas por el análisis anterior, se aplicarán algoritmos de clasificación para mostrar la eficiencia del método propuesto para abordar el problema de resolver similitudes en miles de señales de plasmas calientes confinados magnéticamente.

Wavelet analysis of signals of the TJ-II using machine learning techniques

Author: Alejandro Alvarez Montero

Tutor: Augusto Pereira González

Department: Signal Theory and Communications

Degree: Degree in Electronic Engineering and Industrial Automation

Keywords: Wavelet transform, data analysis, NBI heating and classification program.

Abstract

The analysis of time evolution signals by means of the Wavelet transform allows to obtain other signals of lower dimensionality while conserving their main characteristics. Classification techniques using machine learning allow predicting the membership of new signals to different groups, from a previously trained model with other known initial signals. In this work, the combined use of the Wavelet transform and machine learning algorithms is proposed to recover and classify similar waves from a subset of signals from the TJ-II database. In a first stage, Wavelet analysis will pre-process the plasma signals to reduce their information and extract their main features. In the next stage, and using the smoothed signals produced by the previous analysis, classification algorithms will be applied to show the efficiency of the proposed method to address the problem of resolving similarities in thousands of signals from magnetically confined hot plasmas.

ÍNDICE

Tabla de figuras

1 Estructura de la memoria

En primer lugar, se ha realizado una breve introducción y una explicación de los principales objetivos, de forma que se pueda explicar en que está fundamentado el estudio que se ha estudiado y analizado.

En segundo lugar, se ha comentado las técnicas matemáticas en las cuales se fundamentan todos y cada uno de los procesos que se van a realizar a lo largo de la memoria.

Además, se ha explicado la creación y organización de la base de datos utilizada, en donde se encuentran los datos que se han utilizado para el estudio.

Del mismo modo, el preprocesamiento de los datos y los procesos en los que se fundamenta la clasificación que se va a realizar en la base de datos.

De particular importancia se ha explicado en qué consisten todos y cada uno de los programas de clasificación que se van a utilizar, así como sus líneas de programación.

Especialmente, se han mostrado los resultados de la utilización de dichos programas de clasificación en la base de datos, incluyendo la representación cuando se omiten ciertas variables.

Así mismo, se han expuesto las conclusiones a las que se han llegado después de todo el estudio, comentando cual es o son los mejores programas de clasificación y, a su vez, los más recomendables en este caso.

Por último, se muestra la bibliografía utilizada en todos y cada uno de los aparatados de la memoria.

2 Introducción

El dispositivo experimental de plasmas por confinamiento magnético TJ-II genera decenas de señales de evolución temporal en cada descarga de operación. Cada señal del TJ-II puede albergar de media más de 100000 muestras de información bruta reflejando diferentes parámetros físicos; presión, temperatura, densidad, campos eléctricos, campos magnéticos, etc. Actualmente existen más de 55000 descargas almacenadas con 6 Tb de información comprimida. Ante tal dimensionalidad, se hace necesario contar con técnicas automáticas y métodos de acceso eficientes que puedan buscar similitudes y recuperar datos específicos en tiempos de cómputo razonables.

El análisis de señales de evolución temporal mediante la transformada wavelet permite obtener otras señales de menor dimensionalidad conservando sus principales características. Las técnicas de clasificación mediante aprendizaje automático permiten predecir la pertenencia de nuevas señales a diferentes grupos, a partir de un modelo entrenado previamente con otras señales iniciales conocidas. En este trabajo se propone el uso combinado de la transformada wavelet y algoritmos de aprendizaje automático para recuperar y clasificar ondas similares a partir de un subconjunto de señales de la base de datos del TJ-II. En una primera etapa, el análisis Wavelet pre-procesará las señales de plasma para reducir su información y extraer sus principales características. En la siguiente etapa, y utilizando las señales suavizadas producidas por el análisis anterior, se aplicarán algoritmos de clasificación para mostrar la eficiencia del método propuesto para abordar el problema de resolver similitudes en miles de señales de plasmas calientes confinados magnéticamente.

Los resultados y conclusiones a obtener, a partir del presente estudio y análisis wavelet en la clasificación de señales del TJ-II, proporcionará información relevante para determinar si los valores resultantes (y en qué medida) de la transformada wavelet (los coeficientes de aproximación y los coeficientes de detalle) o combinación de ellos, pueden servir como características válidas en la predicción de señales de evolución temporal generados en el dispositivo TJ-II. En esta línea y a tenor del análisis y resultados obtenidos, dichos coeficientes se han utilizado en otros dos trabajos fin de grado en los cuales se pretenden simular y predecir señales sintéticamente en base al comportamiento anterior de las mismas y haciendo uso de la enorme cantidad de información disponible en la base de datos del dispositivo experimental TJ-II.

3 Objetivos

Previamente al desarrollo de todas y cada una de las partes que comprenden esta memoria, se debe contextualizar los hechos para que se entienda todo de una mejor manera. Para ello, lo primero que se debe mencionar es el dispositivo del que hemos extraído la información con la que se ha realizado el exhaustivo estudio, el TJ-II. Se trata de un dispositivo experimental de fusión termonuclear en el que, a lo largo de los últimos años, se han realizado muchos estudios, y del que se han extraído una gran cantidad de datos, que, posteriormente se han calificado como señales de evolución temporal. Una vez se han extraído los datos, se busca relacionarlos de tal forma que se pueda un patrón o combinación que ayude a clasificarlos.

Por otra parte, el proyecto se ha basado en un procesamiento de datos masivo, las técnicas utilizadas, big data, son las encargadas de realizar tal análisis. Gracias a ellas, se pueden relacionar todos aquellos datos que a ojo del ser humano son imperceptibles. En este caso, cumple con lo descrito del TJ-II, ya que para su gran número de datos procesados se han de utilizar algoritmos o transformadas matemáticas con el fin de reducir el número de datos obtenidos para poder clasificarlos, pero, sin perder sus características esenciales.

Mediante la aplicación de estas técnicas, se pretende comprobar la capacidad predictiva de las mismas, para clasificar señales de evolución temporal atendiendo a los métodos y modos de calentamiento del plasma que se aplican en el dispositivo experimental TJ-II perteneciente al CIEMAT.

Los principales objetivos del presente Trabajo Fin de Grado son los siguientes:

- **Estudiar y analizar** señales de evolución temporal en plasmas de fusión nuclear pertenecientes al dispositivo experimental TJ-II.
- **Predecir** mediante el análisis de señales y los modelos de clasificación, señales y variables físicas aplicando técnicas de aprendizaje automático.
- **Obtener** modelos de clasificación resultantes en base a dichas señales.
- **Evaluar y comparar** la idoneidad de utilizar dichos modelos, para poder clasificar predictivamente las señales.

4 Transformadas matemáticas

4.1 Transformada wavelet y Fourier

La transformada Wavelet consiste en el análisis de una señal/función mediante su descomposición en componentes de tiempo y frecuencia de acuerdo con una escala de resolución y, en donde se retiene la información con gran exactitud, por lo que se considera un método de preprocesamiento de alta precisión. La extracción de datos como principal uso se debe a la capacidad de extracción y reducción que se obtiene sin perder información que nos permita llegar a conclusiones.[1]

Todas y cada una de las funciones se obtienen a partir de una función Wavelet Madre, en la cual las dilataciones se encuentran contraladas por la variable a y las traslaciones por la b.

$$g\,(a,b,t) = \frac{1}{\sqrt{a}}\,.\,gbasic\left(\frac{t-b}{a}\right) \qquad (4.1)$$

Otro dato importante sobre este tipo de transformada es la de poder analizar distintas señales a distintas escalas, ya que los distintos programas o algoritmos clasificatorios que se utilicen procesarán los datos en diferentes escalas y resoluciones. Es decir, estamos observando una señal pequeña con poca resolución solo seremos capaces de visualizar los datos de mayores dimensiones, mientras que en una escala grande de gran resolución será más fácil apreciar los pequeños detalles. Es por ello por lo que al analizar una señal se descompone en una serie de distintas escalas, para poder reproducirla como una superposición de funciones.[2]

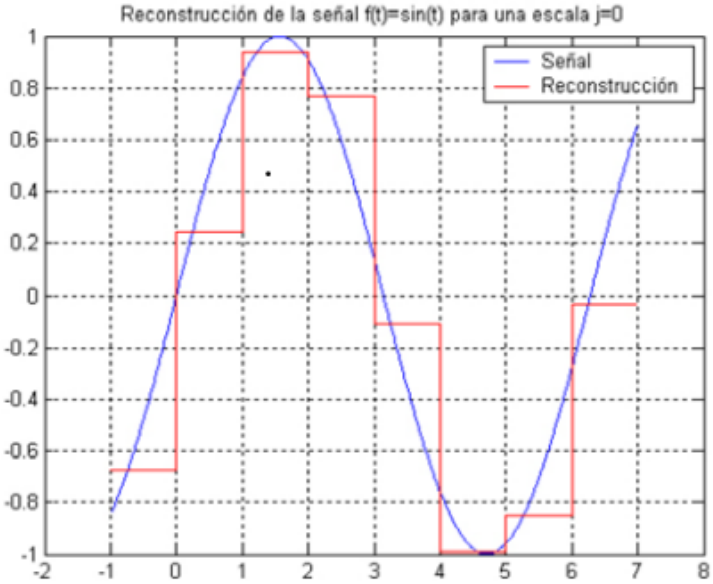


Fig. 1. Reconstrucción de una señal a escala

[1] **La Transformada Wavelet.** [Azor Montoya, 2001]

[2] **Estudio de técnicas basada en la transformada wavelet y optimización de sus parámetros para la clasificación por texturas de imágenes digitales.** [Fernández Sarría,2007]

Por otra parte, la transformada de Fourier es utilizada para el estudio de funciones no periódicas, representando de forma única, las señales en componentes frecuenciales. A medida que se incremente el periodo, las componentes armónicas estarán más cercanas a la frecuencia, convirtiéndose así, para un periodo infinito, la suma de una serie de Fourier en una integral.[3]

$$F(f(t)) = F(\omega) = \int_{-\infty}^{+\infty} f(t).e^{-i\omega t}\,dt, \quad \forall -\infty < \omega < +\infty \qquad (4.2)$$

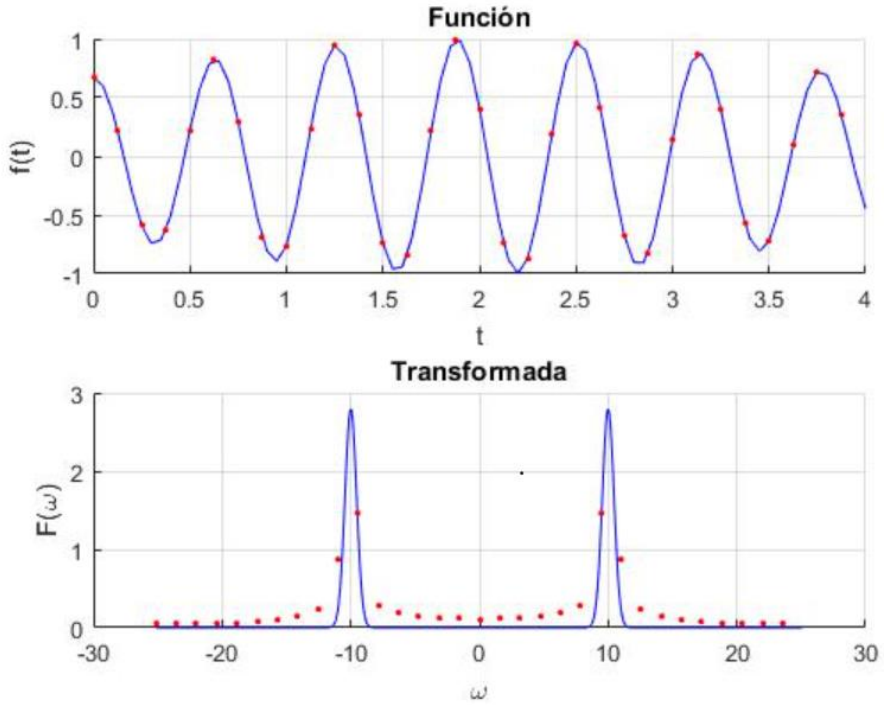


Fig. 2. Transformada de Fourier

Dicha transformada debe cumplir unas condiciones conocidas como "Condiciones de Dirichlet", las cuales nos dicen que las series de Fourier deben ser finitas, es decir, que la función de la que partimos debe ser integrable. La primera condición nos dice que cualquier integral entre dos puntos del módulo de la función tiene que ser menor que infinito, y la otra condición, que en una integral entre menos infinito y más infinito del módulo de la función debe tener un resultado menor que infinito.[4]

Este tipo de transformada tiene muchas aplicaciones en ingeniería, ya que se pueden transformar señales de potencia y energía para poder así enviar información por ondas. Esta transformada nos permite ocupar todo el espectro radioeléctrico.

La principal diferencia que existe entre estas 2 transformadas matemáticas es, que con la transformada Wavelet podemos obtener información muy relevante que no aparece en los

[3] **Series de Fourier**. [Chirinos,2019]
[4] **Series de Fourier y sus aplicaciones**.

[Goñi Ibaceta,2021]

datos brutos del paquete original, además de poder reducir o eliminar el ruido pasando a considerarse de una manera despreciable.

4.2 Transformada Haar

Se encuentra dentro del análisis wavelet discreto y es de gran utilidad debido a su simpleza y uso a la hora de reducir información. Se parte de los datos brutos de una señal discreta, y, se calculan los coeficientes de aproximación gracias a la media de dos muestras consecutivas, consiguiendo así la mitad de los puntos, es por ello necesario que el número de muestras sea potencia de 2. Este tipo de transformada descompone una señal discreta en 2 partes, en coeficientes de aproximación, que son los encargados de dar forma a la onda, y en coeficientes de detalles, en donde aparecen las frecuencias.[5]

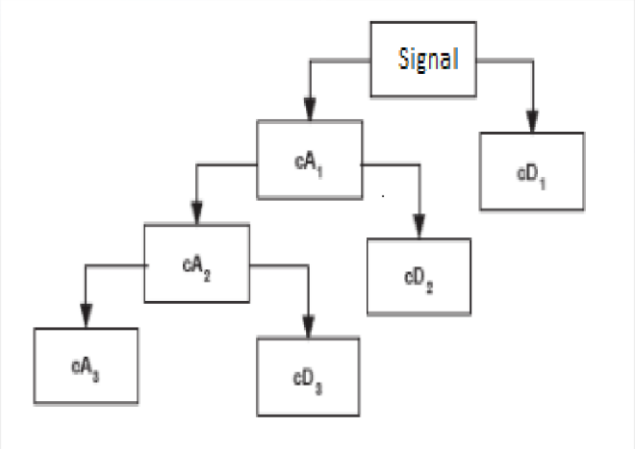


Fig. 3. Descomposición Transformada Haar

El factor √2 es de gran importancia en esta transformada, no solo por ser necesario en la multiplicación de las muestras, sino porque mantiene y compacta la energía de la señal discreta original en los coeficientes mencionados previamente, gracias a ello, se le puede hacer la inversa a la señal original, lo cual nos permite la representación de la señal con menos puntos, pero sin perder prácticamente información visual.[6]

[5] **Introducción a la teoría de wavelets. Construcción de propiedades de wavelets continuas y discretas.** [Martín Martín, 2019]

[6] **Selección de características para el reconocimiento de patrones con datos de alta dimensionalidad en fusión nuclear.** [Pereira,2015]

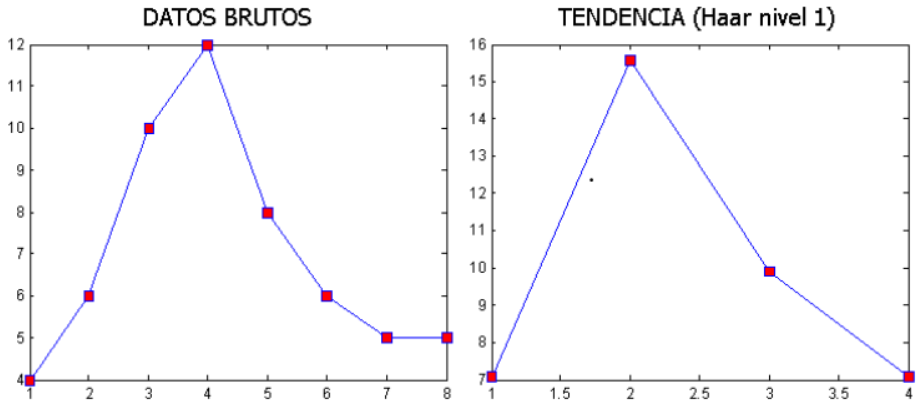


Fig. 4. Comparativa Transformada Haar

Cuantos más coeficientes Haar se apliquen, mayor será la reducción de los datos y menor el intervalo de tiempo entre ellos. Cabe puntualizar que utilizar los máximos coeficientes posibles no es una solución óptima, ya que debido a la reducción de datos se debe mantener una resolución mínima aceptable. Al final las señales original y reducida no serán al 100% iguales, pero de ahí que hablemos de similitud entre ellas y de conservación de información.[7]

[7] **Wavelets de Haar y Daubechies y sus aplicaciones.** [Olivera,2018]

5 Recopilación de datos

Se han utilizado para reducir los datos de entrada más relevantes para poder realizar así un procesamiento y análisis adecuado. Para ello, se han utilizado herramientas como los algoritmos de clasificación, los cuales mediante sus predicciones nos ayudan a ponderar los valores estudiados.

Por otra parte, cabe puntualizar que en las variables independientes no existe la dependencia lineal ni la igualdad entre ellas. Cuando una variable independiente tiene una alta correlación con otra u otras, se dice que es una combinación lineal de varias, y esto está catalogado como multicolinealidad.

El análisis de los datos consiste en la síntesis de la información, reduciendo su tamaño, consiguiendo así la cantidad exacta de variables transformadas manteniendo dentro de lo que cabe, toda la información posible.

A la hora de la recopilación de datos, es necesario el conocimiento de muchos parámetros, los cuales se medirán mediante sensores, también conocidos como diagnósticos, que transforman la magnitud física o variable en tensión eléctrica. Se convierte una señal analógica en digital. Por tanto, en este sistema de recopilación de datos, se toman los datos, se tratan y se transforman para poder procesarlos a ordenador, además, se almacenará, recuperará y visualizará toda la información.

Es por ello, que la interpolación de valores y la discretización de información cobra mucha importante, ya que, interpolando podemos conocer el valor que toma un dato desconocido dentro de un conjunto, y, discretizando, podemos transformar los datos numéricos en categóricos, dividiéndolos así en subconjuntos.

6 Creación de la base de datos

Se ha partido desde una base en la que se ha realizado un ciclo de medidas en el TJ-II, gracias a todos los sensores que tiene y que han sido capaces de captar todas las señales que posteriormente se han parametrizado. En nuestro caso, nuestro archivo de datos está compuesto por 494 archivos ASCII, de 30-40 megabytes cada archivo, lo que cual forma un total de 21 gigabytes de datos brutos.

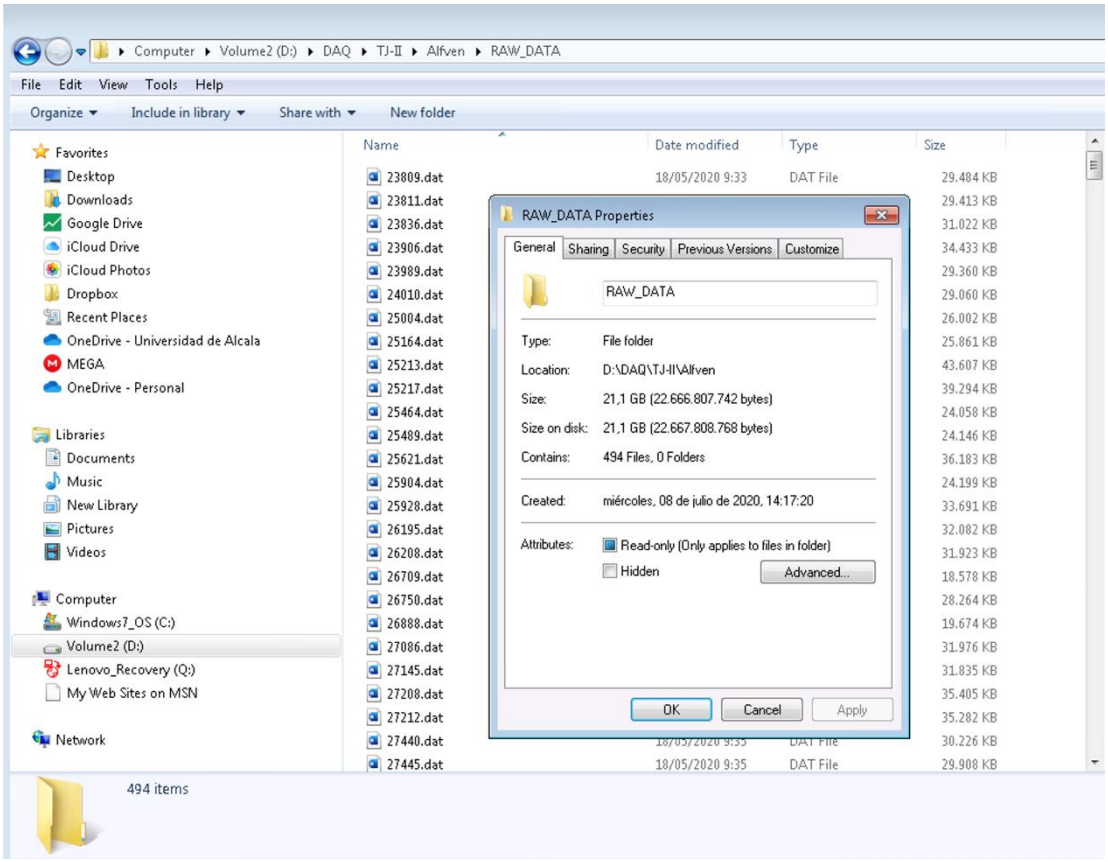
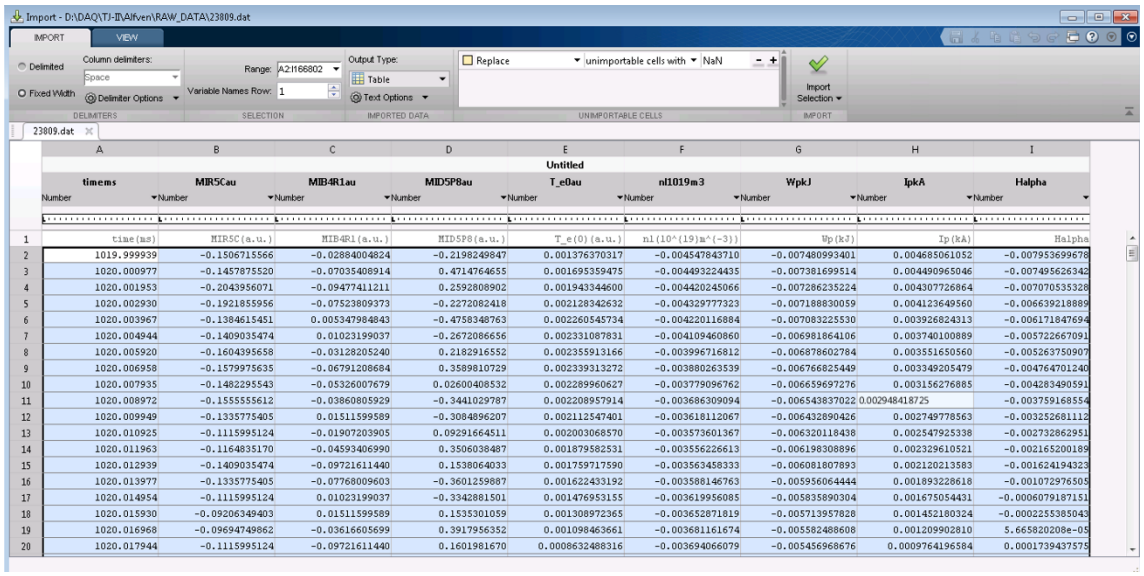


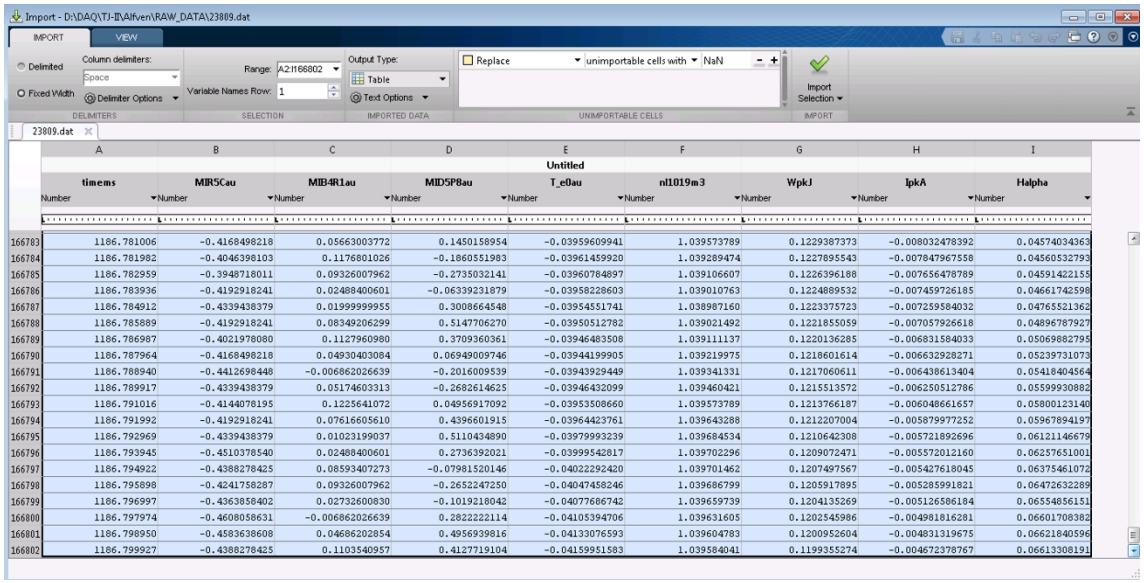
Fig. 5. Memoria Base de datos

Cuando se han traspasado todos los datos a Matlab, los 21 gigabytes de datos brutos forman una estructura la cual está compuesta por 166000 muestras por cada una de las descargas, cada señal está compuesta por 8 señales que hacen que la descarga sea única.



