



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

**DESARROLLO DE ALGORITMO DE MACHINE LEARNING PARA
MONITOREO DE ESTADO DE SALUD SOBRE SISTEMA HIDRÁULICO**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

IGNACIO DARÍO HUERTA COSSIO

PROFESORA GUÍA:
VIVIANA MERUANE NARANJO

MIEMBROS DE LA COMISIÓN:
RUBÉN FERNÁNDEZ URRUTIA
RODRIGO PASCUAL JIMÉNEZ

SANTIAGO DE CHILE
2021

DESARROLLO DE ALGORITMO DE MACHINE LEARNING PARA MONITOREO DE ESTADO DE SALUD SOBRE SISTEMA HIDRÁULICO

El mantenimiento y la confiabilidad de los activos físicos en las industrias siempre ha representado un gran desafío, y es en este contexto donde en el último tiempo se ha introducido el uso del machine learning para predecir fallas o monitorear el estado de los activos a través de diversos métodos. En el siguiente trabajo de título se estudia un dataset generado por terceros, con información proveniente de un banco de pruebas hidráulico donde se simuló distintos tipos de falla sobre 4 equipos (enfriador, válvula, bomba y acumulador) que componen el sistema y sobre el sistema en sí mismo, buscando obtener los mejores resultados en la clasificación del estado de salud de los equipos y del sistema. Los datos fueron obtenidos a partir de 17 sensores instalados a lo largo del sistema, donde dichos sensores midieron variables como la temperatura o la presión del flujo usando diferentes tasas de muestreo dependiendo del sensor, siendo estas 1, 10 o 100 [Hz]. Las mediciones se llevaron a cabo durante 2205 ciclos de operación con una duración de 60 segundos por ciclo.

Debido a que el dataset contiene series temporales, y debido también a que cada sensor aporta con 60, 600 o 6000 datos por ciclo según su tasa de sampleo, se extrajeron parámetros de tiempo (RMS, media, valor peak to peak y varianza) sobre ventanas de igual largo en tiempo, pero diferente en cantidad de datos, resultando así una cantidad de datos igual para cada sensor. Finalmente, buscando la mejor precisión en las diferentes clasificaciones (accuracy), se probaron diferentes combinaciones de sensores para entrenar los modelos random forest, support vector machines, k-nearest neighbors, linear discriminant analysis y redes neuronales.

Finalmente se obtienen muy buenos resultados para todas las clasificaciones de estado de salud llegando en muchas ocasiones a igualar o acercarse mucho a un accuracy del 100%. Entre las diferentes clasificaciones la que obtuvo mejor desempeño fue la del estado del enfriador la cual llegó en la mayoría de las ocasiones al puntaje perfecto, seguida de la estabilidad del sistema, la fuga en la bomba, el estado del acumulador y el estado de la válvula en orden descendente. De forma global, los resultados entre los parámetros de tiempo RMS, peak to peak y media fueron muy parecidos, mientras que los resultados de la varianza fueron un poco más bajos. Con respecto a los parámetros de tiempo, también se obtuvo que estos debían ser aplicados sobre los 60 segundos de duración del ciclo y no sobre una ventana más pequeña, ya que esto reducía el accuracy. En cuanto a los modelos, para la gran mayoría se consiguieron valores de accuracy sobre el 90%, salvo para la válvula, donde todos mostraron una mayor dificultad para clasificar.

A mi Tata Nano

Agradecimientos

En primer lugar agradezco a mis padres por todos estos años en donde me brindaron educación, amor y un apoyo incondicional. A mi profesor Yuri del colegio, quien fue una de las mayores inspiraciones y motivaciones para estudiar ingeniería en la Universidad de Chile. A mi polola por haberme apoyado en los momentos difíciles de la carrera y por haberme enseñado que la vida no es solo estudiar y trabajar. A la profesora Viviana por ayudarme durante el proceso de mi tesis, teniendo siempre la disposición de responder mis dudas. Y finalmente todos mis compañeros de la Universidad que me acompañaron, apoyaron y ayudaron durante los 6 años y medio de carrera.