	A	B	C	D	E	F	G	H	I
	timems	MIR5Cau	MIB4R1au	MID5P8au	T_e0au	nl1019m3	WpkJ	IpkA	Halpha
	Number	Number	Number	Number	Number	Number	Number	Number	Number
1	time(ms)	MIR5C(a.u.)	MIB4R1(a.u.)	MID5P8(a.u.)	T_e(0)(a.u.)	nl(10^(19)m^(-3))	Wp(kJ)	Ip(kA)	Halpha
2	1019.999939	-0.1506715566	-0.02884004824	-0.2198249847	0.001376370317	-0.004547843710	-0.007480993401	0.004685061052	-0.007953699678
3	1020.000977	-0.1457875520	-0.07035408914	0.4714764655	0.001695359475	-0.004493224435	-0.007381699514	0.004490965046	-0.007495626342
4	1020.001953	-0.2043956071	-0.09477411211	0.2592808902	0.001943344600	-0.004420245066	-0.007286235224	0.004307726864	-0.007070535328
5	1020.002930	-0.1921855956	-0.07523809373	-0.2272082418	0.002128342632	-0.004329777323	-0.007188830059	0.004123649560	-0.006639218889
6	1020.003967	-0.1384615451	0.005347984843	-0.4758348763	0.002260545734	-0.004220116884	-0.007083225530	0.003926824313	-0.006171847694
7	1020.004944	-0.1409035474	0.01023199037	-0.2672086656	0.002331087831	-0.004109460860	-0.006981864106	0.003740100889	-0.005722667091
8	1020.005920	-0.1604395658	-0.03128205240	0.2182916552	0.002355913166	-0.003996716812	-0.006878602784	0.003551650560	-0.005263750907
9	1020.006958	-0.1579975635	-0.06791208684	0.3589810729	0.002339313272	-0.003880263539	-0.006766825449	0.003349205479	-0.004764701240
10	1020.007935	-0.1482295543	-0.05326007679	0.02600408532	0.002289960627	-0.003779096762	-0.006659697276	0.003156276885	-0.004283490591
11	1020.008972	-0.1555555612	-0.03860805929	-0.3441029787	0.002208957914	-0.003686309094	-0.006543837022	0.002948418725	-0.003759168554
12	1020.009949	-0.1335775405	0.01511599589	-0.3084896207	0.002112547401	-0.003618112067	-0.006432890426	0.002749778563	-0.003252681112
13	1020.010925	-0.1115995124	-0.01907203905	0.09291664511	0.002003068570	-0.003573601367	-0.006320118438	0.002547925338	-0.002732862951
14	1020.011963	-0.1164835170	-0.04593406990	0.3506038487	0.001879582531	-0.003556226613	-0.006198308896	0.002329610521	-0.002165200189
15	1020.012939	-0.1409035474	-0.09721611440	0.1538064033	0.001759717590	-0.003563458333	-0.006081807893	0.002120213583	-0.001624194323
16	1020.013977	-0.1335775405	-0.07768009603	-0.3601259887	0.001622433192	-0.003588146763	-0.005956064444	0.001893228618	-0.001072976505
17	1020.014954	-0.1115995124	0.01023199037	-0.3342881501	0.001476953155	-0.003619956085	-0.005835890304	0.001675054431	-0.000607918715
18	1020.015930	-0.09206349403	0.01511599589	0.1535301059	0.001308972365	-0.003652871819	-0.005713957828	0.001452180324	-0.000225538504
19	1020.016968	-0.09694749862	-0.03616605699	0.3917956352	0.001098463661	-0.003681161674	-0.005582488608	0.001209902810	5.665820208e-05
20	1020.017944	-0.1115995124	-0.09721611440	0.1601981670	0.0008632488316	-0.003694066079	-0.005456968676	0.0009764196584	0.0001739437575

Fig. 6. Inicio base de datos



	A	B	C	D	E	F	G	H	I
	timems	MIR5Cau	MIB4R1au	MID5P8au	T_e0au	nl1019m3	WpkJ	IpkA	Halpha
	Number	Number	Number	Number	Number	Number	Number	Number	Number
166783	1186.781006	-0.4168498218	0.05663003772	0.1450158954	-0.03959609941	1.039573789	0.1229387373	-0.008032478392	0.04574034363
166784	1186.781982	-0.4046398103	0.1176801026	-0.1860551983	-0.03961459920	1.039289474	0.1227895543	-0.007847967558	0.04560532793
166785	1186.782959	-0.3948718011	0.09326007962	-0.2735032141	-0.03960784897	1.039106607	0.1226396188	-0.007656478789	0.04591422155
166786	1186.783936	-0.4192918241	0.02488400601	-0.06339231879	-0.03958228603	1.039010763	0.1224889532	-0.007459726185	0.04661742598
166787	1186.784912	-0.4339438379	0.01999999955	0.3008664548	-0.03954551741	1.038987160	0.1223375723	-0.007259584032	0.04765521362
166788	1186.785889	-0.4192918241	0.08349206299	0.5147706270	-0.03950512782	1.039021492	0.1221855059	-0.007057926618	0.04896787927
166789	1186.786987	-0.4021978080	0.1127960980	0.3709360361	-0.03946483508	1.039111137	0.1220136285	-0.006831584033	0.05069882795
166790	1186.787964	-0.4168498218	0.04930403084	0.06949009746	-0.03944199905	1.039219975	0.1218601614	-0.006632928271	0.05239731073
166791	1186.788940	-0.4412698448	-0.006862026639	-0.2016009539	-0.03943929449	1.039341331	0.1217060611	-0.006438613404	0.05418404564
166792	1186.789917	-0.4339438379	0.05174603313	-0.2682614625	-0.03946432099	1.039460421	0.1215513572	-0.006250512786	0.05599930882
166793	1186.791016	-0.4144078195	0.1225641072	0.04956917092	-0.03953508660	1.039573789	0.1213766187	-0.006048661657	0.05800123140
166794	1186.791992	-0.4192918241	0.07616605610	0.4396601915	-0.03964423761	1.039643288	0.1212207004	-0.005879977252	0.05967894197
166795	1186.792969	-0.4339438379	0.01023199037	0.5110434890	-0.03979993239	1.039684534	0.1210642308	-0.005721892696	0.06121146679
166796	1186.793945	-0.4510378540	0.02488400601	0.2736392021	-0.03999542817	1.039702296	0.1209072471	-0.005572012160	0.06257651001
166797	1186.794922	-0.4388278425	0.08593407273	-0.07981520146	-0.04022292420	1.039701462	0.1207497567	-0.005427618045	0.06375461072
166798	1186.795898	-0.4241758287	0.09326007962	-0.2652247250	-0.04047458246	1.039686799	0.1205917895	-0.005285991821	0.06472632289
166799	1186.796997	-0.4363858402	0.02732600830	-0.1019218042	-0.04077686742	1.039659739	0.1204135269	-0.005126586184	0.06554856151
166800	1186.797974	-0.4608058631	-0.006862026639	0.2822222114	-0.04105394706	1.039631605	0.1202545986	-0.004981816281	0.06601708382
166801	1186.798950	-0.4583638608	0.04686202854	0.4956939816	-0.04133076593	1.039604783	0.1200952604	-0.004831319675	0.06621840596
166802	1186.799927	-0.4388278425	0.1103540957	0.4127719104	-0.04159951583	1.039584041	0.1199355274	-0.004672378767	0.06613308191

Fig. 7. Final base de datos

7 Preprocesamiento de señales

Una vez obtenida la base de datos en bruto, ya podemos trabajar con ella, para ello, se han utilizado una secuencia de procesos los cuales no podrían ser ejecutados sin el programa Matlab, por lo que sin la ayuda de este programa no podríamos tratar con estos datos.

```
[C L] = wavedec(Te, nivel, 'haar');
    caproxima4 = appcoef(C,L,'haar');
    cdetalle4 = detcoef(C,L,nivel);
```

Fig. 8. Comandos coeficientes de aproximación y detalle

A partir de aquí, lo primero que se ha realizado es la transformada Wavelet-Haar, con la cual seremos capaces de reducir cada señal a 128 muestras, conservando la estructura de onda de la señal y quedándonos con los coeficientes de aproximación de la señal. Para ello, lo primero que haremos para empezar con el procedimiento es, convertir la base de datos en una potencia de 2, ya que esta transformada, se basa en el cálculo de la media agrupando los elementos en grupo de 2.[8]

El primer nivel que se ha encontrado es N/2 wavelets de Haar, los cuales se expresan de la siguiente forma:

$$\begin{aligned} w_1^1 &:= (\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, \dots, 0) \\ w_2^1 &:= (0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, \dots, 0) \\ &\vdots \\ w_{N/2}^1 &:= (0, \dots, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}). \end{aligned}$$

Fig. 9. Primer nivel Wavelet Haar

El conjunto de vectores que obtenemos forma un sistema ortogonal, tomando como referencia que cada uno de ellos son ortogonales.

$$\begin{aligned} w_1^1 \cdot w_2^1 &= (\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, \dots, 0) \cdot (0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, \dots, 0) = \frac{1}{\sqrt{2}} \cdot 0 + \frac{-1}{\sqrt{2}} \cdot 0 + \dots + 0 = 0 \\ w_1^1 \cdot w_3^1 &= (\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, \dots, 0) \cdot (0, 0, 0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, \dots, 0) = \frac{1}{\sqrt{2}} \cdot 0 + \frac{-1}{\sqrt{2}} \cdot 0 + \dots + 0 = 0 \\ &\vdots \\ w_{N/2-1}^1 \cdot w_{N/2}^1 &= (0, \dots, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0) \cdot (0, \dots, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}) = 0 + \dots + \frac{1}{\sqrt{2}} \cdot 0 + \frac{-1}{\sqrt{2}} \cdot 0 = 0. \end{aligned}$$

Fig. 10. Conjunto vectores sistema ortogonal

[8] **Simulación de señales en plasmas de fusión nuclear mediante técnicas de regresión paramétrica.** [Gilaberte, 2021]

Aunque hay que destacar que al realizar el cálculo de cada uno de los vectores que conforman el sistema su resultado es 1.

$$\| w_j^1 \| = \sqrt{(\frac{1}{\sqrt{2}})^2 + (\frac{-1}{\sqrt{2}})^2 + 0^2 + \cdots + 0^2} = \sqrt{\frac{1}{2} + \frac{1}{2}} = \sqrt{1} = 1$$

Fig. 11. Cálculo vector sistema

En cuanto se ha reducido el tamaño de la base de datos con la ayuda de las transformadas, nos damos cuenta de que presenta valores de amplitud tanto positivos como negativos. En nuestro caso, operar con valores negativos no tiene sentido, ya que vamos a trabajar con logaritmo de 10, y, para ello, no puede haber estos valores. Por tanto, se realiza una normalización de datos mediante el método del máximo y mínimo específico, el cual permite relacionar todos los valores de la base de datos comprendidos entre 1 y 10, ya que log10(1)=0 y log10(10)=1.

Finalmente, metemos todas las señales y todas las descargas en un solo archivo Matlab (3,5 Mb) con una estructura característica para mejor manejo y usabilidad.

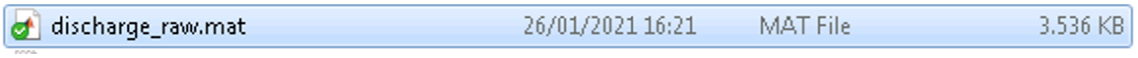


Fig. 12. Archivo Matlab reducido

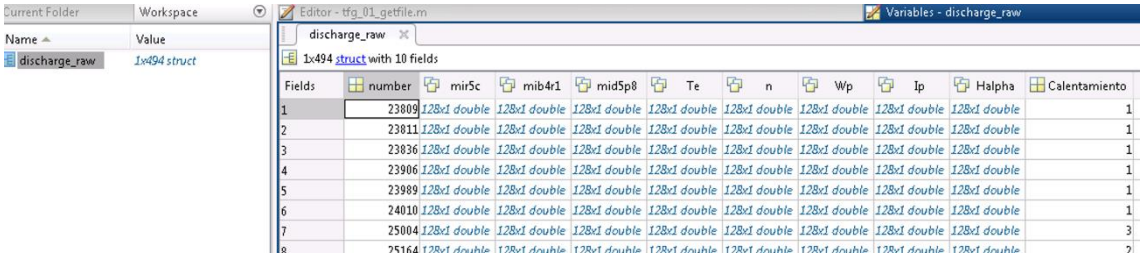


Fig. 13. Lista de descargas base de datos

8 Estudio de la base de datos

Como se ha comentado anteriormente, disponemos de una base de datos que está compuesta por 494 descargas, agrupadas en un documento formato struct llamado “discharge_raw3”, en el cual se encuentran todas las variables en matrices internas, las cuales nos permiten hacer una comparación entre ellas y aplicar programas específicos para su clasificación.

discharge_raw3

1x494 struct with 20 fields

Fields	number	mir5c	mib4r1	mid5p8	Te	n	Wp	Ip	Halpha
43	29206	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
44	29341	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
45	29443	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
46	29569	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
47	29577	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
48	29650	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
49	29807	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
50	29820	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
51	29838	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
52	29852	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
53	29867	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
54	29928	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
55	29929	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
56	29940	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
57	30011	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
58	30121	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
59	30123	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
60	30523	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
61	30846	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
62	30859	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
63	30900	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
64	30955	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
65	31063	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
66	31065	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...
67	31096	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...	128x1 dou...

Fig. 14. Lista de descargas en la base de datos.

Por otro lado, permite el estudio y análisis de todas las señales por descarga, estas señales son, de izquierda a derecha en la ilustración 14, mir5c, mib4r1, mid5p8, temperatura (Te), densidad (n), energía diamagnética (Wp), corriente de plasma (Ip) y Halpha. Por tanto, para poder realizar el análisis de una de estas señales en específico, se ha de tomar la columna entera de datos de cada una, y poder ver así su función para poder después estudiar y comentar los resultados.

8.1 Calentamiento NBI

La clasificación que se va a realizar en nuestro estudio será por calentamiento NBI, es decir, por inyección de neutros, para ello, dentro del struct mencionado anteriormente, existe una columna llamada "Clasi1" que nos muestra la clasificación de todas las descargas en la base de datos según este tipo de calentamiento, el número 1 implica que no existe calentamiento NBI y el número 2 que si, como se puede observar a continuación:

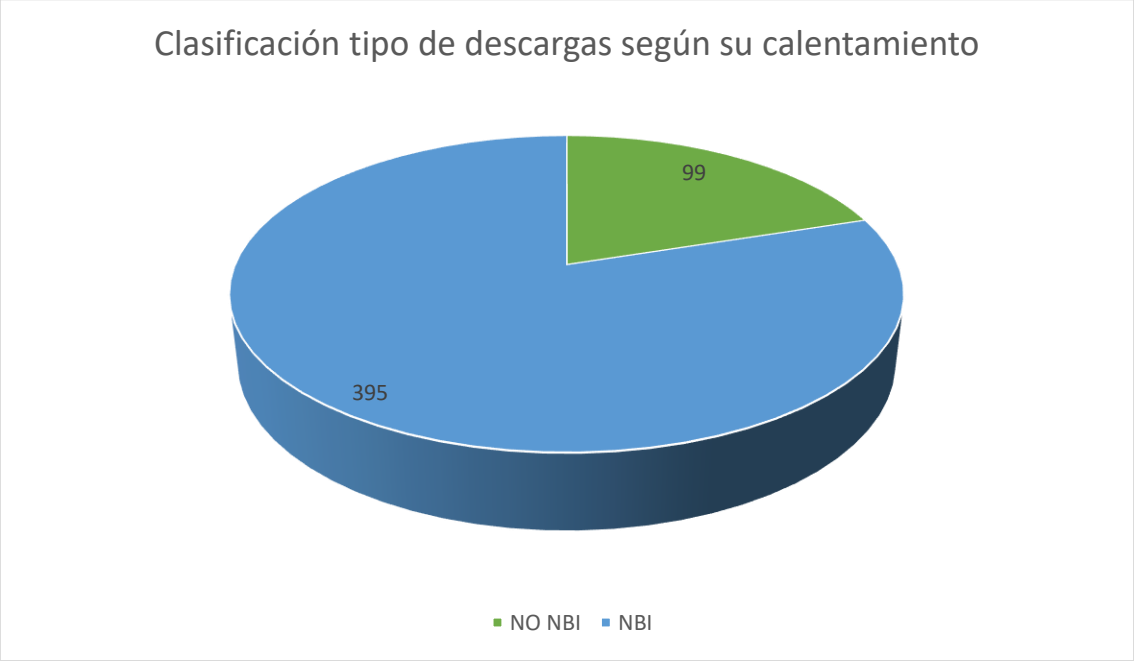


Fig. 15. Lista de clasificación NBI

El calentamiento NBI o por inyección de neutros consiste en la transferencia de un haz de partículas, específicamente de hidrógeno, a las cuales han sido aplicadas decenas de kiloelectronesvoltio (keV) y amperios a sus electrones e iones de plasma.

Las partículas neutras son muy importantes de neutralizar, ya que son las únicas partículas capaces de atravesar campos magnéticos intensos como el que estamos estudiando en el TJ-II. En el plasma, estos neutrones chocan con los electrones y los iones de tal forma que los ionizan, formando así iones rápidos. Lo que repercute al calentamiento del plasma drásticamente. El objetivo es conseguir haces de iones con densidades de corriente de 200 mA /cm^2 y manteniendo la calidad óptica del haz. Este calentamiento no depende de la configuración magnética.[9]

[9] **Transmisión del Haz de Neutros de Calentamiento en TJ-II.** [Fuentes López, 2007]

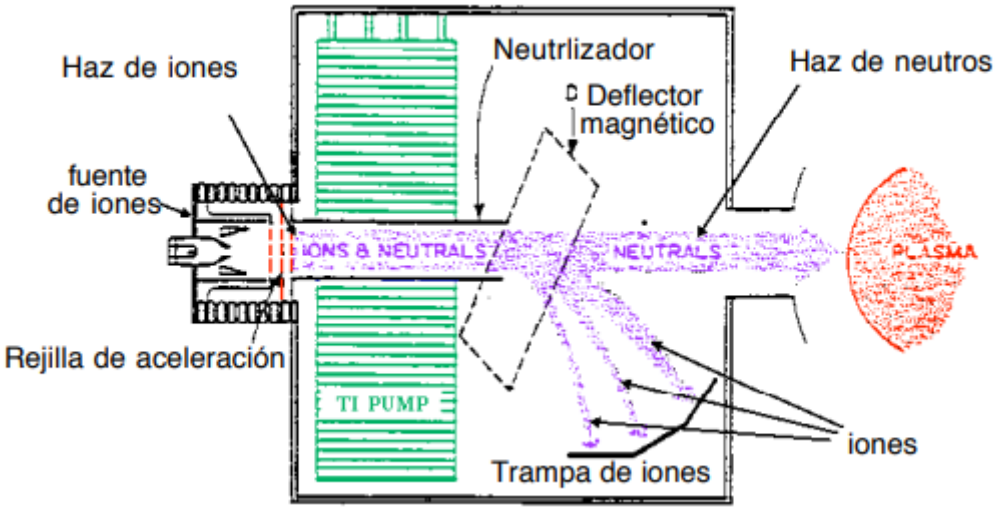


Fig. 16. Esquema inyector de neutros

Lo que hace tan particular a este tipo de calentamiento en el TJ-II es que, la estructura helicoidal de las configuraciones magnéticas hace que la inyección, transmisión y absorción, de la potencia del haz del que se está hablando sea realmente dificultosa en comparación con otros dispositivos de fusión termonuclear. Es por ello, que se ha requerido el diagnóstico de distintas variables para poder realizar esta inyección.[10]

8.2 Hiperplano y maximal margin classifier

Un hiperplano es un subespacio de dimensiones p-1 el cual no es necesario que pase por el origen, en un espacio de dos dimensiones se considera una recta y en uno de tres un subespacio de dos dimensiones.[11]

$$\beta 0 + \beta 1x1 + \beta 2x2 = 0 \qquad (8.2.1)$$

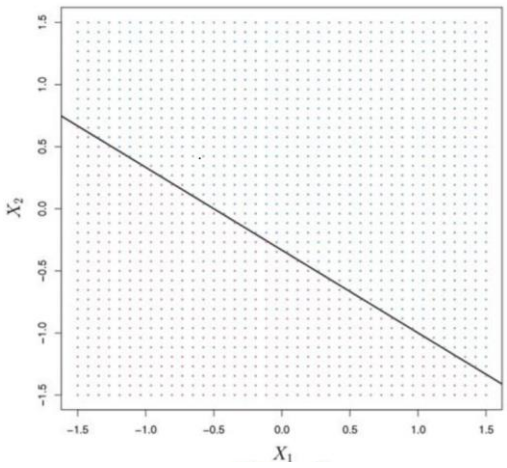


Fig. 17. Hiperplano dividiendo espacio

[10] **El proyecto de fusión nuclear ITER.** [ITER, 2015]
[11] **Support Vector Reggression: Propiedades y aplicaciones.** [Martín Guareño,2016]

Los parámetros beta y los pares de valores x son los encargados de cumplir la igualdad de puntos en el hiperplano.

$$\beta 0 + \beta 1x1 + \beta 2x2 + \cdots + \beta pxp = 0 \quad (8.2.2)$$

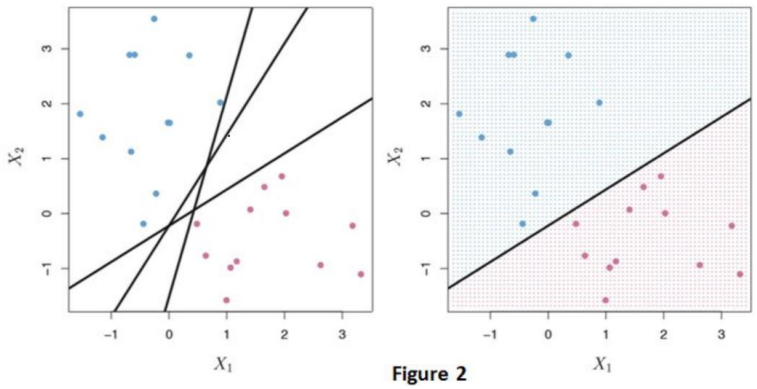


Fig. 18. Posibles hiperplanos separadores

Y todos sus puntos quedan definidos por el vector x, es por ello, que podemos afirmar que un hiperplano divide en dos mitades un espacio p-dimendional, para saber en qué lado se encuentra, solo habría que aplicar la ecuación.

$$\beta 0 + \beta 1x1 + \beta 2x2 + \cdots + \beta pxp < 0 \quad (8.2.3)$$

$$\beta 0 + \beta 1x1 + \beta 2x2 + \cdots + \beta pxp > 0 \quad (8.2.4)$$

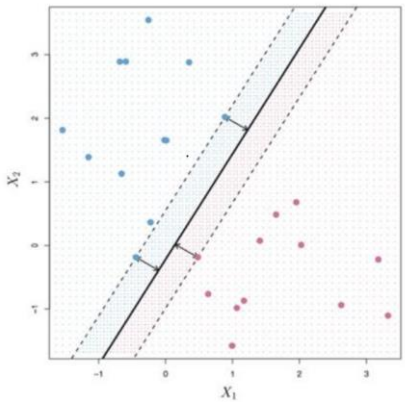


Fig. 19. Hiperplano margen máximo

En nuestro caso, disponemos de una gran cantidad de descargas, ordenadas según el tipo de variable, además de la oportunidad de clasificar esos datos según sus coeficientes de aproximación o de detalle. Gracias a todo esto se nos ha permitido predecir a que grupo pertenece cada uno según las observaciones que se han tomado.[12][Amat Rodrigo,2017]

[12] **Máquinas de Vector Soporte (Support Vector Machines, SVMs).** [Amat Rodrigo,2017]

9 Análisis previo mediante diagramas de dispersión

Antes de analizar las descargas en distintos programas de clasificación, es aconsejable hacer un análisis previo de cara a organizarnos un poco las ideas de a qué nos enfrentamos cuando hablamos de analizar bases de datos tan grandes. En los siguientes casos, se han realizado unas gráficas de dispersión con las distintas desviaciones estándar de las variables implicadas, es decir, como ya se ha comentado, cada descarga está compuesta por coeficientes de aproximación y de detalle, y, por tanto, lo que se ha realizado es la desviación estándar de cada uno de ellos en cada descarga y se ha reflejado en un diagrama de dispersión. Todo ello con una finalidad, reflejar la dificultad del análisis al que nos estamos enfrentando, lo cual nos sugiere que por muy buen análisis clasificatorio hagamos posteriormente, si los resultados no son los ideales o los mejores posibles sería lo normal, ya que la base de datos que estamos analizando es muy extensa y los datos están fuertemente desbalanceados. Comentar también que en las gráficas siguientes solo se tendrán en cuenta los puntos de color azul y verde, que son los que diferencian si hay calentamiento NBI o no.[13]

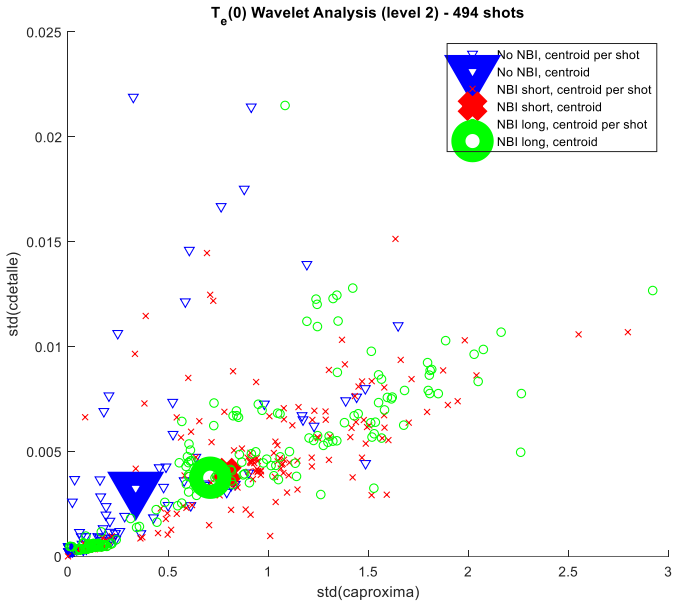


Fig. 20. Análisis señal Te del TJ-II

En esta primera gráfica, nos muestra el análisis de dispersión de la temperatura, se puede comprobar de una forma bastante nítida la separación entre los dos grupos. Mientras que los datos que implican que hay calentamiento NBI se encuentran muy compactos, los que no, se encuentran muy dispersos, lo cual nos ayuda a definir bien su centroide. Destacar también que cuando la desviación estándar de ambos coeficientes en próxima a cero los datos de superponen y es prácticamente imposible clasificarlos.

[13] **Diagrama de dispersión: Relación entre variables.** [Aiteco,2022]

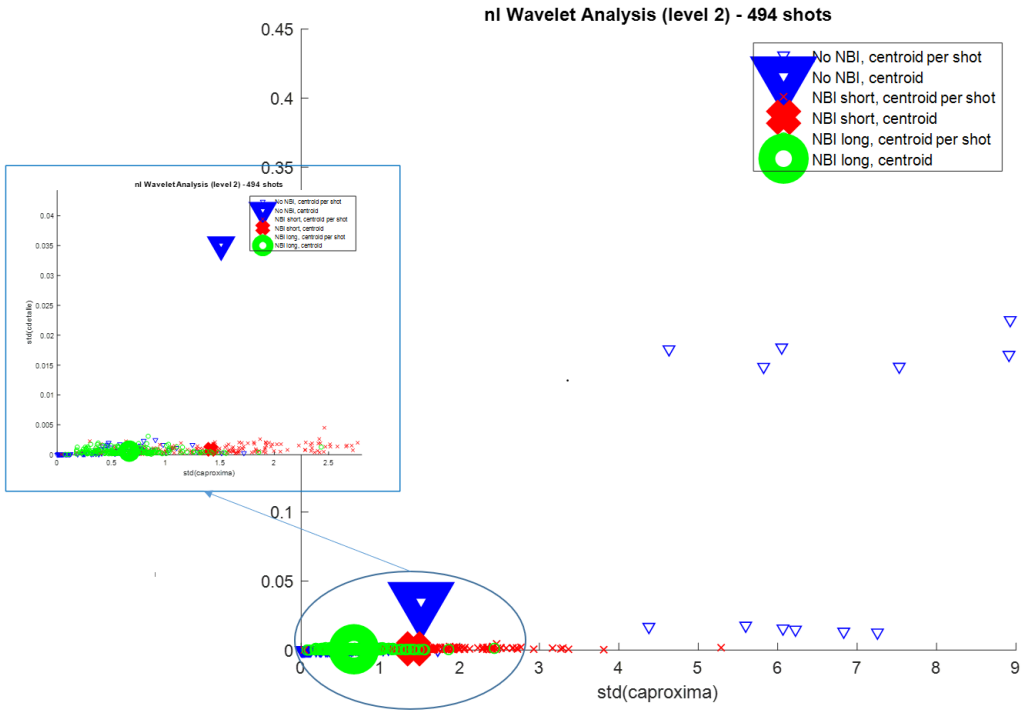


Fig. 21. Análisis señal n del TJ-II

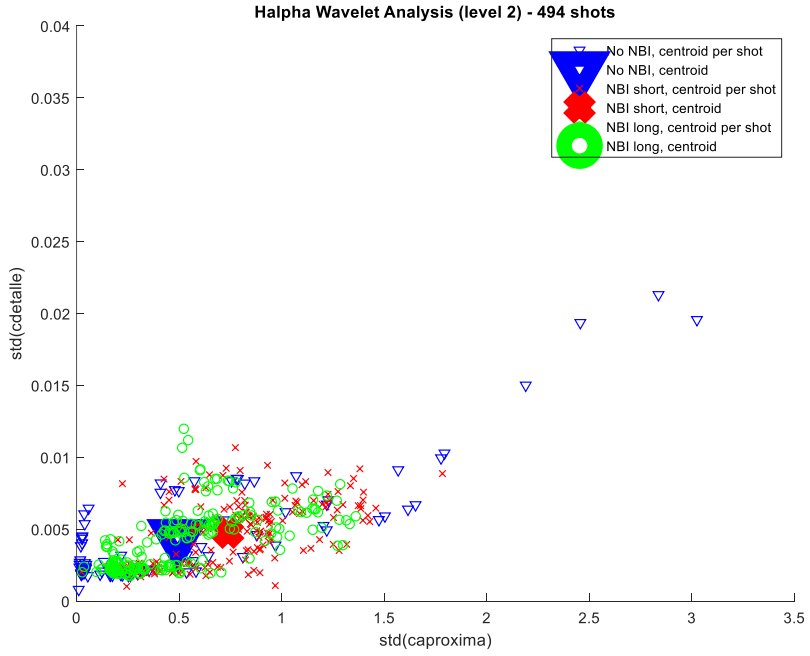


Fig. 22. Análisis señal Halpha del TJ-II

En este otro caso, es el caso más difícil que nos podemos encontrar a la hora de clasificar datos, porque como se puede comprobar, casi todos están superpuestos entre sí, y a la hora de clasificarlos resulta una tarea prácticamente imposible.

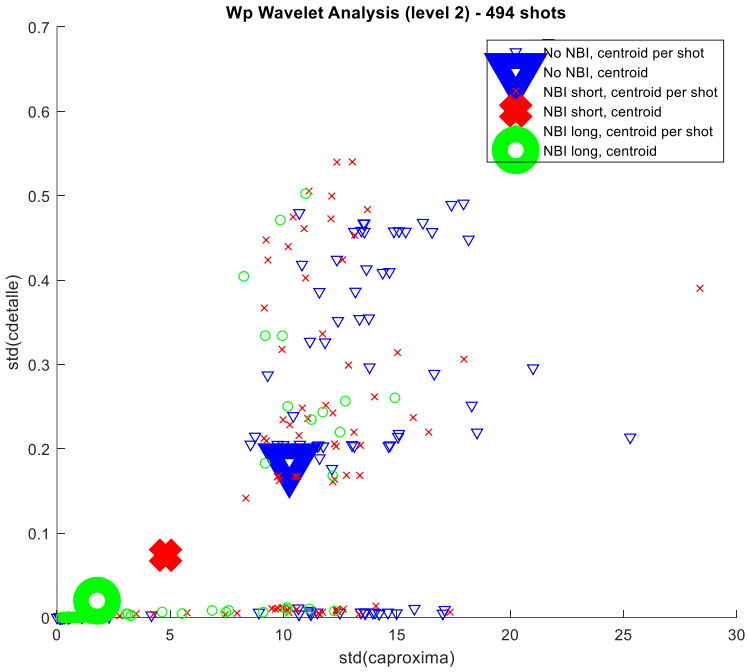


Fig. 23. Análisis señal Wp del TJ-II

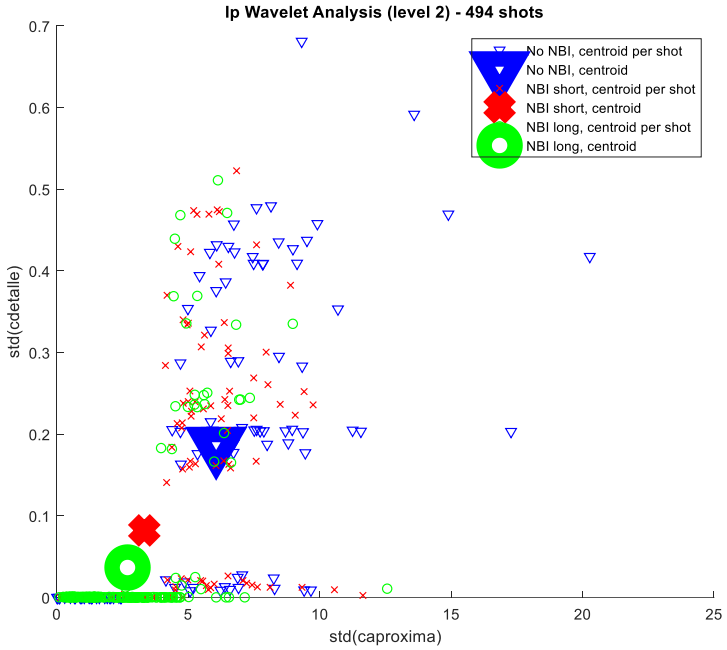


Fig. 24. Análisis señal Ip del TJ-II

Por otra parte, nos encontramos ante dos gráficos de dispersión distintos, unos que va por zonas, es decir, tenemos dos zonas claramente diferenciadas, una donde la desviación estándar de los coeficientes de detalle es casi 0 y es muy difícil clasificar los datos, y otra donde los datos están bastante bien distribuidos y podemos diferenciar con claridad los dos grupos en los cuales queremos clasificar los datos. La idea que se puede sacar de este gráfico es que no es suficientemente eficiente una clasificación de datos en las que solo se pueden clasificar con claridad el 50% de ellos aproximadamente.

10 Estructura y descripción de los programas de clasificación

Para poder clasificar todas las descargas de cada una de las variables mencionadas, se han utilizado distintos programas, realizados en Matlab, cada uno de ellos nos va a permitir plantearnos la tasa de acierto gracias a la utilización de descargas como test, y así, poder predecir si tiene calentamiento NBI o no. Todos los programas tienen sus particularidades, y, es por ello, que dependiendo del estudio que se ha realice, se va a requerir más la ayuda de uno que de otro.

10.1 SVM Linear

También conocido como Support Vector Classifiers se ha utilizado debido a que se encuentran pocos casos en los que, al clasificar los datos, sean perfectos y linealmente separables. Es más robusto, pero al aplicar nuevas predicciones va a tener una mejor capacidad predictiva.

svm_linear.m | svm_polynomial.m | svm_rbf.m | knn_2.m | knn_6.m | discriminant_linear.m | discriminant_quadratic.m

```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasi1];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateSVM('Standardize',true,'KernelFunction','linear');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);


    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasi1];
    end;
```

Fig. 25. Primera parte código programa SVM Linear

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/length(labels);
    SR_NBIfalso(k) = FP*100/length(labels);
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('SVM linear');
```

Fig. 26. Segunda parte código SVM Linear

Mediante este programa se clasifican de manera excelente la mayoría de los datos, los pocos que no se consiguen es debido a la optimización convexa. Existe un hiperparámetro dentro de este tipo de programa llamado "tuning", el cual controla las veces que se viola el margen del hiperplano, cuanto más próximo esté este parámetro a 0, más errores va a cometer, y, por tanto, más datos erróneos encontraremos, es el encargado del balanceo de los datos que se procesen. Todos los datos que se encuentren en el margen del hiperplano se les considerará vectores soporte, y son los que definirán perfectamente el clasificador utilizado.

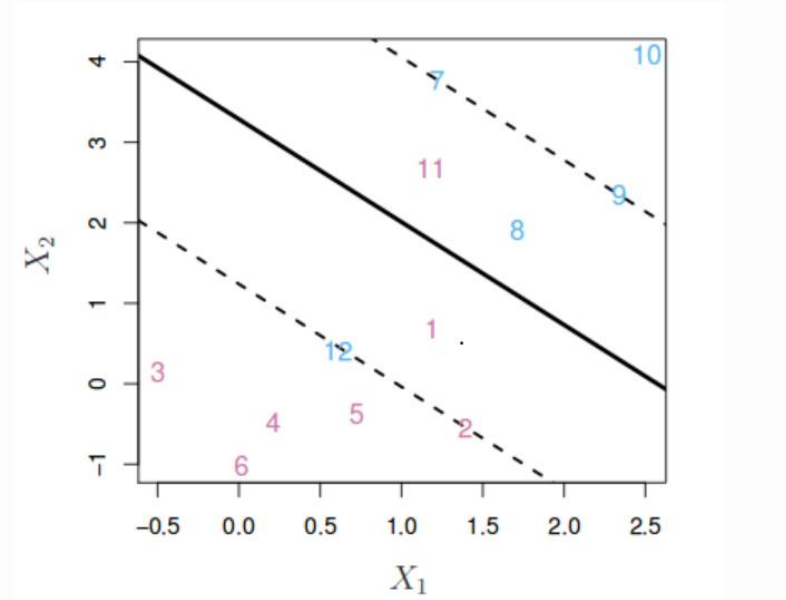


Fig. 27. Clasificador vector soporte

10.2 SVM Polynomial

Se trata de un método basado en la teoría de los polinomios ortogonales, es decir, de la localización de sus ceros, puntos críticos y asíntotas. Y es así debido a que todos estos parámetros son de gran utilidad en la interpolación, las fórmulas cuadráticas y las aproximaciones racionales y electrostáticas.

```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasi1];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateSVM('Standardize',true,'KernelFunction','polynomial');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);

    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasi1];
    end;

    labels = predict(Mdl,Xtest);
```

Fig. 28. Primera parte código SVM Polynomial

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/length(labels);
    SR_NBIfalso(k) = FP*100/length(labels);
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('SVM polynomial');
```

Fig. 29. Segunda parte código SVM Polynomial

Un polinomio de grado igual o mayor a 0 se representa de la siguiente manera:

$$p_n(x) = k_n x^n + \cdots + k1x + k0 \in P \qquad (10.2.1)$$

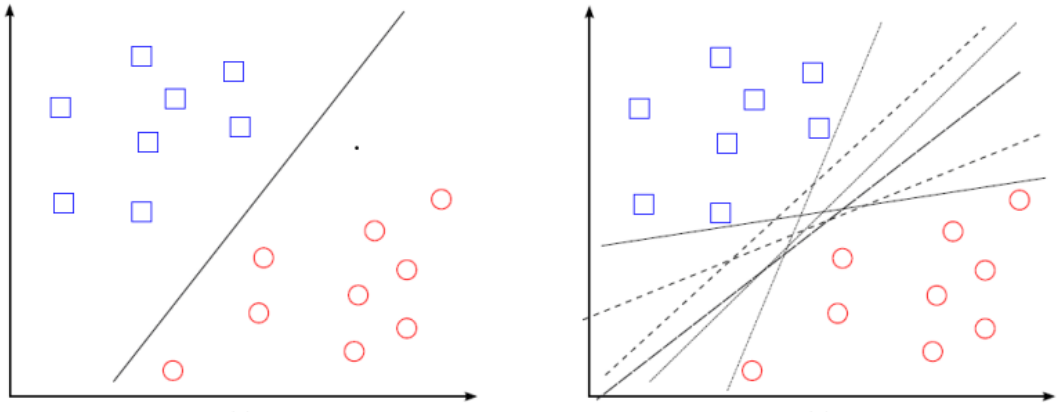


Fig. 30. Hiperplanos de separación en un espacio bidimensional

Los coeficientes k son los coeficientes reales, mientras que la P es el espacio lineal de los polinomios con esos coeficientes. El coeficiente kn se le conoce como coeficiente líder, ya que si es igual a 1 el polinomio es Mónico. También se considera una secuencia de polinomios ortogonal, ya que lo es respecto a la función no negativa en un intervalo acotado.[14]

$$\int_E Pn(x)x^m\omega(x)dx = \begin{cases} \neq 0, si\ m = n \\ = 0, si\ m < n \end{cases} \qquad (10.2.2)$$

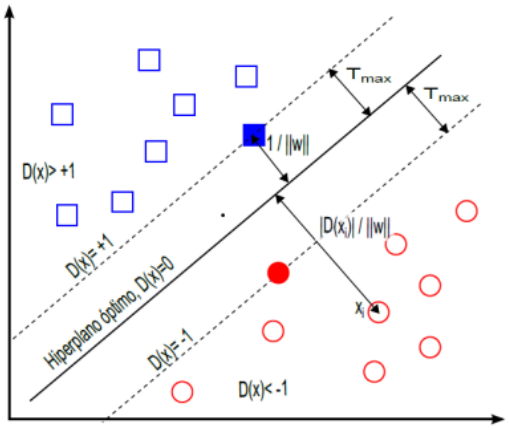


Fig. 31. Hiperplano de separación óptimo

La particularidad de este método es, que permite relacionar las secuencias de polinomios entre sí, de forma que se puedan extraer conclusiones e información valiosa para poder clasificar los datos analizados.[15]

[14] **Introducción a las máquinas de vector soporte (SVM) en aprendizaje supervisado.** [Campo León,2016]
[15] **Máquinas de soporte vectorial con núcleos de polinomios ortogonales para problemas de clasificación.** [Benayas Alamos,2018]

10.3 SVM RBF

Este tipo de programa está definido por los parámetros gamma y c del kernel de SVM en función de la base radial RBF. El parámetro gamma representa la distancia a la que pueden llegar los parámetros de entrenamiento, que también pueden representarse como el inverso del radio de influencia por el modelo SVM. Y el parámetro c es el encargado de compensar la clasificación de los parámetros de entrenamiento teniendo en cuenta el margen de la función, cuanto menor sea c, mayor será el margen y menor la precisión de la función.


```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasil];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateSVM('Standardize',true,'KernelFunction','rbf');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);

    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasil];
    end;

    labels = predict(Mdl,Xtest);
```

Fig. 32. Primera parte código SVM RBF

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/length(labels);
    SR_NBIfalso(k) = FP*100/length(labels);
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('SVM RBF');
```

Fig. 33. Segunda parte código SVM RBF

Este método depende mucho de gamma, ya que, si es muy grande, el radio del área de SVM solo incluye el vector soporte y ningún valor de c podrá regularizar este desajuste. Y cuando es muy pequeño no logra capturar de forma correcta todos los datos ya que quedaría demasiado restringido. Este modelo trabaja como los anteriores, toma unos datos como entrenamiento, y los restantes los utiliza de test para así poder crear una tasa de acierto. Finalmente, aunque lo buscado es un valor intermedio de gamma, no todos los valores cuentan, ya que a partir de cierto punto el rendimiento no cambia, y eso quiere decir que por mucho que cambiemos el valor de c, el vector soporte tampoco.[16]

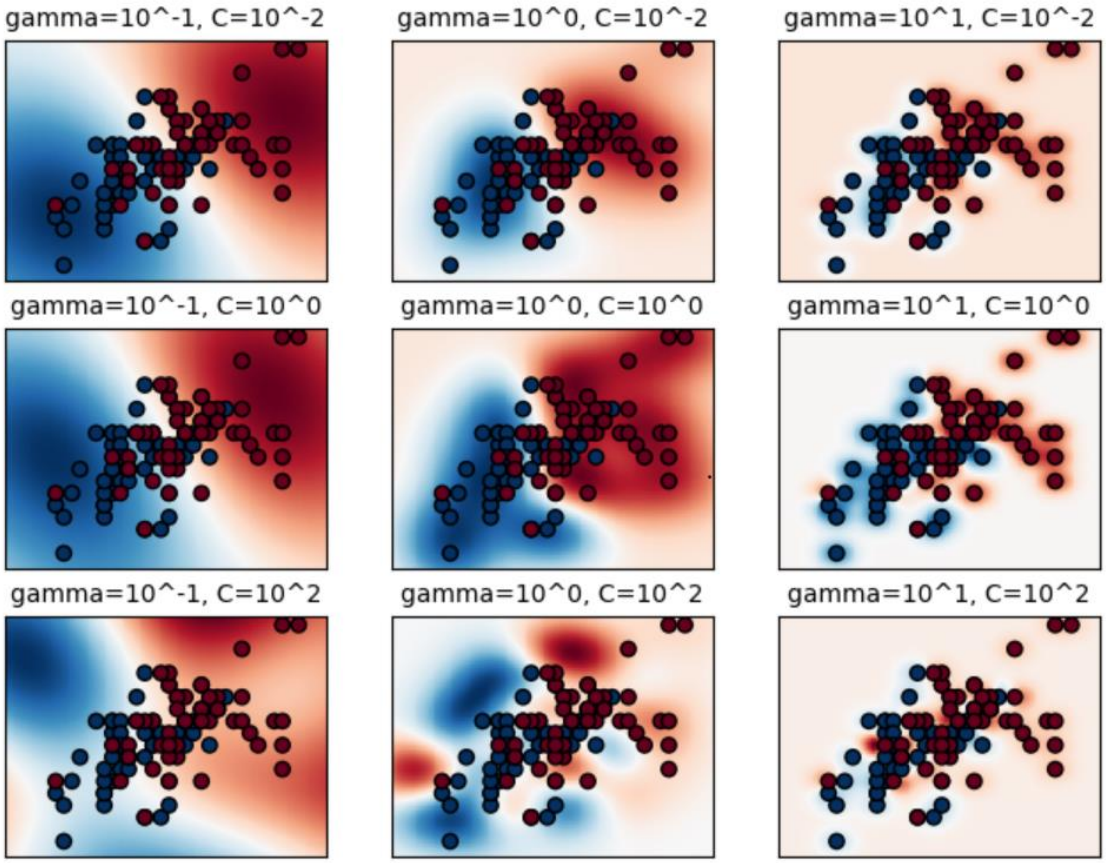


Fig. 34. Mapa de calor de la precisión de gamma y c

[16] **Parámetro RBF SVM.** [Scitik Internet,2007]

10.4 KNN

Otros programas que utilizaremos para el análisis serán los KNN2 y KNN6, la particularidad de éstos es, que están fundamentados en la idea del vecino más próximo, es decir, clasificar cada dato en función de los que le rodean, este método de clasificación es muy simple e intuitivo y hace que su implementación sea muy cómoda.