Tabla de Contenido

1. Contextualización del trabajo de título	1
1.1. Introducción	1
1.2. Antecedentes generales	2
1.2.1. Antecedentes de los datos a utilizar	2
1.2.2. Antecedentes Machine Learning	3
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
1.4. Alcances	4
2. Antecedentes	5
2.1. Antecedentes del dataset	5
2.2. Conceptos generales machine learning	8
2.2.1. Aprendizaje supervisado y aprendizaje no supervisado	8
2.2.2. Underfitting y overfitting	9
2.2.3. Métricas de desempeño	10
2.2.3.1. Clasificación binaria	10
2.2.3.2. Clasificación no binaria	11
2.2.4. Escalar los datos	11
2.2.5. Ventanas de tiempo y parámetros de tiempo	12
2.3. Dimensionalidad del dataset	14
2.3.1. La maldición de la dimensionalidad	14
2.3.2. Métodos de reducción de dimensionalidad	14
2.3.2.1. Linear Discriminant Analysis	15
2.3.3. Selección hacia adelante	15
2.4. Modelos a usar	16
2.4.1. Random Forests	16
2.4.2. K-Nearest Neighbors	18
2.4.3. Support Vector Machines	18
2.4.4. Redes neuronales	21
2.4.4.1. Hiperparámetros	23
3. Metodología	25
3.1. Recopilación de antecedentes y exploración del dataset	25
3.2. Preprocesamiento de los datos	27
3.2.1. Separación en conjuntos de entrenamiento y testeo	28
3.2.2. Obtención de parámetros de tiempo	28

3.2.3.	Escalamiento de los datos	29
3.2.4.	Cambio en las dimensiones de las etiquetas	29
3.3.	Selección de features y de hiperparámetros	29
3.3.1.	Selección de features	29
3.3.2.	Selección de parámetros	31
3.4.	Análisis de resultados	32
4.	Resultados	33
4.1.	Gráficos de los datos sin procesar	33
4.1.1.	Gráficos de la temperatura	33
4.1.2.	Gráficos de Flujo	37
4.1.3.	Gráficos de Presión	40
4.1.4.	Gráficos de otras variables	46
4.2.	Gráficos de los parámetros de tiempo	52
4.2.1.	Diferencias entre los parámetros de tiempo	52
4.2.2.	Cambios según las diferentes clasificaciones	56
4.2.3.	Cambios diferentes ventanas de tiempo	59
4.3.	Resultados a partir de los modelos LDA	62
4.4.	Resultados a partir de los modelos RF	66
4.5.	Resultados a partir de los modelos KNN	72
4.6.	Resultados a partir de los modelos SVM	77
4.6.1.	Accuracy en función del largo de ventana para diferentes kernels	77
4.6.2.	Heatmaps con la variación de los parámetros C y gamma	79
4.7.	Resultados a partir de los modelos NN	84
4.7.1.	Accuracy en función del largo de ventana para diferentes funciones de activación	84
4.7.2.	Heatmaps con la variación del learning rate y el parámetro alpha	89
5.	Discusión	95
5.1.	Efecto del largo de las ventanas de tiempo	95
5.2.	Diferencias entre parámetros de tiempo	98
5.3.	Efecto del cambio de parámetros de los modelos	100
5.3.1.	Random Forest	100
5.3.2.	K Nearest Neighbors	100
5.3.3.	Linear Discriminant Analysis	100
5.3.4.	Support Vector Machines	100
5.3.5.	Multi layer perceptron (MLP)	101
5.4.	Diferencias entre sensores	102
5.5.	Resultados Generales	103
6.	Conclusiones	104
	Bibliografía	105
	Anexo A. Resultados ampliados	108
A.1.	Gráficos de los datos sin procesar	108
A.1.1.	Gráficos de temperatura	108
A.1.1.1.	Gráficos del enfriador	108