```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasi1];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateKNN('NumNeighbors',2,'Standardize',1,'Distance','cosine');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);

    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasi1];
    end;

    labels = predict(Mdl,Xtest);
```

Fig. 35. Primera parte código KNN2

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/length(labels);
    SR_NBIfalso(k) = FP*100/length(labels);
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('KNN NumNeighbors=2, Distance = cosine');
```

Fig. 36. Segunda parte código KNN2

knn_6.m | discriminant_linear.m | discriminant_quadratic.m

```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasil];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateKNN('NumNeighbors',6,'Standardize',1,'Distance','cosine');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);

    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasil];
    end;

    labels = predict(Mdl,Xtest);
```

Fig. 37. Primera parte código KNN6

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/length(labels);
    SR_NBIfalso(k) = FP*100/length(labels);
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('KNN NumNeighbors=6, Distance = cosine');
```

Fig. 38. Segunda parte código KNN6

El algoritmo KNN básico consiste en un fichero de N casos con n variables predictoras y la clase C, que es la clase que se va a predecir. Se calcularán las distancias de todos los casos que ya han sido clasificados a un nuevo caso, que es el que se queremos clasificar, y, una vez seleccionados esos casos se le asignará un valor a la variable C más frecuente entre los objetos. Es decir, busca el vecino más cercano y filtra las probabilidades de pertenecer a dicha clase teniendo en cuenta las de los vecinos.[17]

		X_1	...	X_j	...	X_n	C
$(\mathbf{x}_1, c_1)$	1	x_{11}	...	x_{1j}	...	x_{1n}	c_1
	⋮	⋮		⋮		⋮	⋮
$(\mathbf{x}_i, c_i)$	i	x_{i1}	...	x_{ij}	...	x_{in}	c_i
	⋮	⋮		⋮		⋮	⋮
$(\mathbf{x}_N, c_N)$	N	x_{N1}	...	x_{Nj}	...	x_{Nn}	c_N
$\mathbf{x}$	$N+1$	$x_{N+1,1}$	...	$x_{N+1,j}$	...	$X_{N+1,n}$	?

Fig. 39. Notación del algoritmo KNN

En comparación con el resto de los programas clasificatorios, el KNN es un poco distinto, ya que el resto de los programas están basados en un proceso de inducción y posterior deducción del modelo clasificatorio, mientras que, en este clasificador, esos dos procesos están solapados, lo que se denomina transinducción.[18]

En caso de que se produzca un empate entre varias clases, existe una regla, conocida como regla heurística, la cual sirve para romper ese empate seleccionando la clase que más contente al vecino más próximo o seleccionando la que esté a menor distancia.

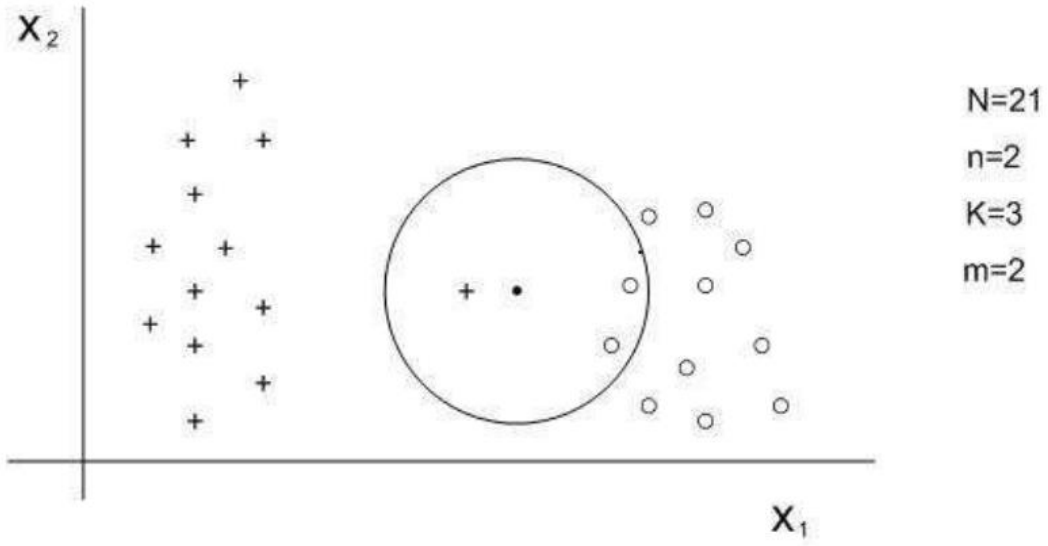


Fig. 40. Ejemplo de aplicación del algoritmo KNN

[17] **Clasificadores K-NN.** [Moujahid et al,2022]
[18] **Diseño de un algoritmo KNN aplicado a la detección de cáncer cerebral mediante imágenes espectrales.** [Bermejo, 2017]

10.5 Discriminant Linear

Se trata de un programa cuyo objetivo es explicar la influencia de varias variables cuantitativas sobre una cualitativa y, predecir si uno de los datos pertenece a una clase a partir de los datos registrados de las predicciones. Las variables dependientes son las que dependen de la clase a la que pertenecen y las independientes son las que designan a que grupo pertenecen.

Por tanto, su uso es, principalmente, saber las diferencias entre un grupo de variables pertenecientes a distintos grupos, seleccionar las variables predictoras que explican las diferencias entre grupos y establecer un proceso de clasificación a partir de las variables independientes. Gracias a ello, podremos pronosticar a donde pertenece cada tipo de dato y, saber que variables independientes tienen mayor discriminación y predicción a la hora de clasificar.

discriminant_linear.m | discriminant_quadratic.m

```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasil];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateDiscriminant('DiscrimType','linear');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);

    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasil];
    end;

    labels = predict(Mdl,Xtest);
```

Fig. 41. Primera parte código Discriminant Linear

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/length(labels);
    SR_NBIfalso(k) = FP*100/length(labels);
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('Regularized linear discriminant analysis (LDA) - Type: linear');
```

Fig. 42. Segunda parte código Discriminant Linear

Este tipo de programa se utiliza ya que los parámetros de regresión son verdaderamente inestables, en cambio, mediante este tipo de análisis, si mantenemos un tamaño muestral pequeño y una distribución de variables predictoras correcto, se consigue una mayor estabilidad que con la regresión, Además, es el método más común cuando se tienen más de 2 clases.[19]

Su procedimiento está basado en un número n de individuos con información de p variables, todo ello recogido en una variable y con dos o más categorías para saber a qué grupo pertenece cada dato. Todo ello obtiene como resultado una ecuación conocida como función discriminante, la cual se encarga de que haya una gran variabilidad entre los distintos grupos para así poder diferenciarlos mejor.[20]

[19] **Aplicación de técnicas de clasificación a la detección de cáncer.** [Cazorla Piñar,2019]
[20] **Análisis discriminante mediante SPSS.** [Torrado-Fonseca et al, 2013]

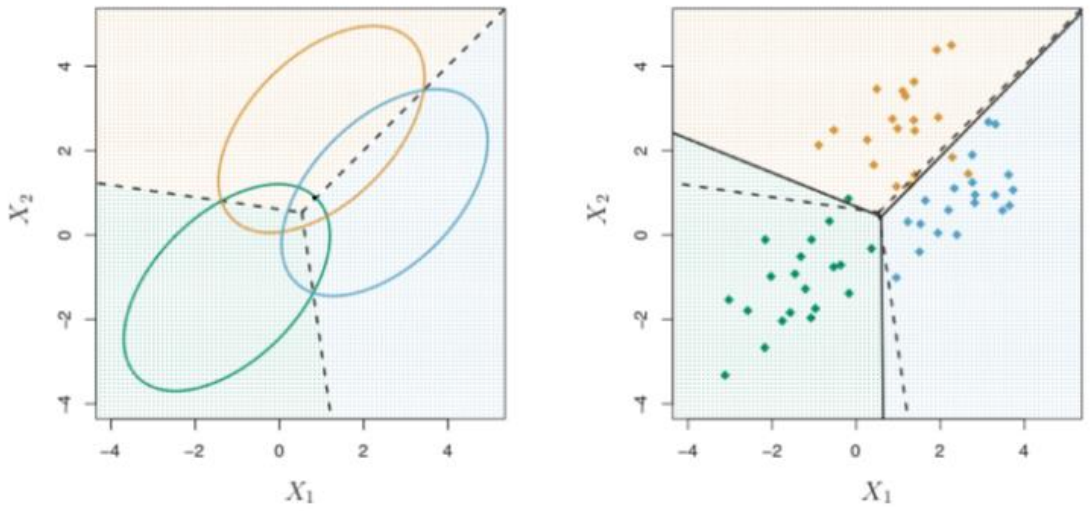


Fig. 43. Ejemplo discriminación Linear de 3 clases

Esta ecuación habrá cumplido con su función si consigue que se disminuyan los errores que puedan ocurrir a la hora de clasificar datos sin perder información sobre ellos.

$$Y = a0 + a1X1 + a2X2 + \cdots + apXp \qquad (10.5.1)$$

X = Variables independientes

a0 =constante

ap = coeficientes de discriminación

10.6 Discriminant Quadratic

El análisis discriminante, en este caso cuadrático, tiene sus semejanzas con el lineal, como que los datos se clasifican en función de la distancia al cuadrado más pequeña. Pero, en este caso, no se parte de la base que tienen matrices de covarianza iguales.

discriminant_quadratic.m

```
clear all;
load('discharge_raw3.mat');
discharge_raw = discharge_raw3;

for k=46:493;
k
    X = [];
    Y = [];
    shot = [];
    for i=1:k;
        X = [X; discharge_raw(i).Te', discharge_raw(i).n', discharge_raw(i).Wp', discharge_raw(i).Ip', discharge_raw(i).Halpha'];
%         X = [X; discharge_raw(i).Te_d', discharge_raw(i).n_d', discharge_raw(i).Wp_d', discharge_raw(i).Ip_d', discharge_raw(i).Halpha_d'];
        Y = [Y; discharge_raw(i).Clasi1];
    end;

    classOrder = unique(Y);
    rng(1); % For reproducibility
    t = templateDiscriminant('DiscrimType','quadratic');
    Mdl = fitcecoc(X,Y,'Learners',t,'ClassNames',classOrder);

    Xtest = [];
    Ytest = [];
    for fila=k+1:494;
        Xtest = [Xtest; discharge_raw(fila).Te', discharge_raw(fila).n', discharge_raw(fila).Wp', discharge_raw(fila).Ip', discharge_raw(fila).Halpha'];
%         Xtest = [Xtest; discharge_raw(fila).Te_d', discharge_raw(fila).n_d', discharge_raw(fila).Wp_d', discharge_raw(fila).Ip_d', discharge_raw(fila).Ha
        Ytest = [Ytest; discharge_raw(fila).Clasi1];
    end;

    labels = predict(Mdl,Xtest);
```

Fig. 44. Primera parte código Discriminant Quadratic

```
    end;

    labels = predict(Mdl,Xtest);
    numel(find((Ytest==labels)==1))*100/length(labels)
    SR_general(k) = numel(find((Ytest==labels)==1))*100/length(labels);
    TP=0;FN=0;TN=0;FP=0;
    for v=1:length(labels);
        if (Ytest(v)==2) && (labels(v)==2) TP=TP+1; end;
        if (Ytest(v)==2) && (labels(v)==1) FN=FN+1; end;
        if (Ytest(v)==1) && (labels(v)==1) TN=TN+1; end;
        if (Ytest(v)==1) && (labels(v)==2) FP=FP+1; end;
    end;
    SR_NBI(k) = TP*100/numel(find(Ytest==2));
    SR_noNBI(k) = TN*100/numel(find(Ytest==1));
    SR_NBIperdido(k) = FN*100/numel(find(Ytest==2));
    SR_NBIfalso(k) = FP*100/numel(find(Ytest==1));
end;
figure;
plot(SR_general);
hold on;
plot(SR_NBI);
plot(SR_noNBI);
legend('SR-general','SR-NBI','SR-noNBI');
xlabel('Size of (Training-left vs Test-right)');
ylabel('Success Rate');
title('Quadratic discriminant analysis (QDA) - Type: quadratic');
```

Fig. 45. Segunda parte código Discrimant Quadratic

La diferencia que existe entre ambas es que la cuadrática no es simétrica, es decir, genera límites de discusión curvos, por lo que la separación entre los distintos grupos no será lineal.[21][Gil Martín, 2018]

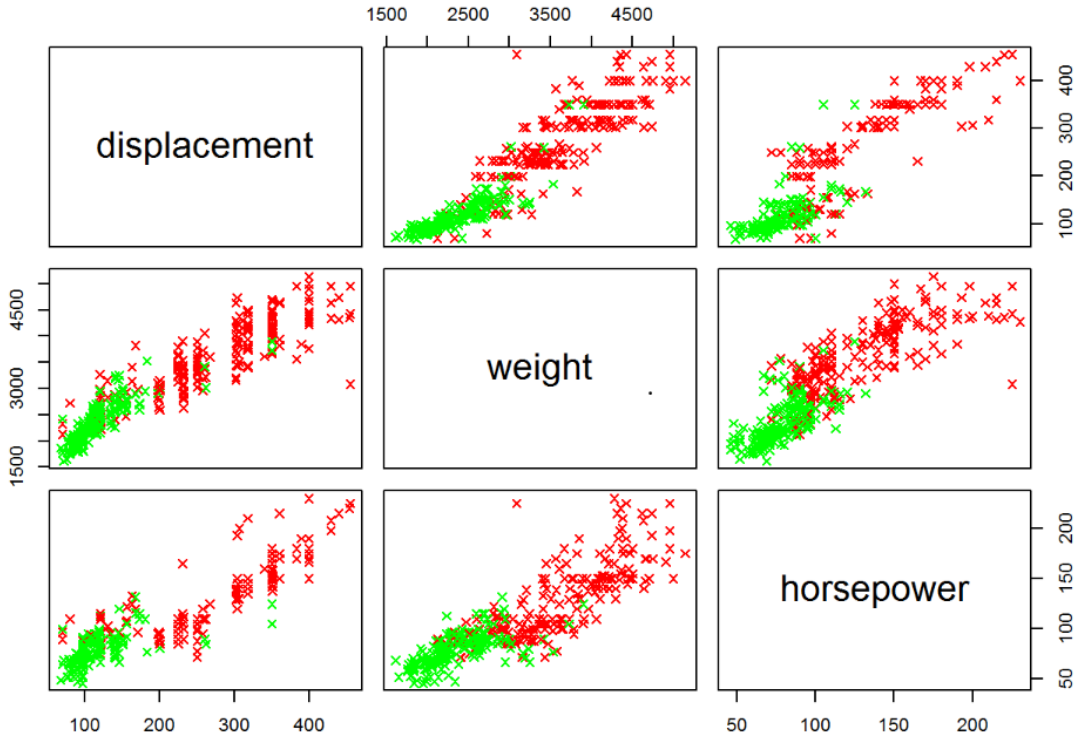


Fig. 46. Ejemplo discriminación lineal cuadrática

[21] **Análisis Discriminante Lineal y Cuadrático.** [Gil Martín, 2018]

11 Proceso de búsqueda de características relevantes

A partir de aquí, se han realizado gráficas con los distintos programas mencionados anteriormente, para poder visualizar de forma gráfica dos cosas, la diferencia de resultados si lo que utilizamos son coeficientes de aproximación o de detalle, y, la distinta información que se obtiene según el programa utilizado.

Además de haber comentado las diferencias entre las gráficas dependiendo del tipo se coeficiente, también se ha realizado un estudio si eliminas cada una de las variables, con el objetivo de obtener unos resultados más exactos.

11.1 Caso 1. Aplicación del clasificador SVM Linear

En este primer apartado se ha realizado el estudio con la ayuda del programa "SVM Linear".

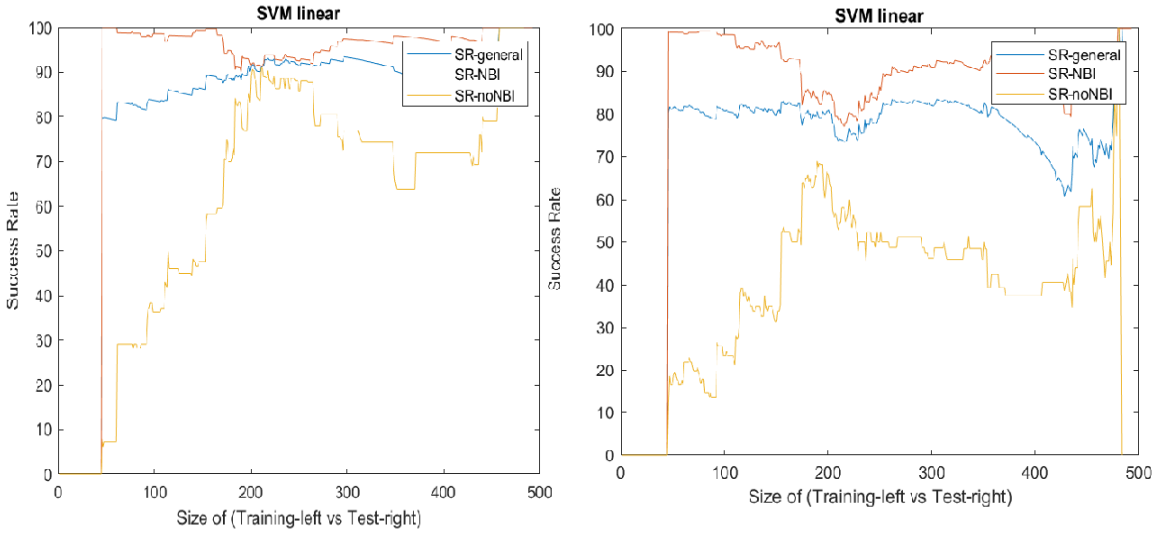


Fig. 47. Estudio SVM Linear. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se puede observar, en ambas gráficas comienza el estudio a partir de la descarga 46, y eso es debido a que es a partir de ahí cuando se produce el primer cambio de tipo de calentamiento, es decir, las primeras 45 descargas de la base de datos son de un tipo de calentamiento, en este caso, tipo 2, es decir, calentamiento NBI, inyección de neutros, y, a partir de la muestra 46 aparece el primero de tipo 1, sin calentamiento NBI, lo que implica que el estudio empieza en ese punto.

Del resto de la gráfica se pueden observar enormes diferencias, el estudio mediante coeficientes de aproximación es mucho mejor que el de coeficientes de detalle, ya que mediante el primero podemos llegar a la conclusión que, cuando se cogen unas 210 muestras como entrenamiento, la tasa de acierto es de un 90%, lo cual es un resultado más que aceptable a la hora que clasificar todas estas descargas. Aunque hay que tener en consideración, que, tanto para menos de 200 muestras de entrenamiento, como para más de 250, se encuentra fuertemente desbalanceada, lo cual sugiere que este tipo de estudio es válido si coges entre 210 y 250 muestras de entrenamiento.

Mientras, en la gráfica en la que se utilizan los coeficientes de detalle, se puede observar que no está bien balanceada, ya que no existe ningún punto en la gráfica donde a partir de un buen número de muestras de entrenamiento consigamos una tasa de acierto alta.

11.1.1 Caso 1: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

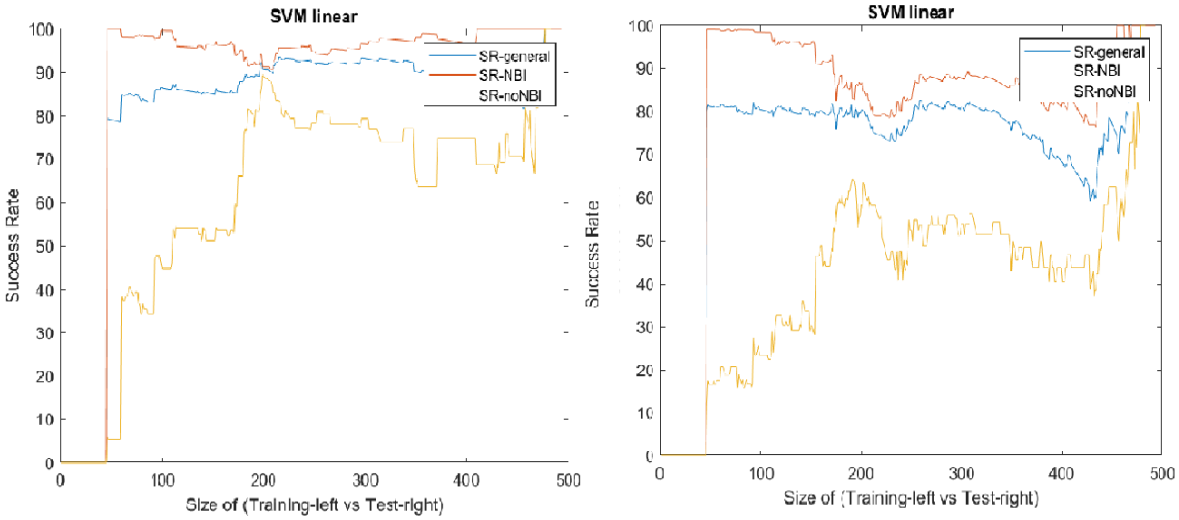


Fig. 48. Estudio SVM Linear sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.1.2 Caso 1: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

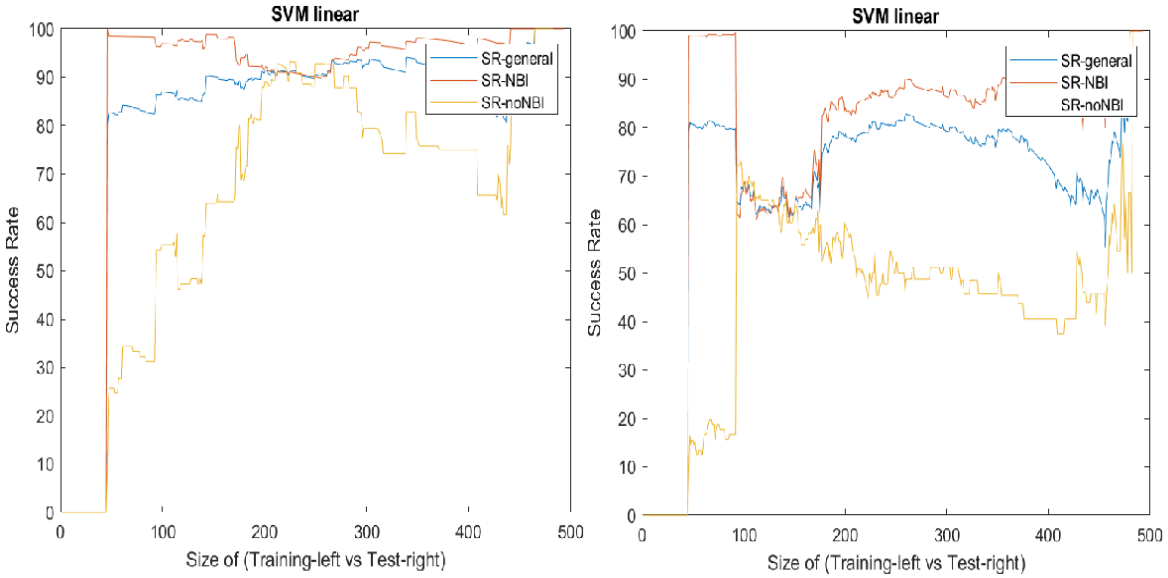


Fig. 49. Estudio SVM Linear sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.1.3 Caso 1: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

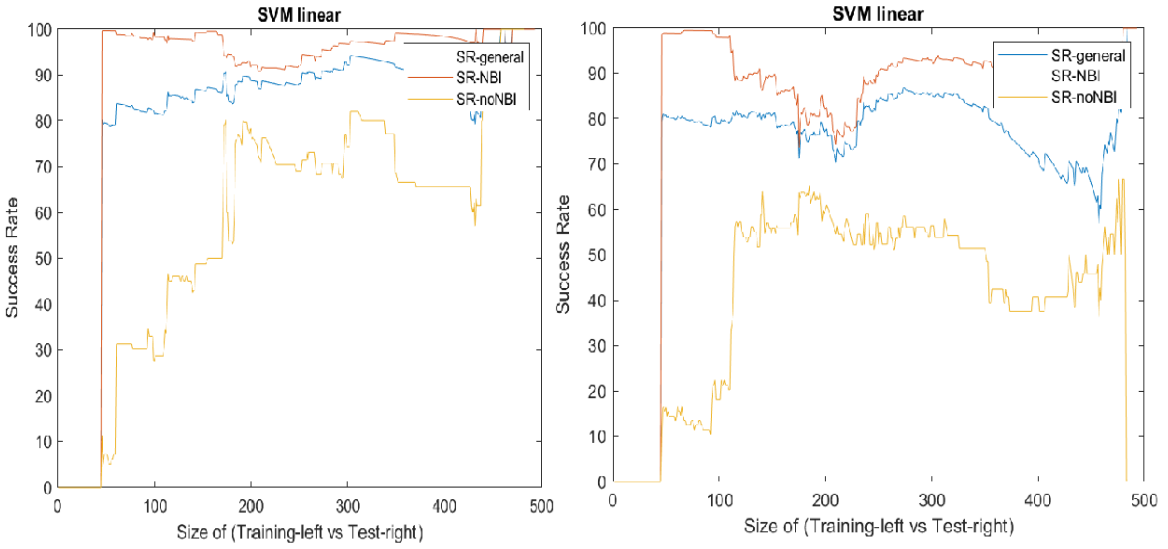


Fig. 50.Estudio SVM Linear sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.1.4 Caso 1: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

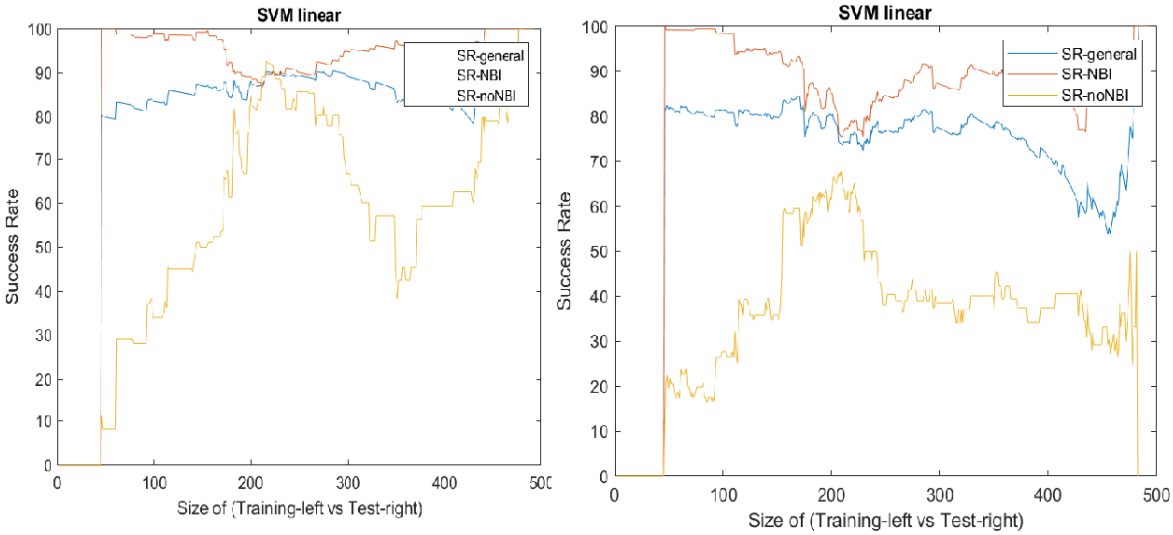


Fig. 51. Estudio SVM Linear sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.1.5 Caso 1: Sin variable "Halpha"

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

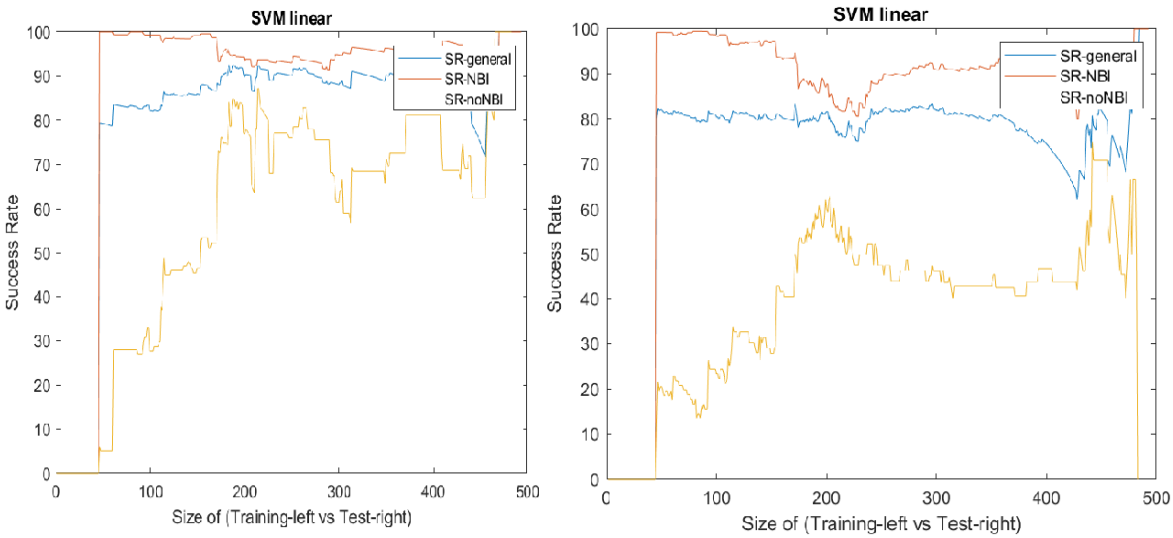


Fig. 52. Estudio SVM Linear sin "Halpha". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, de todos estos casos que se han analizado, donde se iba eliminando cada variable para ver si mejoraban los resultados, se puede comentar que para el estudio utilizando los coeficientes de aproximación los resultados siguen siendo relativamente parecidos al general que se muestra en el caso 1, y, en el caso del estudio utilizando los coeficientes de detalle, el único a destacar es el que eliminas la variable densidad, ya que consigues una zona de conversión entre las 100 y las 200 muestras de entrenamiento, en la cual consigues una tasa de acierto del 65%, la cual es insuficiente.

11.2 Caso 2. Aplicación del clasificador SVM Polynomial

En este segundo apartado se ha realizado el estudio con la ayuda del programa "SVM Polynomial".

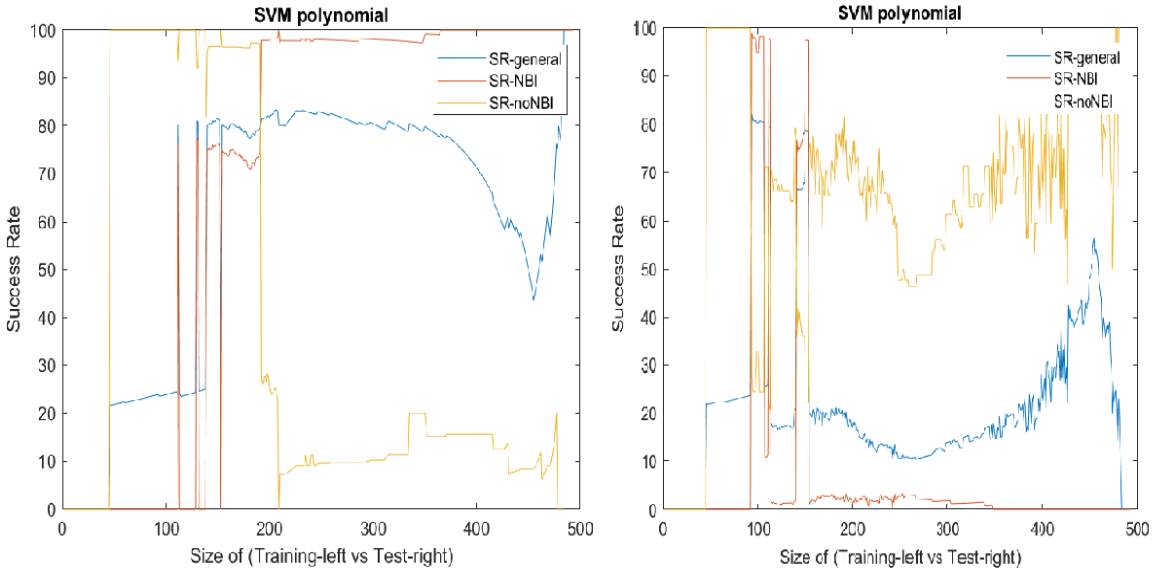


Fig. 53. Estudio SVM Polynomial. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se ha mencionado en el anterior caso, en ambas gráficas se comenzará a analizar por la descarga 46. Del resto de las gráficas podemos resaltar que tanto los coeficientes de aproximación como los de detalle no son suficientes para poder sacar conclusiones positivas, como se puede observar cuando tienes 150 muestras de entrenamiento, que consigues una tasa de acierto en torno al 80%, lo cual está realmente bien, pero, al ser tan irregular y solo ocurre ahí, no se puede tomar como un buen resultado, ya que las gráficas que se consiguen están fuertemente desbalanceadas tomando cualquier otro número de muestras de entrenamiento. En ciertos puntos clasifica verdaderamente bien las de un tipo y en otros puntos las del otro tipo, lo que nos hace concluir que este tipo de programa no es el más adecuado para clasificar datos.

11.2.1 Caso 2: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

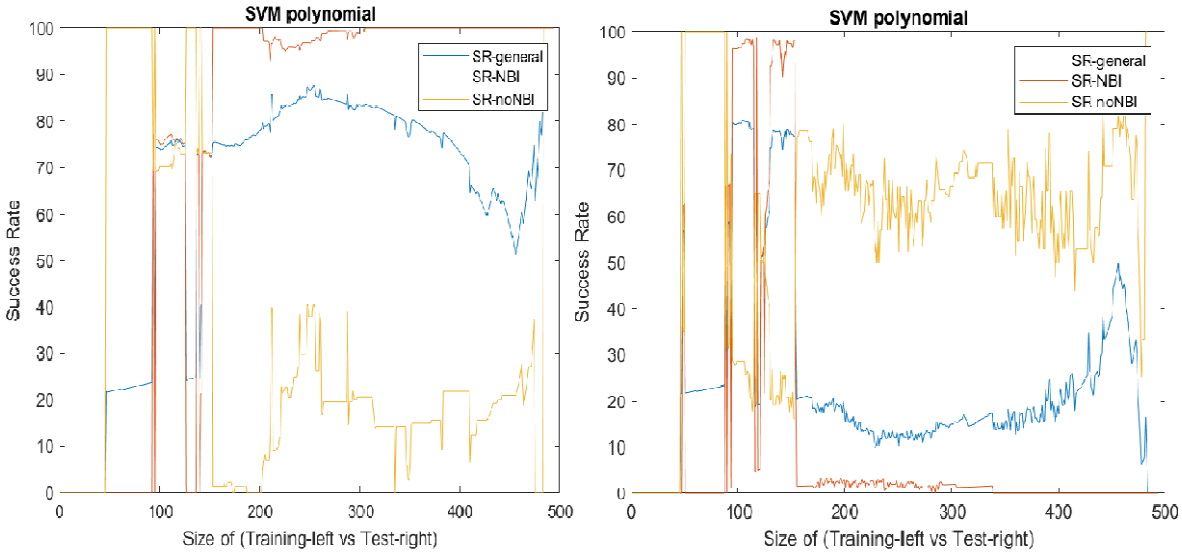


Fig. 54. Estudio SVM Polynomial sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.2.2 Caso 2: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

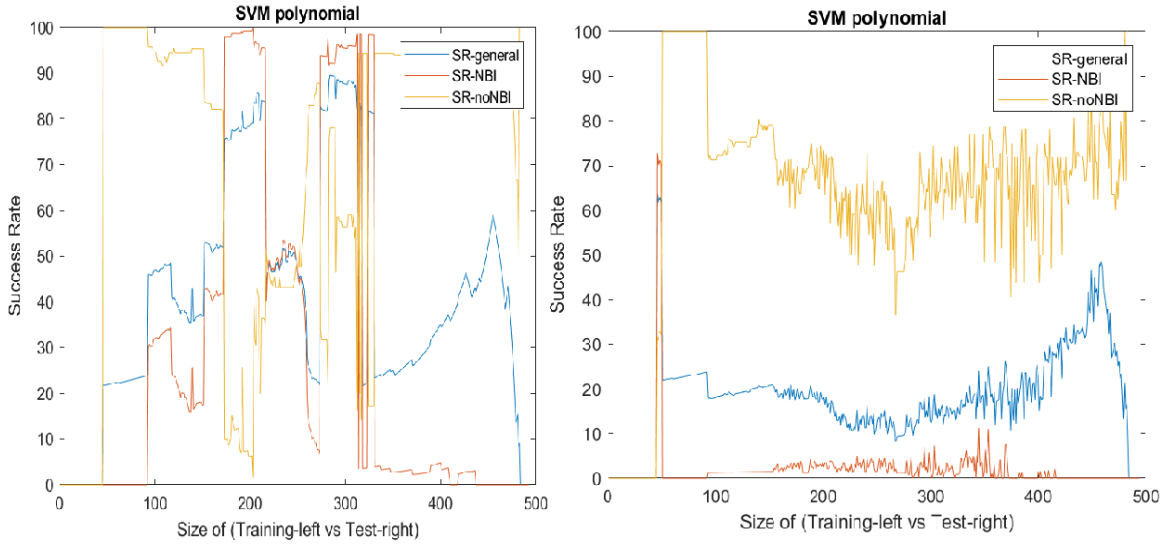


Fig. 55. Estudio SVM Polynomial sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.2.3 Caso 2: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

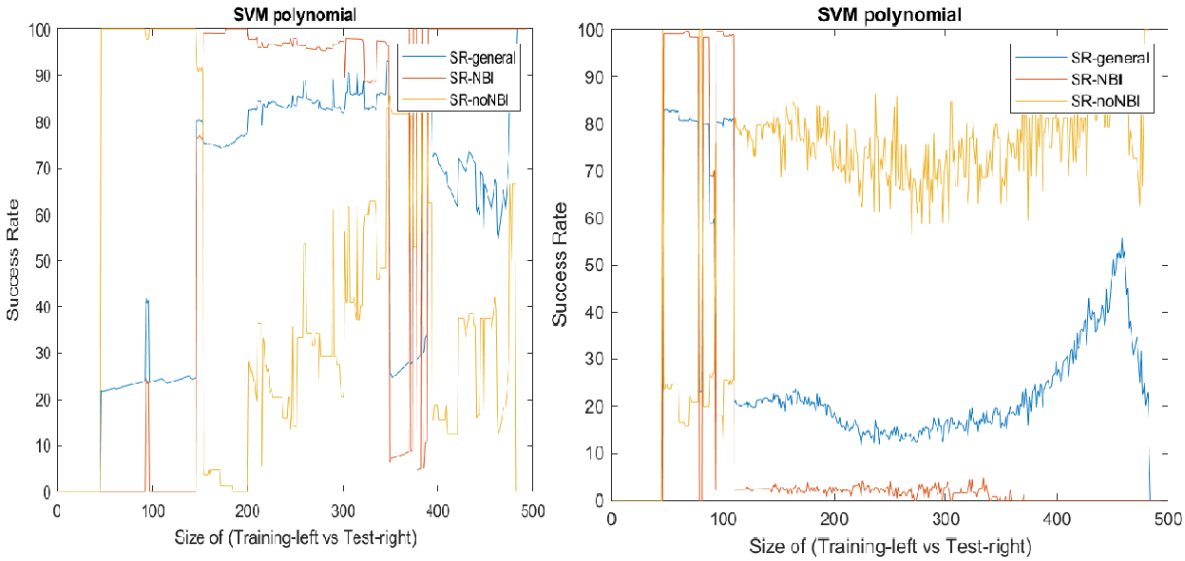


Fig. 56. Estudio SVM Polynomial sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.2.4 Caso 2: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

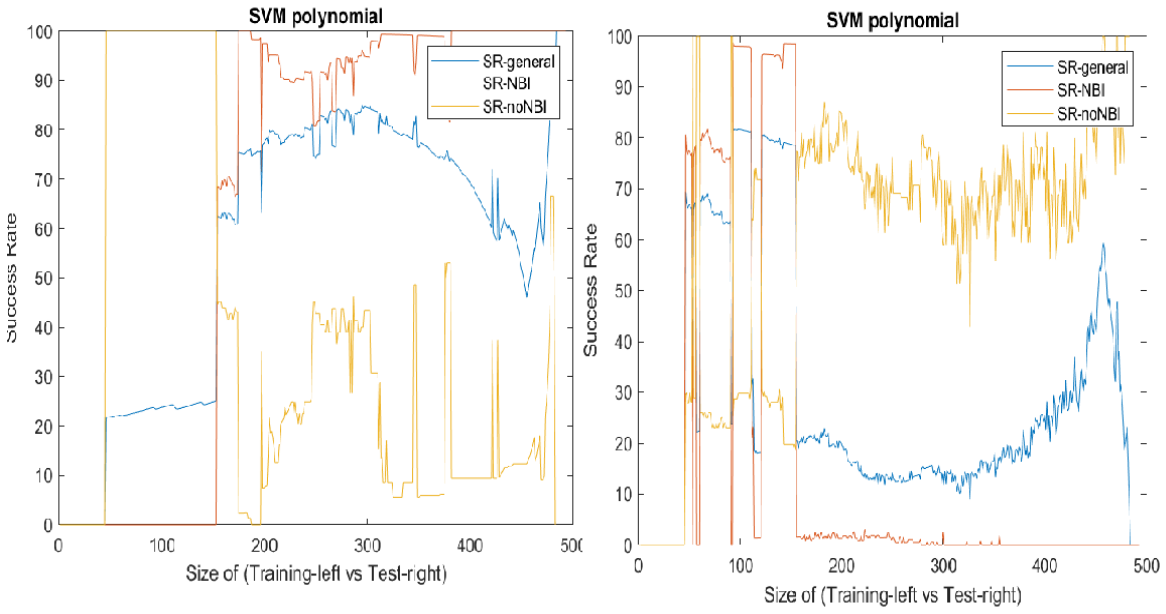


Fig. 57. Estudio SVM Polynomial sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.2.5 Caso 2: Sin variable "Halpha"

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

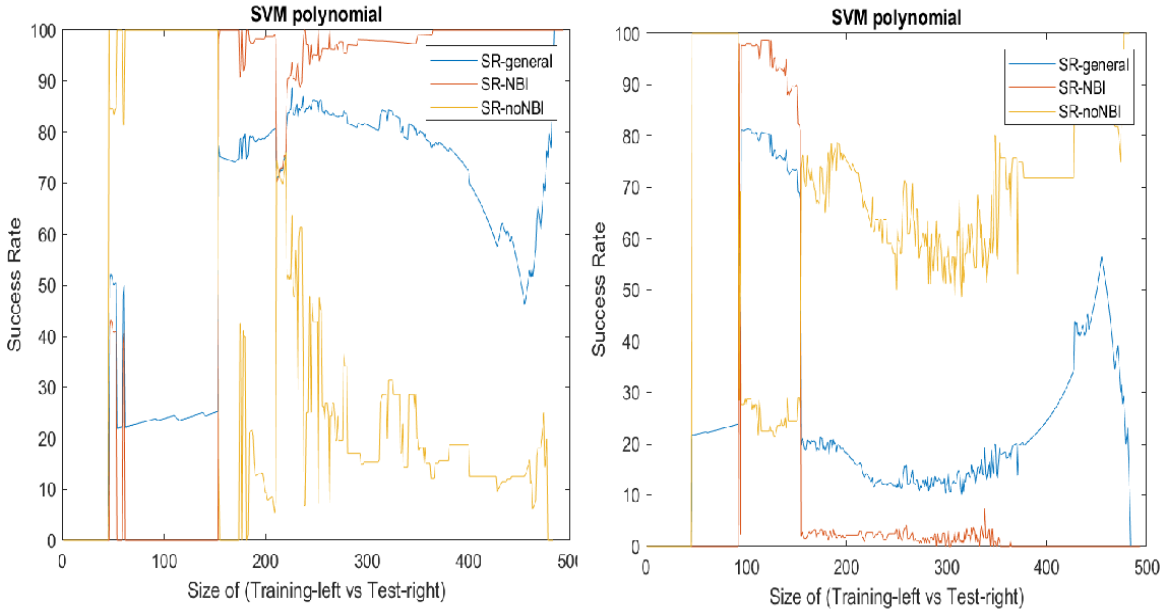


Fig. 58. Estudio SVM Polynomial sin "Halpha". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, de todos estos casos que se han analizado, donde se iba eliminando cada variable para ver si mejoraban los resultados, comentar que en general ocurre lo mismo que en el caso 2, hay zonas puntuales con cierto número de muestras de entrenamiento donde los resultados son realmente buenos, pero son muy irregulares en el resto de los puntos, lo cual no es lo que no interesa a modo de estudio para poder clasificar datos. Se han de puntualizar dos excepciones, que, aunque les ocurra lo que se acaba de comentar, es digno de resaltar, la primera, es en la gráfica de los coeficientes de aproximación quitando la variable de la temperatura, donde alrededor de las 100-120 muestras de entrenamiento, se consigue una tasa de acierto en torno al 70%, y, la segunda, también en el estudio utilizando los coeficientes de aproximación pero, en este caso, sin la variable de energía diamagnética, en la que con 350 muestras se consigue una tasa de acierto altísima del 92%.

11.3 Caso 3. Aplicación del clasificador SVM RBF

En este tercer apartado se ha realizado el estudio con la ayuda del programa "SVM RBF".

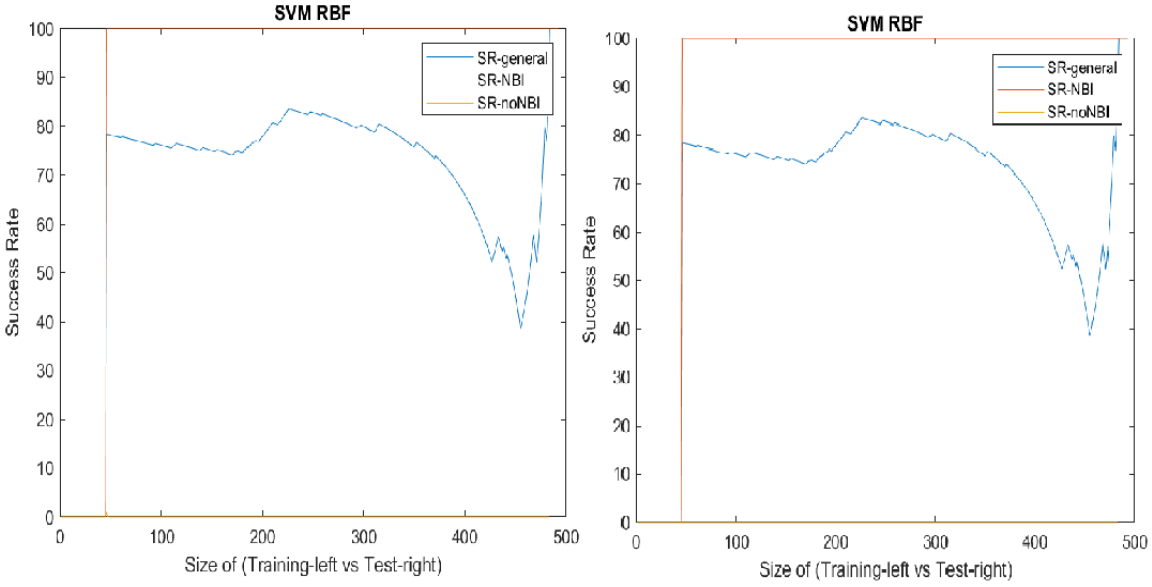


Fig. 59. Estudio SVM RBF. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se ha mencionado en los casos anteriores, en ambas gráficas se comenzará a analizar por la descarga número 46. Del resto de las gráficas, tanto de la del estudio de los coeficientes de aproximación como la de detalle, ambas son claramente insuficientes e incapaces de clasificar de forma correcta los datos. Se puede observar claramente que solo es capaz de clasificar datos de un solo tipo, lo cual nos resulta inservible a la hora de sacar conclusiones claras y objetivas. Por tanto, este tipo de programa de es de utilidad para el estudio en el que nos encontramos.

11.3.1 Caso 3: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

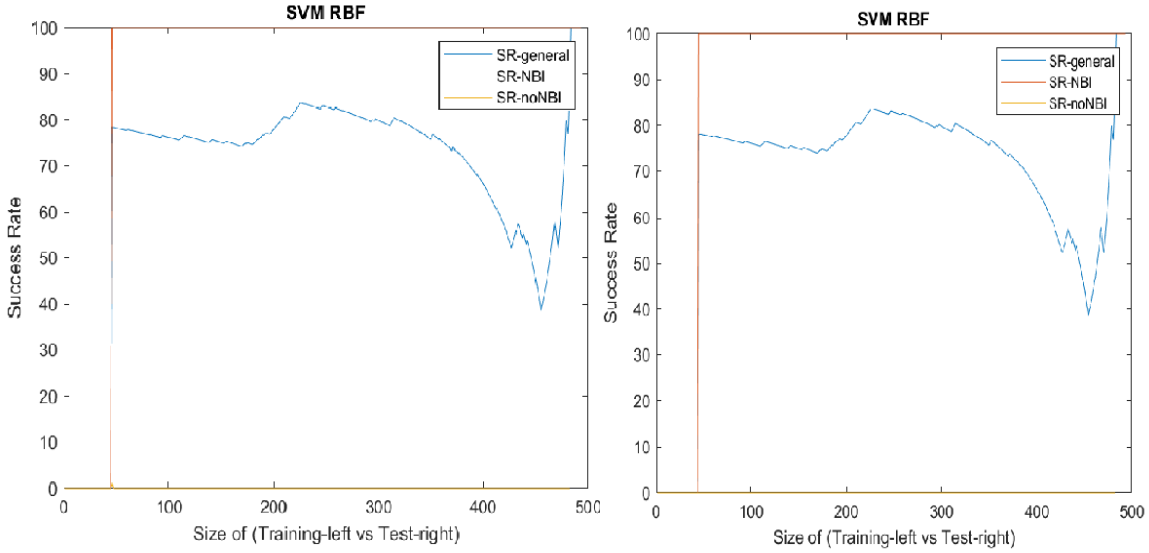


Fig. 60. Estudio SVM RBF sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.3.2 Caso 3: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

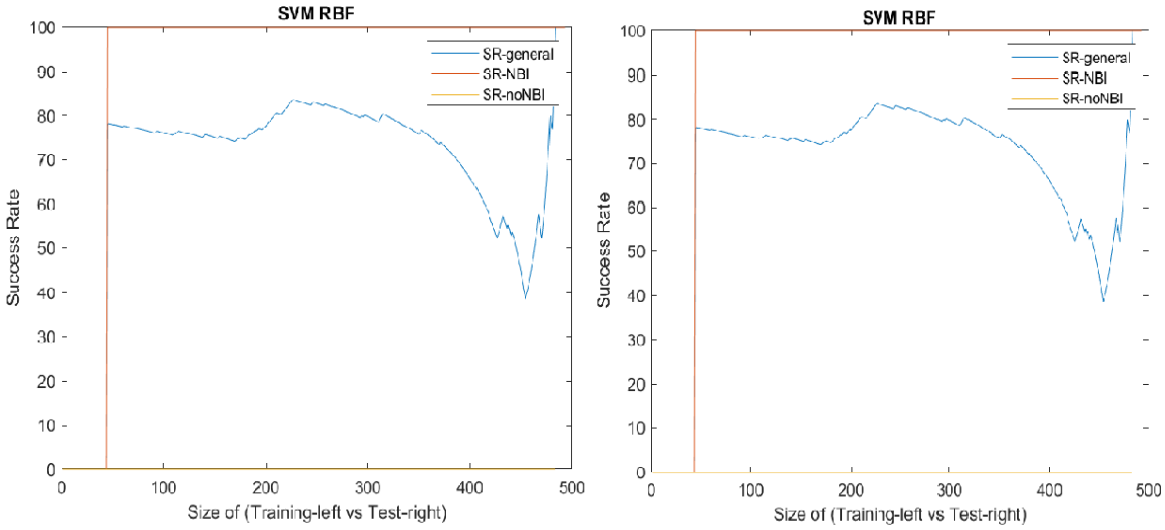


Fig. 61. Estudio SVM RBF sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.3.3 Caso 3: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

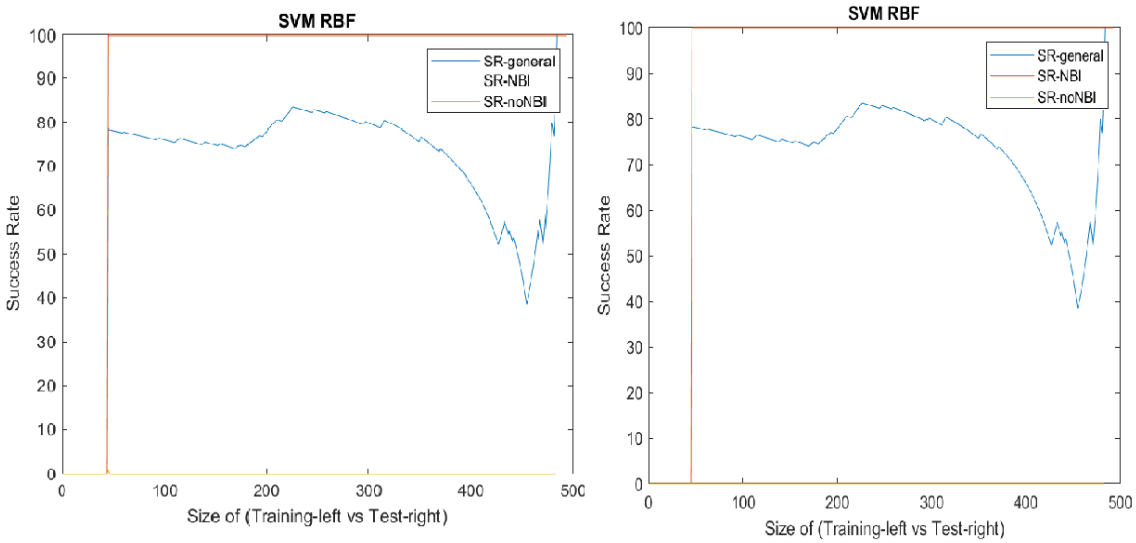


Fig. 62. Estudio SVM RBF sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.3.4 Caso 3: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

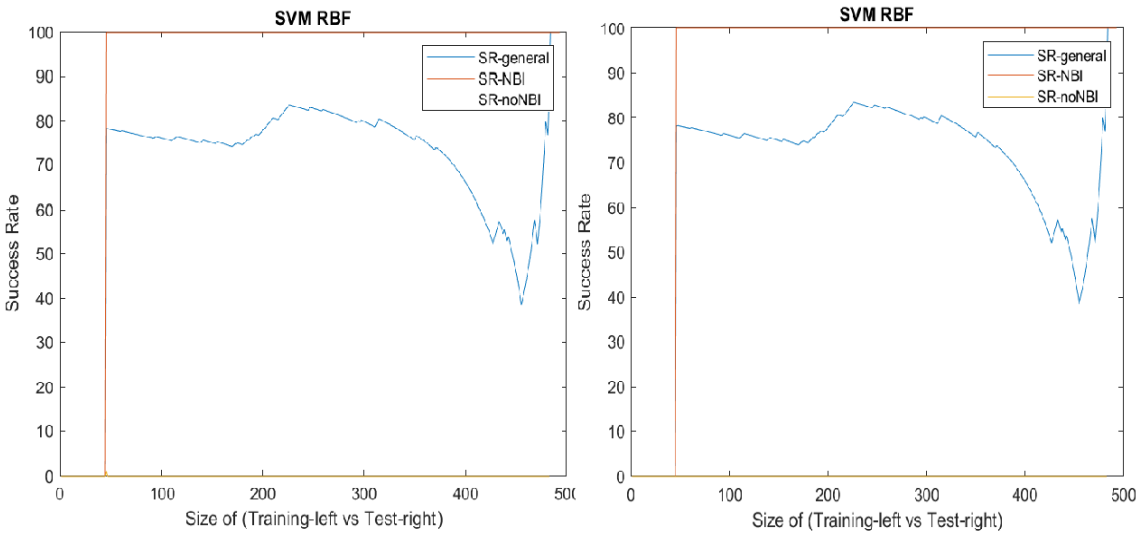


Fig. 63. Estudio SVM RBF sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.3.5 Caso 3: Sin variable "Halpha"

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

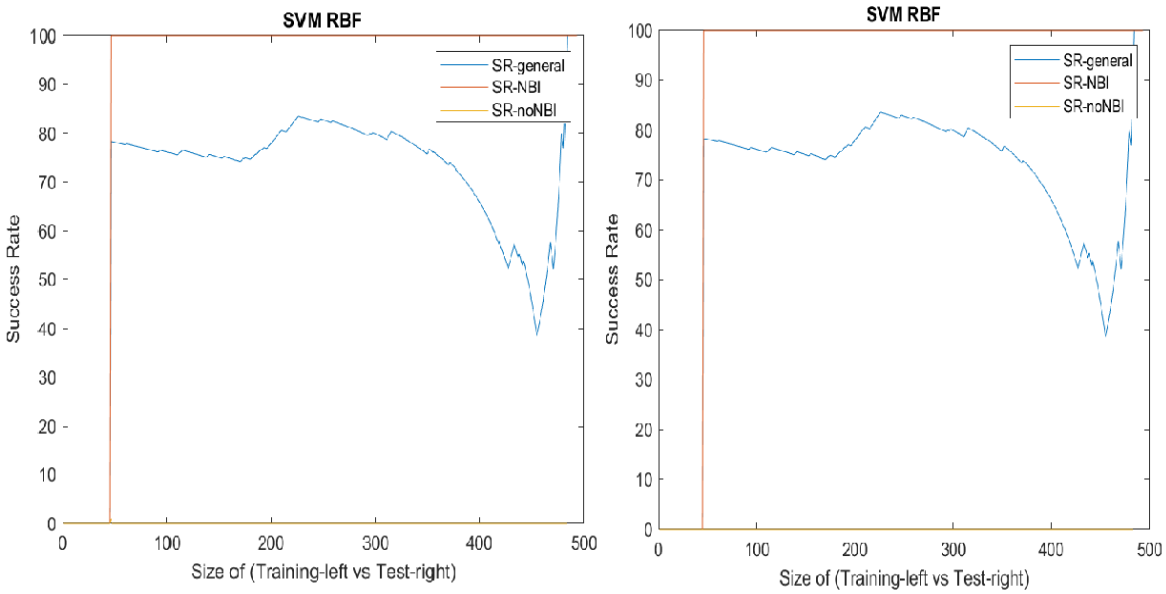


Fig. 64. Estudio SVM RBF sin "Halpha". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, en todos estos casos que se han analizado, donde se iba eliminando cada variable para ver si mejoraban los resultados, se puede comentar que tanto para el estudio de los coeficientes de aproximación como de los de detalle, los resultados siguen siendo los mismo que para el caso 3, sin ninguna excepción. Lo que nos confirma que este tipo de programa no es aconsejable de utilizar en nuestro caso ya que no se clasificarían de forma correcta los datos.

11.4 Caso 4. Aplicación del clasificador KNN2

En este cuarto apartado se ha realizado el estudio con la ayuda del programa "KNN2".

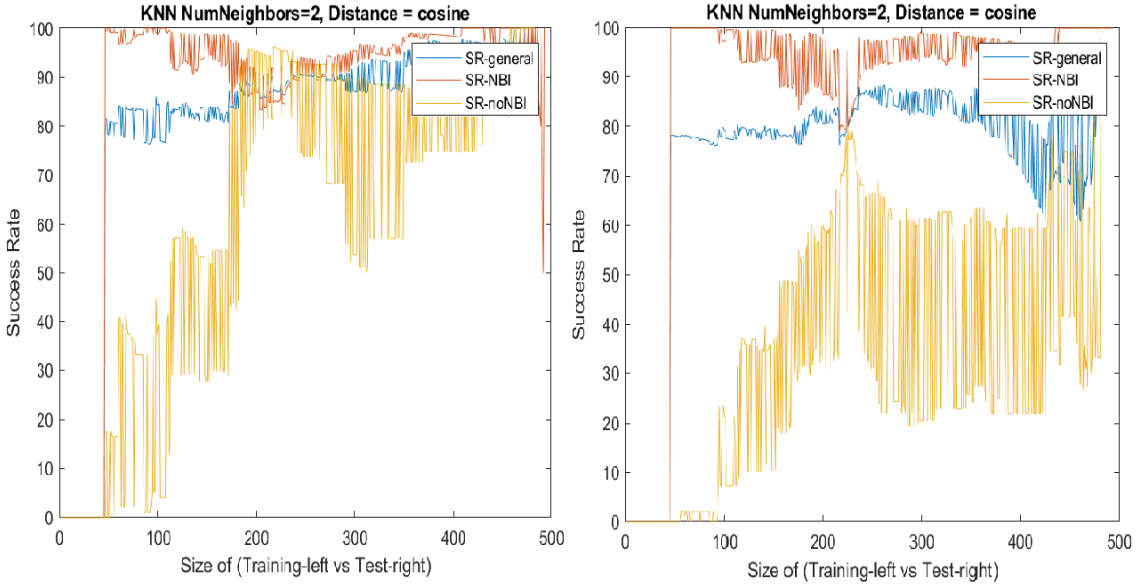


Fig. 65. Estudio KNN2. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se ha mencionado en los anteriores casos, en ambas gráficas se comenzará a analizar por la descarga 46. Del resto de las gráficas podemos resaltar que, en ambos casos, tanto en el estudio de los coeficientes de aproximación como de los de detalle se consiguen resultados más que esperanzadores. Sobre todo, en el caso de los coeficientes de aproximación, donde entre las 200 y 300 muestras de entrenamiento conseguimos una tasa de acierto en torno al 90%, la cual es muy alta, y, aunque a partir de las 300 muestras esa tasa de acierto disminuya, se sigue considerando un muy buen resultado de cara a clasificar correctamente los datos de nuestro estudio. Por otra parte, en la gráfica de detalle solo se consigue una tasa de acierto aceptable del 80% cuando utilizamos 220 muestras de entrenamiento, lo cual, es un buen resultado de cara a estudiar un proceso como el actual. Por tanto, el programa clasificador KNN2 es un gran recurso de cara a clasificación de bancos de datos si utilizamos coeficientes de aproximación.

11.4.1 Caso 4: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

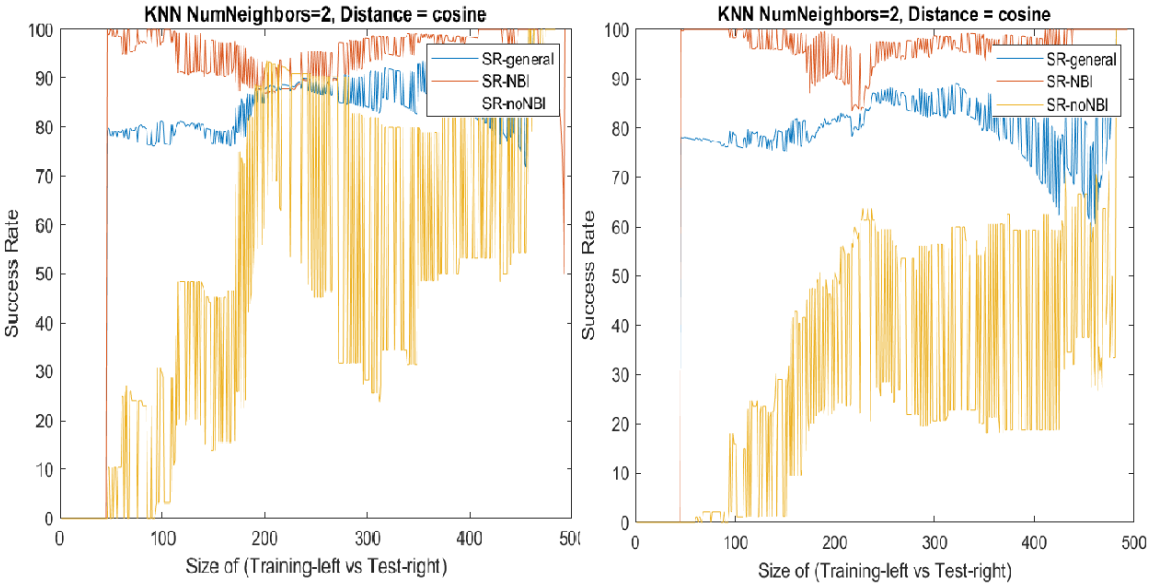


Fig. 66. Estudio KNN2 sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.4.2 Caso 4: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

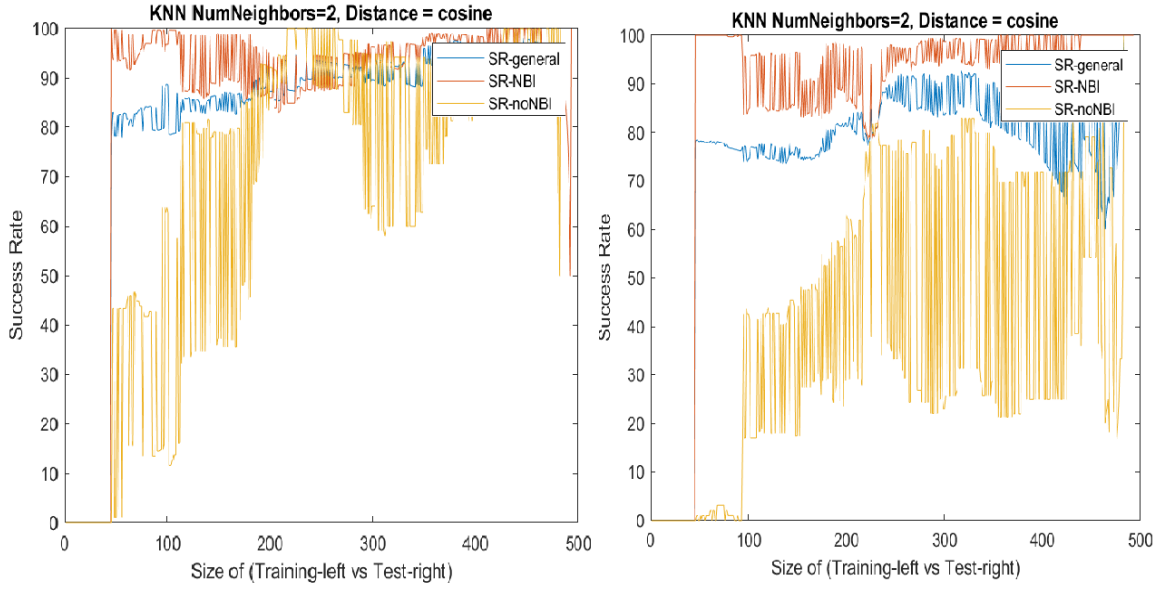


Fig. 67. Estudio KNN2 sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.4.3 Caso 4: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

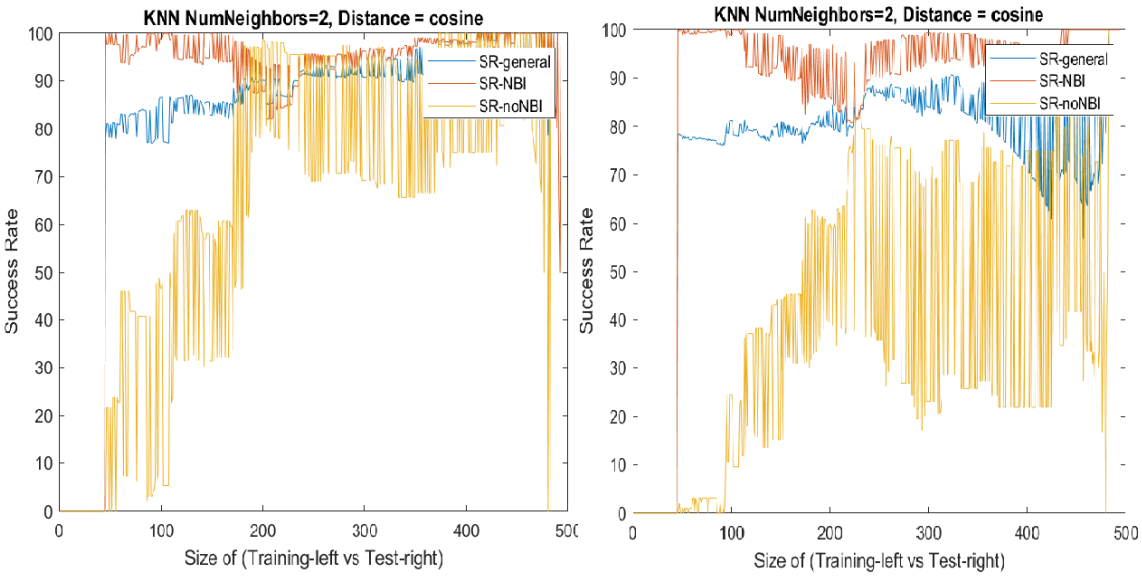


Fig. 68. Estudio KNN2 sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.4.4 Caso 4: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

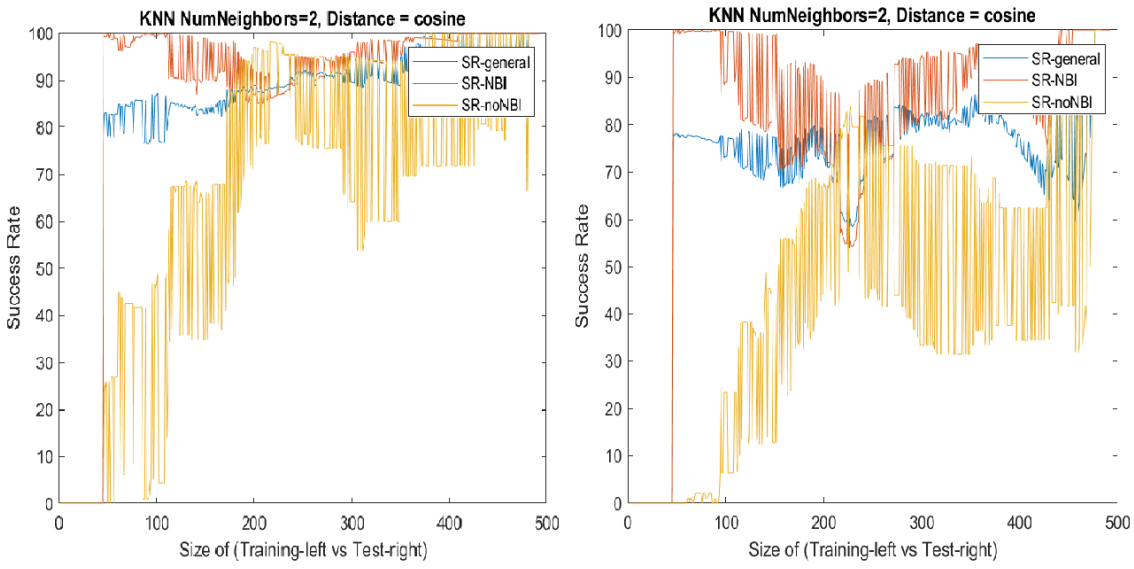


Fig. 69. Estudio KNN2 sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.4.5 Caso 4: Sin variable "Halpha"

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

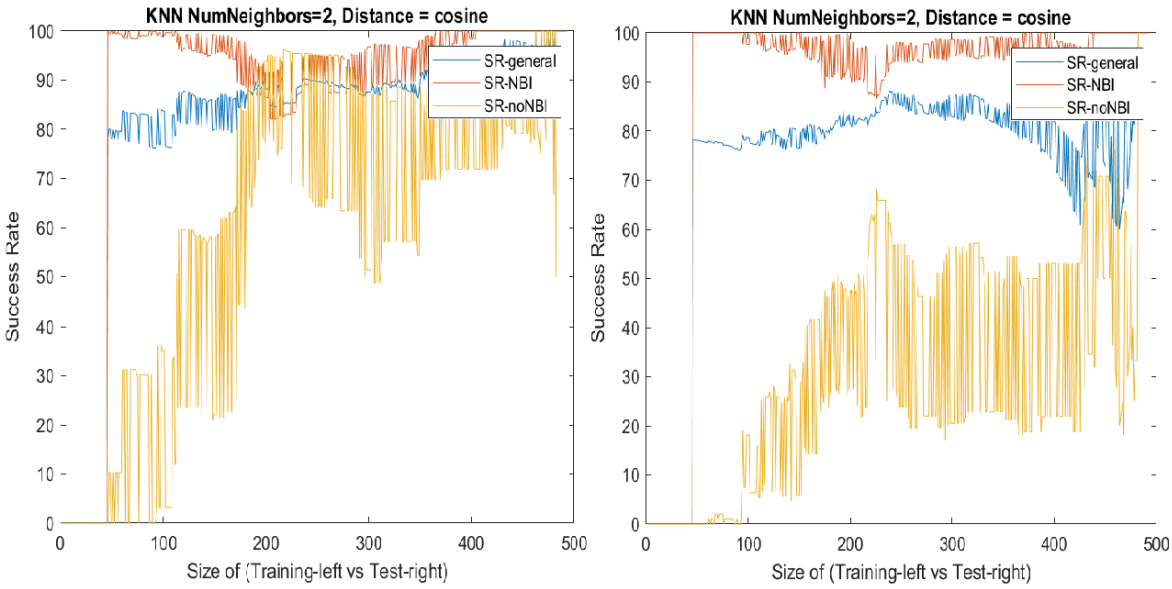


Fig. 70. Estudio KNN2 sin "Halpha". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, de todos estos casos que se han analizado, donde se iba eliminando cada variable para ver si mejoraban los resultados, podemos resaltar lo siguiente, mediante los coeficientes de aproximación, exceptuando cuando quitas la temperatura que obtienes los mismos resultados, cuando eliminas cada una del resto de las variables la gráfica mejora notablemente, hasta el punto que a partir de las 200 muestras de entrenamiento hasta el final, es decir, la 494, se consigue una tasa de acierto altísima, del 90%. Por el otro lado, en el estudio mediante los coeficientes de detalle, a medida que suprimes variables el resultado empeora, incluso obteniendo gráficas en las que ninguna descarga se clasifica correctamente con una tasa de acierto elevada. Por tanto, podemos concluir que el programa KNN2 es de gran utilidad, en el que se obtienen resultados más que sorprendentes a la hora de clasificar datos, y, que, si encima suprimes alguna variable, esos datos mejoran, eso sí, siempre utilizando los coeficientes de aproximación.

11.5 Caso 5. Aplicación del clasificador KNN6

En este quinto apartado se ha realizado el estudio con la ayuda del programa "KNN6".

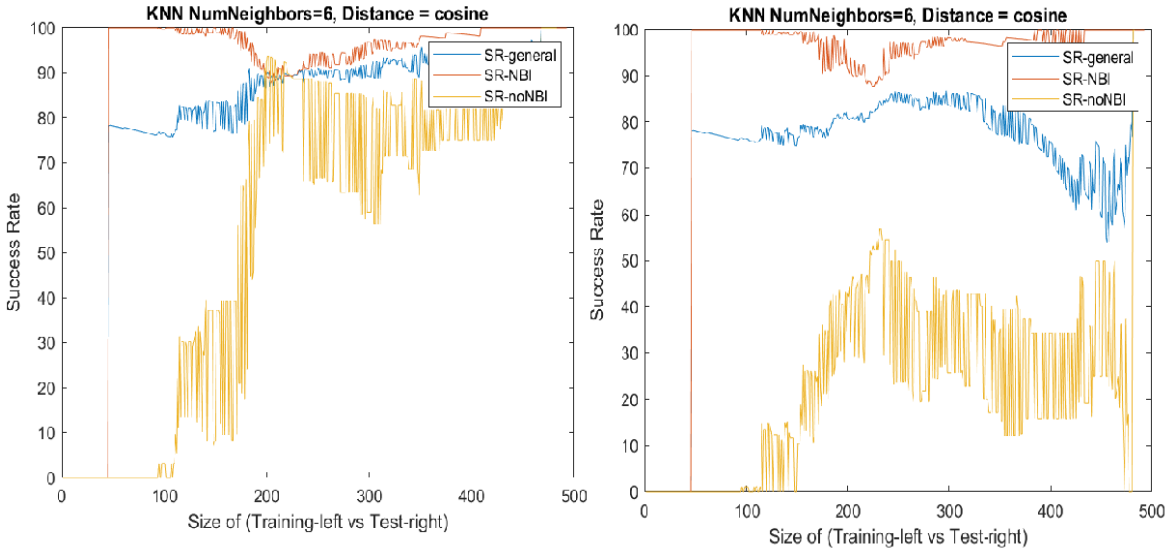


Fig. 71. Estudio KNN6. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se ha mencionado en anteriores casos, en ambas gráficas se comenzará a analizar por la descarga 46. Del resto de las gráficas podemos resaltar que en el estudio utilizando los coeficientes de aproximación se consiguen buenos resultados, obteniendo una tasa de acierto del 90% cuando se utilizan entre 200 y 220 muestras de entrenamiento. Por el contrario, cuando utilizamos los coeficientes de detalle no se consigue el resultado esperado. Por tanto, el uso de este programa se considera óptimo de cara a la clasificación de datos.

11.5.1 Caso 5: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

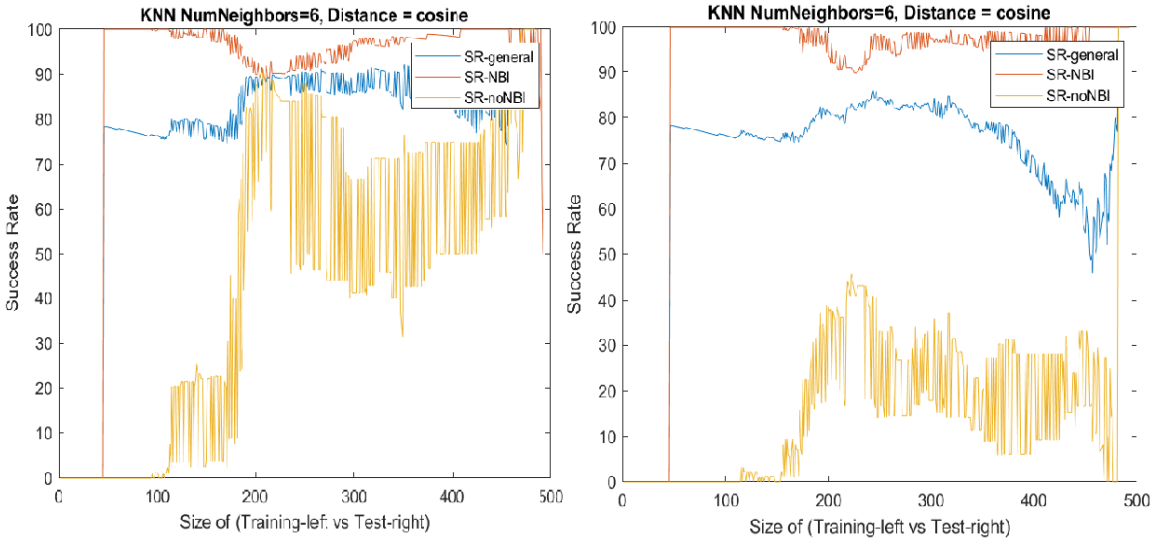


Fig. 72. Estudio KNN6 sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.5.2 Caso 5: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

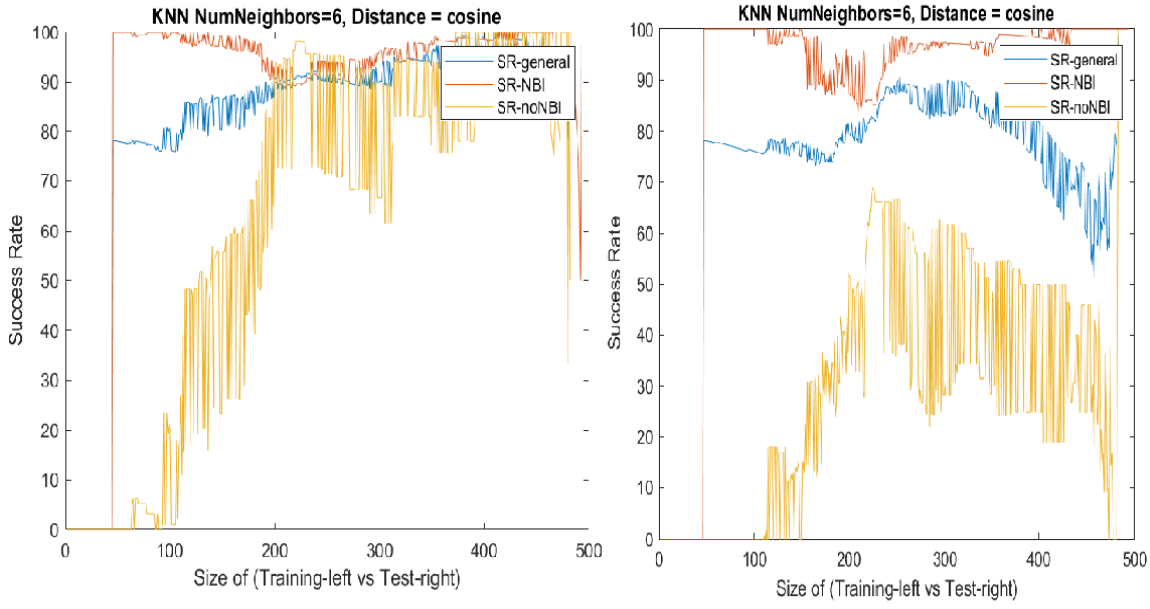


Fig. 73. Estudio KNN6 sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.5.3 Caso 5: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

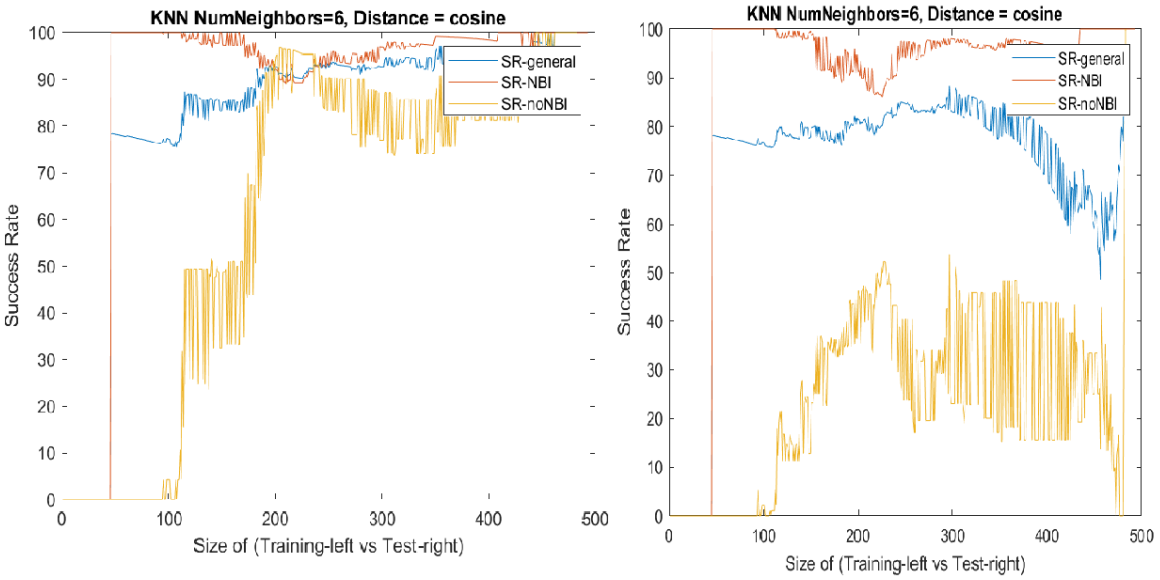


Fig. 74. Estudio KNN6 sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.5.4 Caso 5: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

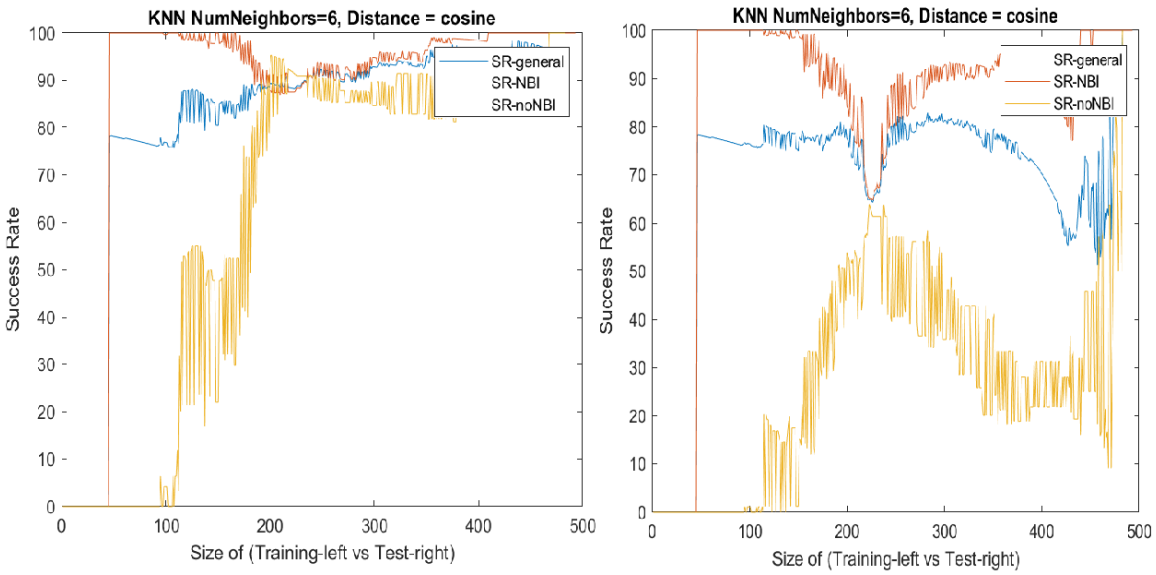


Fig. 75. Estudio KNN6 sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.5.5 Caso 5: Sin variable “Halpha”

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

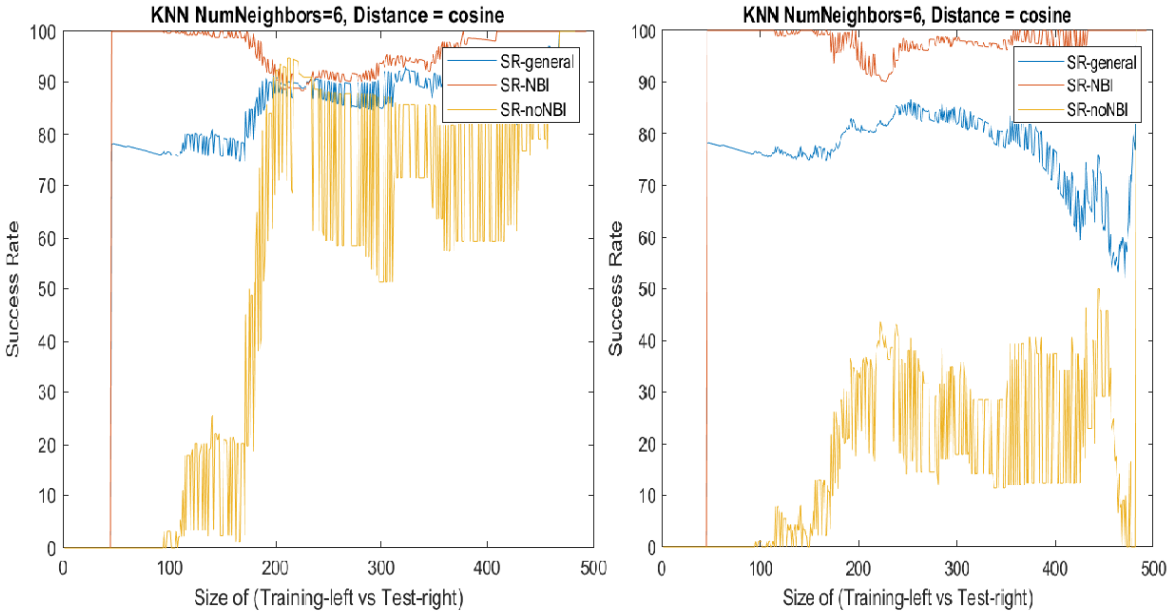


Fig. 76. Estudio KNN6 sin “Halpha”. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, de todos estos casos que se han analizado, donde se iba eliminando cada variable para ver si mejoraban los resultados, podemos resaltar lo siguiente, únicamente mediante el estudio de los coeficientes de aproximación se consiguen resultados interesantes y válidos a la hora de evaluar y clasificar los datos. Resaltar que cuando se elimina la variable densidad se obtiene una tasa de acierto muy alta, del 90%, desde que tomas 200 muestras de entrenamiento hasta el final, es decir, las 494 muestras. En el resto de los casos se obtienen los mismos resultados que en el caso 5 general, es decir, unos buenos resultados para un programa cuya finalidad es la clasificación de datos sin la pérdida esencial de información.

11.6 Caso 6. Aplicación del clasificador Discriminant Linear

En este sexto apartado se ha realizado el estudio con la ayuda del programa "Discriminant Linear".

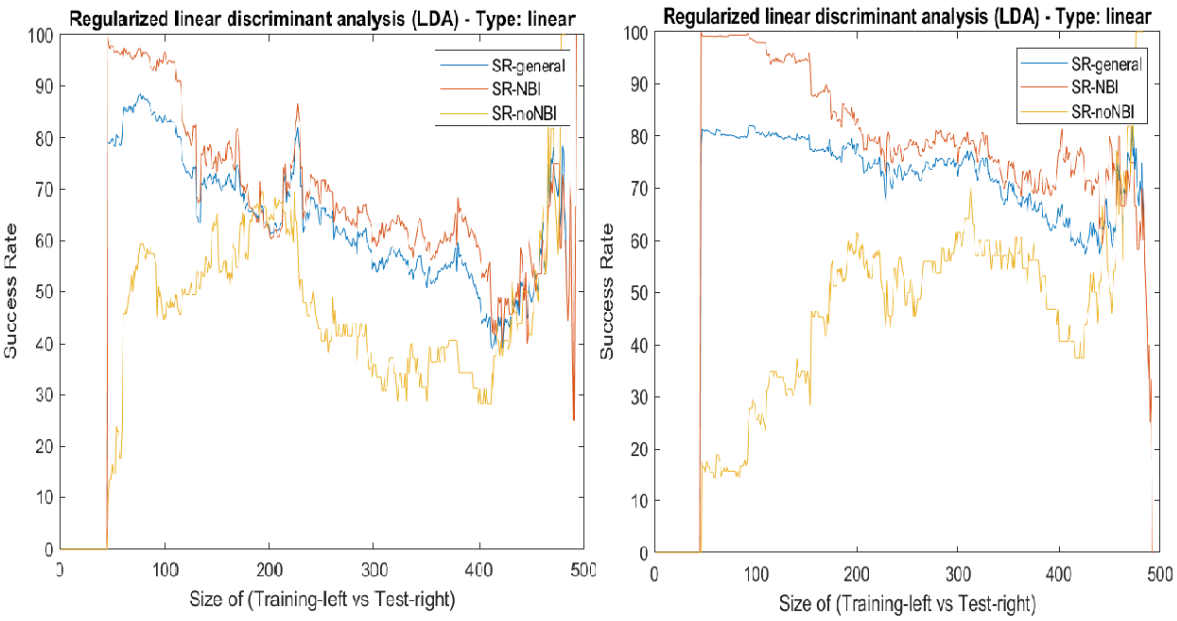


Fig. 77. Estudio Discriminant Linear. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se ha mencionado en el anterior caso, em ambas gráficas se comenzará a analizar por la descarga 46. Del resto de las gráficas podemos resaltar que tanto los coeficientes de aproximación como los de detalle no son suficientes para poder sacar conclusiones positivas. En el caso de los coeficientes de aproximación, se puede observar que, cuando hay 200 muestras de entrenamiento, se consigue una tasa de acierto en torno al 70 %, lo cual se consideraría como un resultado aceptable si no fuese porque el resto de la gráfica es muy irregular, es decir, está fuertemente desbalanceada, por tanto, este tipo de programa no se puede recomendar para su uso para clasificar datos como los que estamos analizando.

11.6.1 Caso 6: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

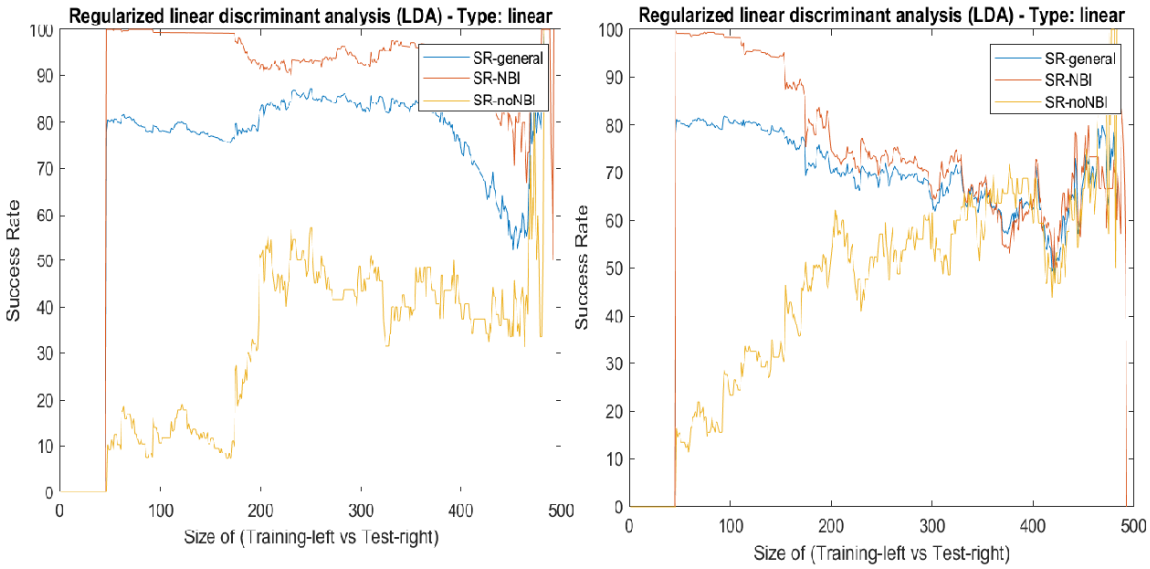


Fig. 78. Estudio Discriminant Linear sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.6.2 Caso 6: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

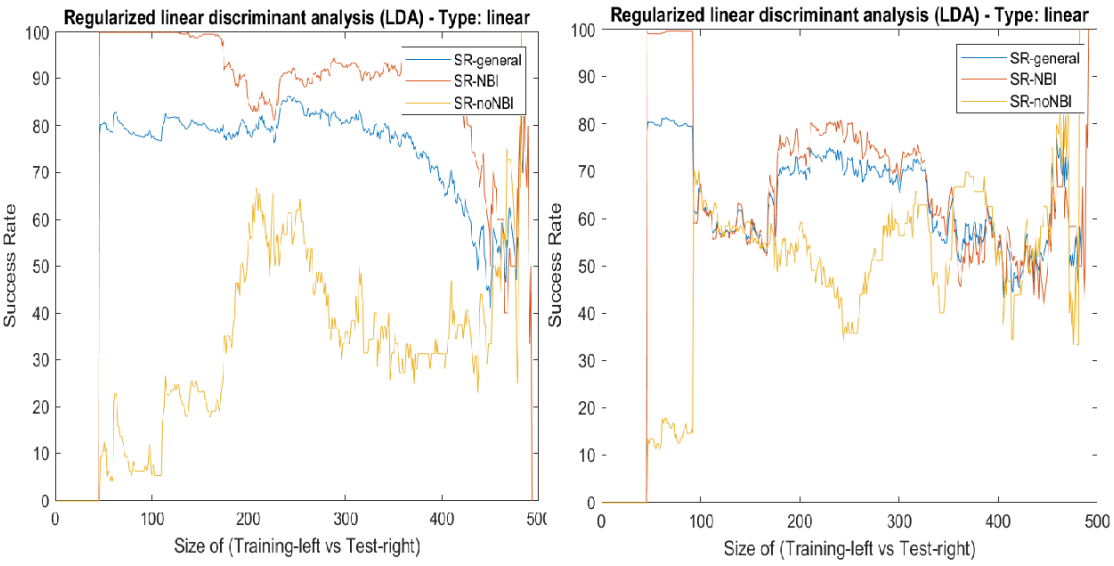


Fig. 79. Estudio Discriminant Linear sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.6.3 Caso 6: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

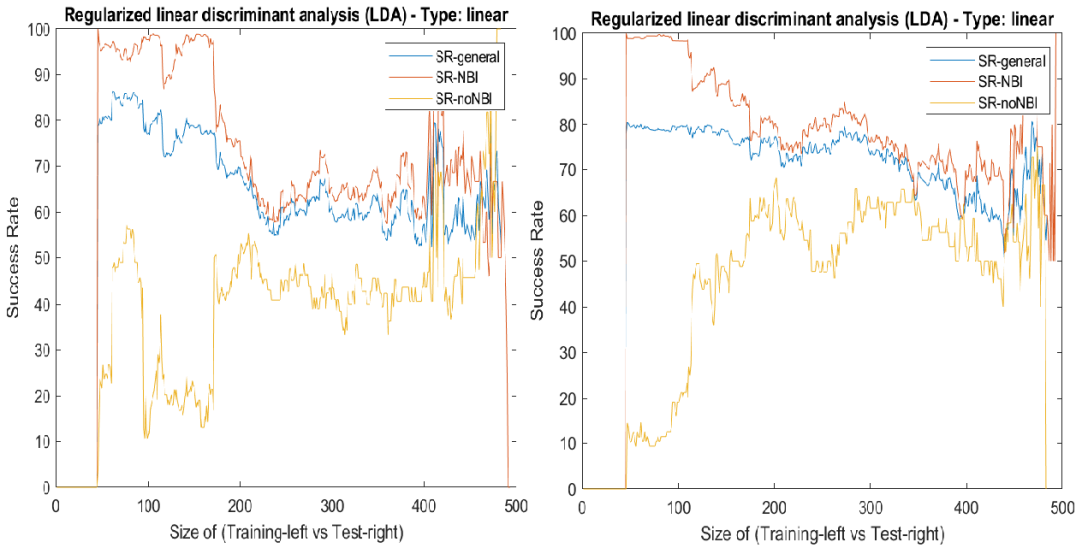


Fig. 80. Estudio Discriminant Linear sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.6.4 Caso 6: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

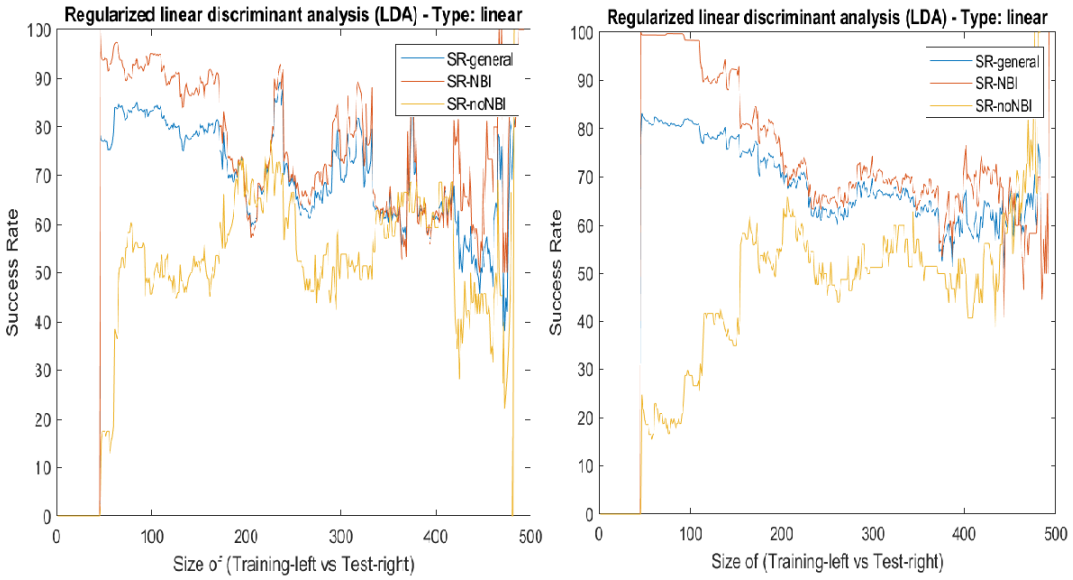


Fig. 81. Estudio Discriminant Linear sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.6.5 Caso 6: Sin variable “Halpha”

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

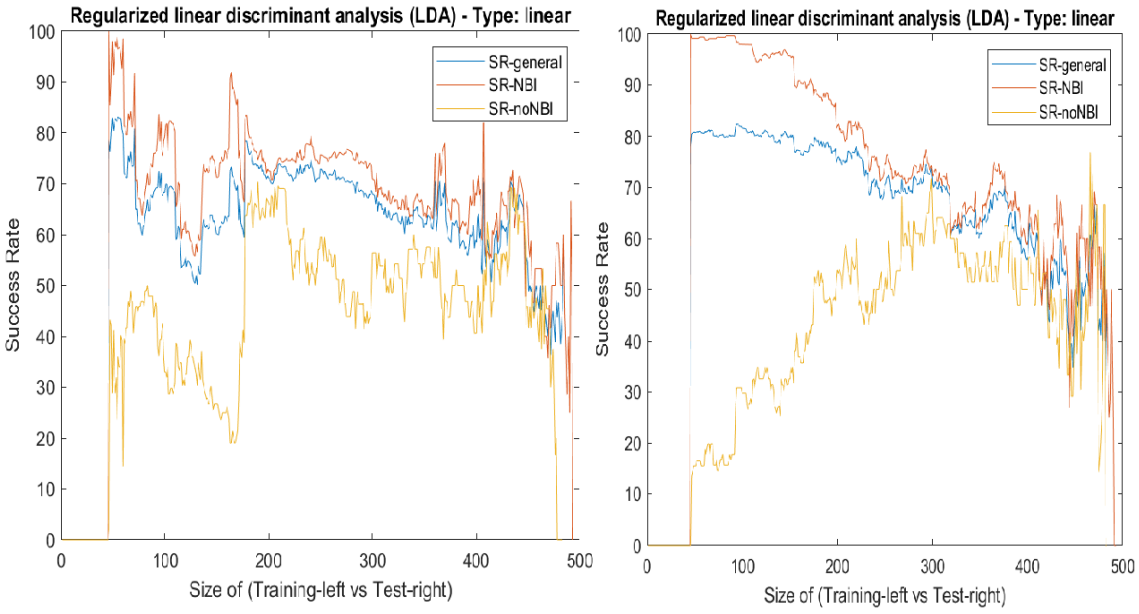


Fig. 82. Estudio Discriminant Linear sin “Halpha”. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, de todos estos casos que se han analizado, donde se iba eliminando cada variable para ver si mejoraban los resultados, se puede comentar que ni el estudio mediante coeficientes de aproximación ni el estudio mediante coeficientes de detalle es suficiente. Es cierto que, cuando se han utilizado los coeficientes de detalle, cuando quitabas cualquier variable, a partir de las 300 muestras de entrenamiento se obtenían una tasa de acierto aceptable pero no suficiente. Y, cuando utilizabas los de aproximación, cuando quitabas la variable de corriente de plasma también se obtienen unos resultados aceptables por tramos. Pero, en general, se ha llegado a la conclusión de que este programa no es el adecuado para clasificar datos, ya que, aparte de fiabilidad, lo que se busca es regularidad, y este tipo de programa no la da.

11.7 Caso 7. Aplicación del clasificador Discriminant Quadratic

En este sexto apartado se ha realizado el estudio con la ayuda del programa "Discriminant Quadratic".

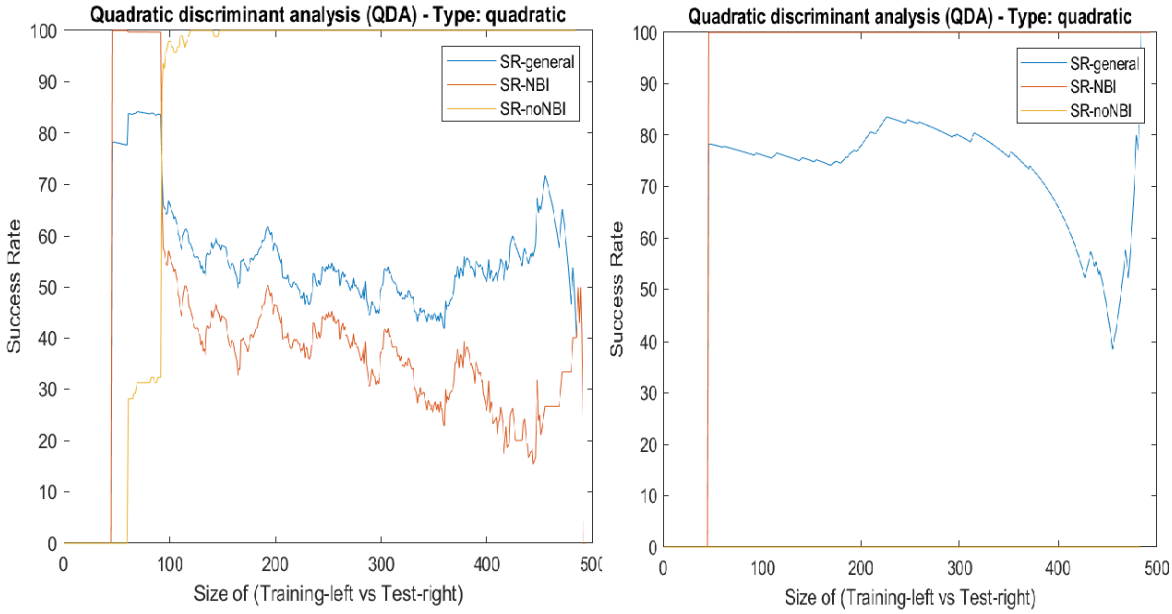


Fig. 83. Estudio Discriminant Quadratic. Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

Como se ha mencionado en el anterior caso, en ambas gráficas se comenzará por la descarga 46. Del resto de las gráficas podemos resaltar que tanto los coeficientes de aproximación como los de detalle no son suficientes para sacar conclusiones positivas. Se ha observado que en ninguno de los casos consigues que, para cierto número de muestras de entrenamiento la tasa de acierto sea al menos aceptable, por tanto, este tipo de programa para nada es válido, no es aconsejable su utilización para casos de clasificación de datos como en el que nos encontramos.

11.7.1 Caso 7: Sin variable "Te"

Se han analizado las descargas sin tener en cuenta la variable de temperatura, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

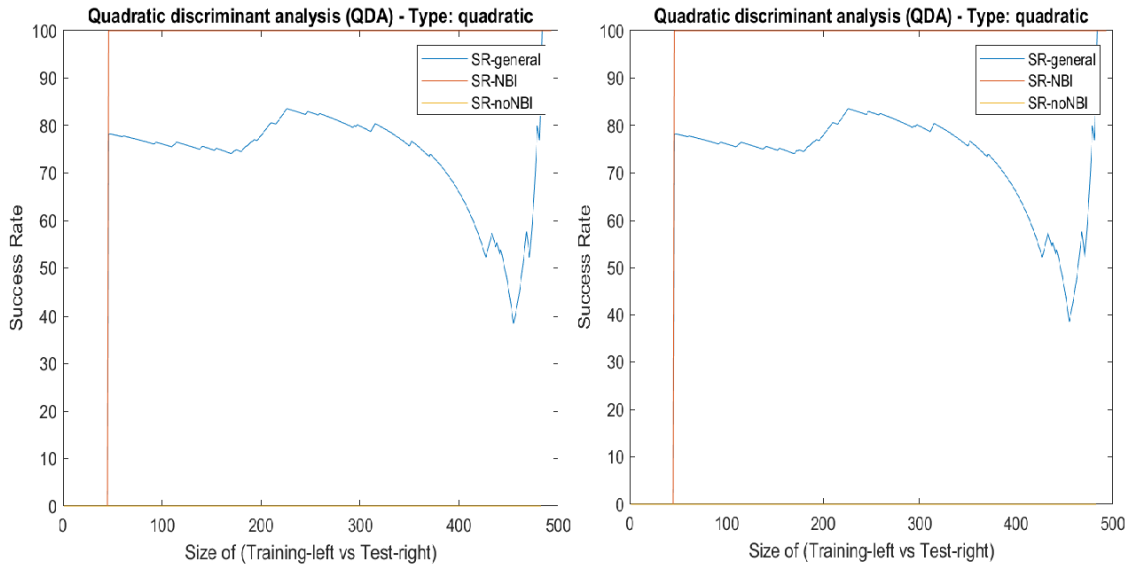


Fig. 84. Estudio Discriminant Quadratic sin "Te". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.7.2 Caso 7: Sin variable "n"

Se han analizado las descargas sin tener en cuenta la variable de densidad, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

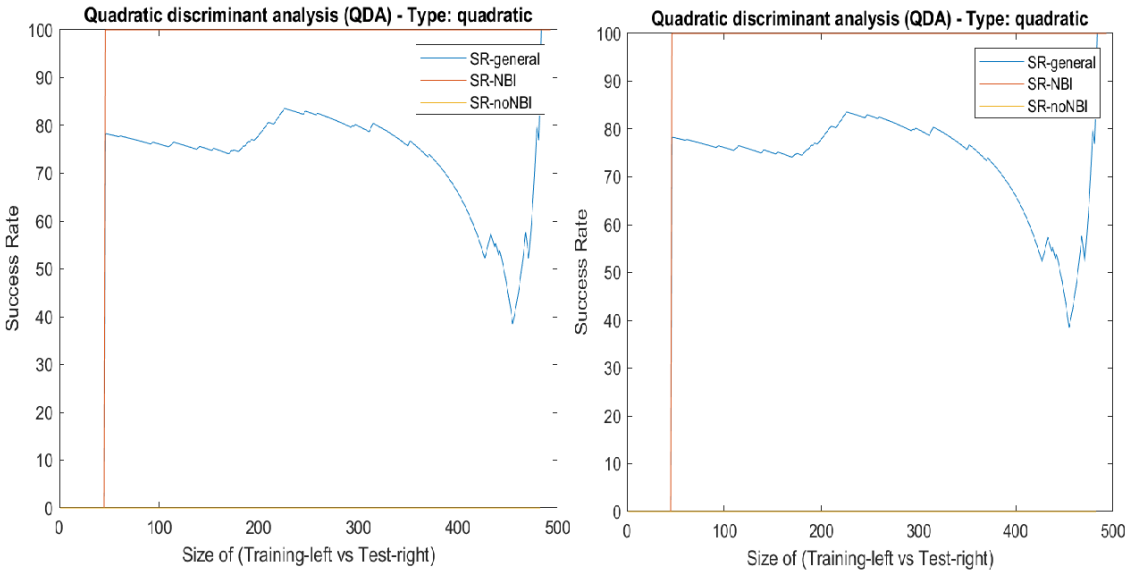


Fig. 85. Estudio Discriminant Quadratic sin "n". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.7.3 Caso 7: Sin variable "Wp"

Se han analizado las descargas sin tener en cuenta la variable de energía diamagnética, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

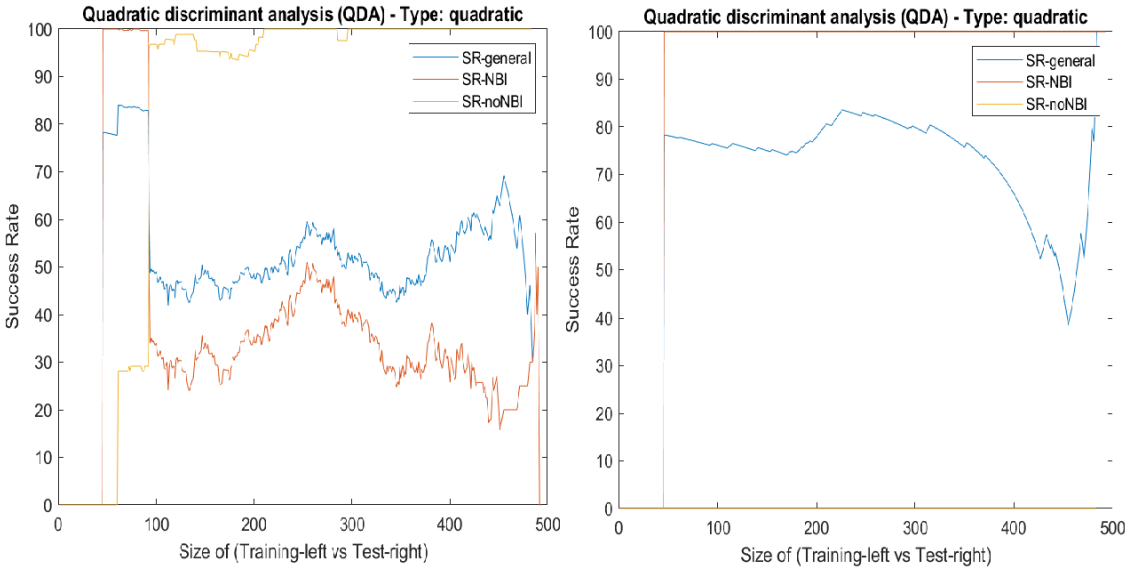


Fig. 86. Estudio Discriminant Quadratic sin "Wp". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.7.4 Caso 7: Sin variable "Ip"

Se han analizado las descargas sin tener en cuenta la variable de corriente de plasma, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

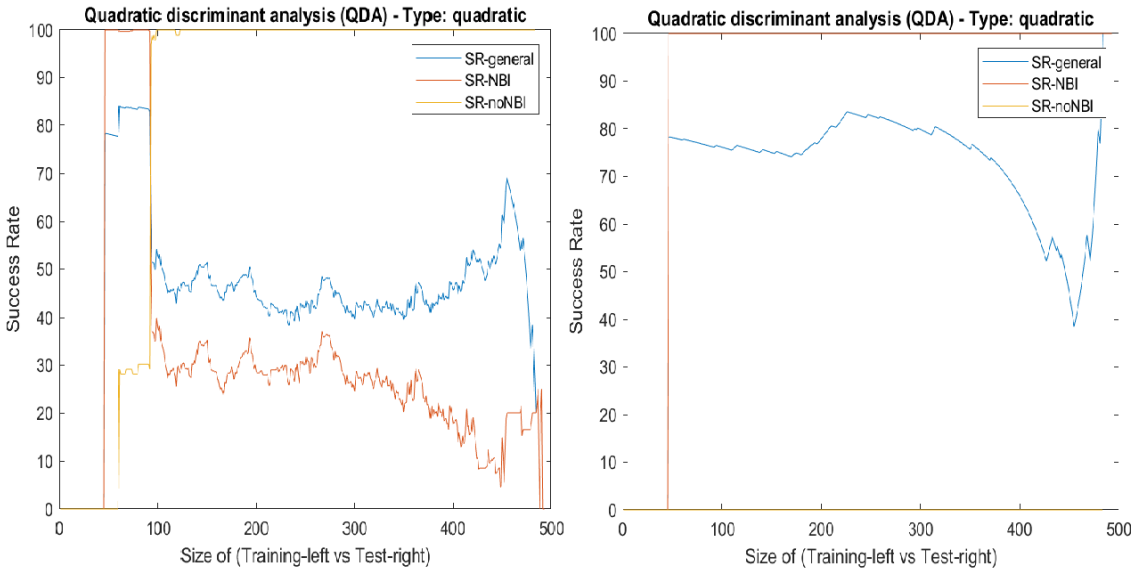


Fig. 87. Estudio Discriminant Quadratic sin "Ip". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

11.7.5 Caso 7: Sin variable "Halpha"

Se han analizado las descargas sin tener en cuenta la variable de Halpha, con la finalidad de poder clarificar y mejorar los resultados obtenidos y, poder, si es posible, determinar una mejor tasa de acierto.

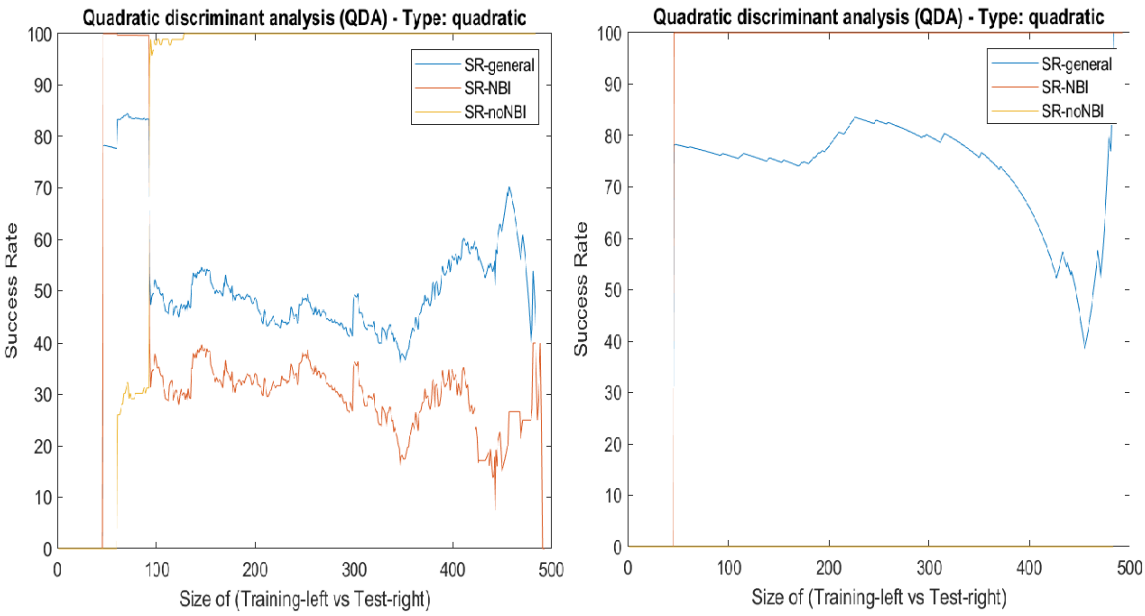


Fig. 88. Estudio Discriminant Quadratic sin "Halpha". Coeficientes de aproximación (Izquierda) / Coeficientes de detalle (Derecha)

A modo de conclusión, en todos estos casos que se han analizado, donde se iban eliminando cada variable para ver si mejoraban los resultados, se puede comentar que tanto para el estudio de los coeficientes de aproximación como de los de detalle, los resultados siguen siendo igual de decepcionantes que para el caso 7, sin ninguna excepción. Lo que nos confirma que este tipo de programa no es aconsejable de utilizar en nuestro caso ya que no se clasificarían de forma correcta los datos.

12 Conclusiones

Los resultados obtenidos en el presente TFG a partir del análisis wavelet-Haar aplicado a señales de evolución temporal del TJ-II arrojan información relevante, significativa y muy práctica.

Se confirma que los coeficientes de aproximación de la transformada wavelet permiten reducir notablemente la información de una señal de evolución temporal mientras se conserva al mismo tiempo la forma de onda estructural de dicha señal. Esta característica permite que se puedan manipular datos de una forma más rápida y eficiente. No obstante, la gran cantidad de información utilizada en el presente trabajo (inicialmente 21 Gb de información bruta perteneciente a un subconjunto de descargas y señales del TJ-II) dificulta igualmente la clasificación de muchas señales de evolución temporal, incluso utilizando una base de datos similar, pero de menor información (3,5 Mb) habiéndole aplicado convenientemente la transformada matemática wavelet-Haar.

Un primer análisis gráfico de los atributos utilizados para todas las descargas de operación, utilizando diagramas de dispersión tanto a los coeficientes de aproximación como a los coeficientes de detalle, muestra y arroja la dificultad de discernir nítidamente los diferentes grupos de clasificación en base al calentamiento del plasma utilizado. Esta primera conclusión fruto del análisis gráfico, se confirmó al emplear toda la base de datos (todas las descargas y todas las señales a la vez) en la agrupación binaria (solamente dos grupos de clasificación) del tipo de calentamiento del TJ-II y haciendo uso de diferentes algoritmos de clasificación automática.

Además, se comprobó que, utilizando una combinación o selección inferior de características, se obtenían igualmente diferentes resultados, incluso en ocasiones con mejores tasas de aciertos que si se utilizaban todas las características disponibles de una sola vez. Esto sugiere que una selección de señales o características previo se hace necesario.

Tras la realización del estudio de todos los programas y su relación con las variables de las bases de datos, se ha podido llegar a ciertas conclusiones. Como que los programas de clasificación de datos que han conseguido los mejores resultados han sido SVM Linear y los dos KNN, mientras que el resto de los programas han denotado una clara insuficiencia en este tipo de casos. Se constata igualmente que con los coeficientes de aproximación se obtienen tasas de acierto más elevadas y cuando las muestras de entrenamiento están entre las 200 y las 250, se consiguen unos resultados en torno al 90% de tasas de acierto, lo cual para este tipo de estudio es un resultado magnífico y bastante esclarecedor, ya que podemos asegurar que para ese número de muestras de entrenamiento se van a conseguir resultados realmente buenos.

La dificultad en la clasificación del tipo de calentamiento del plasma con otros algoritmos de clasificación pone de manifiesto que el número de señales utilizadas en la base de datos son insuficientes para poder discernir dicha clasificación. Fortalecido además por ser una base de datos fuertemente desbalanceada (muchos más datos de una clase que de otro) y solapada en cuanto a los datos para poder discernir el tipo de clasificación utilizada. Por ejemplo, uno de los mejores resultados obtenidos es que utilizando el kernel lineal y solamente alrededor de 190 descargas como conjunto de entrenamiento para la obtención del modelo, se obtenía un 90% de tasas de acierto en la clasificación de las restantes 304 descargas de test. Tradicionalmente, son los kernel RBF los que mejores márgenes de clasificación y mejores tasas de acierto arrojan.

El hecho de que los kernel RBF no obtuvieran buenos resultados es muy significativo, anticipando que los datos de partida corroboran el desbalanceo y el solapamiento de los datos de confusión utilizados en el presente trabajo.

La información fruto de este análisis ha sido utilizada en la consecución e implementación tanto del TFG titulado "Simulación de señales en plasmas de fusión nuclear mediante técnicas de regresión paramétrica" **[Gilaberte P., 2021]** como en el TFG titulado "Predicción no paramétrica de señales en plasmas de fusión nuclear" **[Martínez Susilla, A., 2022]**. En sendos trabajos se ha utilizado la trasformada wavelet-Haar en la predicción de señales (de forma paramétrica y no paramétrica, respectivamente). No obstante, en lugar de utilizar previamente para el entrenamiento del modelo, toda la información de una señal disponible para todas las descargas de la base de datos se ha optado por hacer un clustering previo de información consistente en buscar solamente las descargas más similares a una dada con el objeto de obtener un modelo más consistente y robusto para poder predecir una señal ausente y no adquirida durante una sesión de operación del TJ-II.

13 Bibliografía

[Aiteco,2022] Aiteco consultores desarrollo y gestión. **Diagrama de dispersión: Relación entre variables.**

https://www.aiteco.com/diagrama-de-dispersion/#:~:text=El%20diagrama%20de%20dispersi%C3%B3n%20permite,existencia%20de%20una%20correlaci%C3%B3n%20positiva.

[Amat Rodrigo,2017] Joaquín Amat Rodrigo. **Máquinas de Vector Soporte (Support Vector Machines, SVMs)**

https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines

[Azor Montoya, 2001] Jesús Rubén Azor Montoya. **La Transformada Wavelet.**

https://webcache.googleusercontent.com/search?q=cache:u9vP1d10LywJ:https://www.um.edu.ar/ojs2019/index.php/RUM/article/view/22/24+&cd=12&hl=es&ct=clnk&gl=es

[Benayas Alamos,2018] Alberto José Benayas Alamos. **Máquinas de soporte vectorial con núcleos de polinomios ortogonales para problemas de clasificación.**

https://oa.upm.es/52249/1/TFM_ALBERTO_BENAYAS_ALAMOS.pdf

[Bermejo, 2017] Guillermo Bermejo Casla. **Diseño de un algoritmo KNN aplicado a la detección de cáncer cerebral mediante imágenes espectrales.**

https://oa.upm.es/52332/1/TFG_GUILLERMO_BERMEJO_CASLA.pdf

[Campo León,2016] Elena Campo León. **Introducción a las máquinas de vector soporte (SVM) en aprendizaje supervisado.**

https://zaguan.unizar.es/record/59156/files/TAZ-TFG-2016-2057.pdf

[Cazorla Piñar,2019] Ignacio Cazorla Piñar. **Aplicación de técnicas de clasificación a la detección de cáncer.**

https://idus.us.es/bitstream/handle/11441/90003/Cazorla%20Pi%C3%B1ar%20Ignacio%20TFG.pdf?sequence=1&isAllowed=y

[Chirinos,2019] Jonathan E. Chirinos Rodríguez. **Series de Fourier**

https://riull.ull.es/xmlui/bitstream/handle/915/15738/Series%20de%20Fourier.pdf;jsessionid=54474A3E6E25063B5B9C605DF34D5370?sequence=1

[Fernández Sarría,2007] Jorge Fernández Sarría. **Estudio de técnicas basadas en la transformada wavelet y optimización de sus parámetros para la clasificación por texturas de imágenes digitales.**

https://riunet.upv.es/bitstream/handle/10251/1955/tesisUPV2573.pdf

[Fuentes López, 2007] C. Fuentes López. **Transmisión del Haz de Neutros de Calentamiento en TJ-II**.

http://www-fusion.ciemat.es/InternalReport/IR1116.pdf

[Gil Martín, 2018] Cristina Gil Martín. **Análisis Discriminante Lineal y Cuadrático.**

https://rpubs.com/Cristina_Gil/389151

[Gilaberte, 2021] Pablo Gilaberte. **Simulación de señales en plasmas de fusión nuclear mediante técnicas de regresión paramétrica**.

https://ebuah.uah.es/dspace/handle/10017/49087

[Goñi Ibaceta,2021] Irene Goñi Ibaceta. **Series de Fourier y sus aplicaciones.**

https://repositorio.unican.es/xmlui/bitstream/handle/10902/23810/GoniIbacetaIrene-TFG-Matematicas.pdf?sequence=1

[ITER, 2015] ITER y Foro Nuclear. **El proyecto de fusión nuclear ITER.**

https://www.foronuclear.org/actualidad/a-fondo/el-proyecto-de-fusion-nuclear-iter/

[Martín Guareño,2016] Juan José Martín Guareño. **Support Vector Regression: Propiedades y aplicaciones.**

https://idus.us.es/bitstream/handle/11441/43808/Mart%C3%ADn%20Guare%C3%B1o%2C%20Juan%20Jos%C3%A9%20TFG.pdf?sequence=1&isAllowed=y

[Martín Martín, 2019] Laura Martín Martín. **Introducción a la teoría de wavelets. Construcción de propiedades de wavelets continuas y discretas.**

https://uvadoc.uva.es/bitstream/handle/10324/38198/TFG-G3590.pdf?sequence=1

[Moujahid et al,2022] Abdelmalik Moujahid, Iñaki Inza y Pedro Larrañaga. **Clasificadores K-NN.**

http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t9knn.pdf

[Olivera,2018] Nahuel Olivera Rodríguez. **Wavelets de Haar y Daubechies y sus aplicaciones.**

http://repositori.uji.es/xmlui/bitstream/handle/10234/177642/TFG_Olivera_Rodriguez%2C_Nahuel.pdf?sequence=1&isAllowed=y

[Pereira,2015] Augusto Pereira González. **Selección de características para el reconocimiento de patrones con datos de alta dimensionalidad en fusión nuclear.**

http://e-spacio.uned.es/fez/view/tesisuned:IngInf-Apereira

[Scitik Internet,2007] Scitik-aprender. **Parámetro RBF SVM.**

https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

[Torrado-Fonseca et al, 2013] Mercedes Torrado-Fonseca y Vanesa Berlanga-Silvente. **Análisis Discriminante mediante SPSS.**

https://redined.educacion.gob.es/xmlui/bitstream/handle/11162/99849/SPSS.pdf?sequence=1&isAllowed=y

Universidad de Alcalá

Escuela Politécnica Superior

ESCUELA POLITECNICA SUPERIOR