A.1.1.2.	Gráficos de la válvula	111
A.1.1.3.	Gráficos de la bomba	113
A.1.1.4.	Gráficos del acumulador	116
A.1.1.5.	Gráficos de la estabilidad del sistema	118
A.1.2.	Gráficos de flujo	121
A.1.2.1.	Gráficos del enfriador	121
A.1.2.2.	Gráficos de la válvula	123
A.1.2.3.	Gráficos de la bomba	126
A.1.2.4.	Gráficos del acumulador	128
A.1.2.5.	Gráficos de la estabilidad del sistema	131
A.1.3.	Gráficos de presión	134
A.1.3.1.	Gráficos del enfriador	134
A.1.3.2.	Gráficos de la válvula	139
A.1.3.3.	Gráficos de la bomba	144
A.1.3.4.	Gráficos del acumulador	149
A.1.3.5.	Gráficos de la estabilidad del sistema	154
A.1.4.	Gráficos de otras variables	159
A.1.4.1.	Gráficos del enfriador	159
A.1.4.2.	Gráficos de la válvula	164
A.1.4.3.	Gráficos de la bomba	169
A.1.4.4.	Gráficos del acumulador	174
A.1.4.5.	Gráficos de la estabilidad del sistema	179
A.2.	Gráficos de los parámetros de tiempo	184
A.2.1.	Gráficos del enfriador	184
A.2.1.1.	Gráficos usando 1 dato por ciclo	184
A.2.1.2.	Gráficos usando 2 datos por ciclo	188
A.2.1.3.	Gráficos usando 6 datos por ciclo	192
A.2.2.	Gráficos de la válvula	195
A.2.2.1.	Gráficos usando 1 dato por ciclo	196
A.2.2.2.	Gráficos usando 2 datos por ciclo	200
A.2.2.3.	Gráficos usando 6 datos por ciclo	204
A.2.3.	Gráficos de la bomba	207
A.2.3.1.	Gráficos usando 1 dato por ciclo	208
A.2.3.2.	Gráficos usando 2 datos por ciclo	212
A.2.3.3.	Gráficos usando 6 datos por ciclo	216
A.2.4.	Gráficos del acumulador	219
A.2.4.1.	Gráficos usando 1 dato por ciclo	220
A.2.4.2.	Gráficos usando 2 datos por ciclo	224
A.2.4.3.	Gráficos usando 6 datos por ciclo	228
A.2.5.	Gráficos de la estabilidad del sistema	231
A.2.5.1.	Gráficos usando 1 dato por ciclo	232
A.2.5.2.	Gráficos usando 2 datos por ciclo	236
A.2.5.3.	Gráficos usando 6 datos por ciclo	240
A.3.	Resultados a partir de los modelos RF	244
A.4.	Resultados a partir de los modelos KNN	249
A.5.	Resultados a partir de los modelos SVM	254

A.5.1. Accuracy en función de la cantidad de ventanas de entrenamiento para distintos kernel	254
A.5.2. Accuracy al variar los parámetros “C” y “gamma”	259

Índice de Tablas

2.1.	Descripción de los componentes que se clasificaran y sus variables de control. .	6
2.2.	Clases correspondientes a las clasificaciones del estado de salud de los componentes y del sistema	7
2.3.	Resumen de los sensores utilizados para adquirir los datos.	8
2.4.	Ejemplo de ventanas de tiempo con overlap	12
3.1.	Dimensiones de los archivos cargados.	26
3.2.	Parámetros usados para la función <code>train_test_split</code>	28
3.3.	Combinaciones de largos de ventana y overlap usados para generar los conjuntos de entrenamiento y testeo.	29
3.4.	Modelos usados para realizar la selección de sensores.	30
3.5.	Parámetros usados en la selección de sensores para los modelos <code>MLPClassifier</code>	31
4.1.	Combinaciones parámetro de tiempo-cantidad de datos por ciclo para probar los parámetros alpha y learning rate	89
5.1.	Resultados globales	103

Índice de Ilustraciones

1.1.	Layout del sistema hidráulico. Fuente [1]	2
2.1.	Layout del sistema hidráulico por partes. Fuente [1]	6
2.2.	Ejemplo de datos etiquetados y datos no etiquetados. Fuente [29]	9
2.3.	Ejemplo de overfitting. Fuente [30]	10
2.4.	Ejemplo de una matriz de confusión. Fuente [31]	10
2.5.	Evolución del desempeño de un modelo al aumentar la dimensionalidad del dataset. Fuente [32]	14
2.6.	Ejemplo de LDA. Fuente [13]	15
2.7.	Ejemplo de un árbol de decisión. Elaboración propia.	16
2.8.	Árbol de decisión vs Random Forest. Fuente [34]	17
2.9.	Esquema de funcionamiento de un Random Forest. Fuente [15]	17
2.10.	Esquema del algoritmo de clasificación KNN, con k=3. Fuente [17]	18
2.11.	Izquierda, hiperplanos dividiendo un espacio bidimensional, derecha, hiperplano dividiendo un espacio tridimensional. Fuente [19]	18
2.12.	Izquierda, posibles hiperplanos, derecha, hiperplano óptimo. Fuente [19]	19
2.13.	En este caso en la izquierda un hiperplano no es capaz de dividir los datos en el espacio bidimensional, pero si se agrega una dimensión, derecha, un hiperplano es capaz de separar los datos. Fuente [20]	20
2.14.	Hiperplano computado usando un Kernel Gaussiano. Fuente [35]	21
2.15.	Resultados de una red neuronal. Fuente [25]	22
2.16.	Esquema de una neurona artificial. Fuente [36]	22
2.17.	Esquema de una red neuronal. Fuente [37]	23
2.18.	Funciones de activación. Elaboración propia	24
4.1.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado enfriador.	34
4.2.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 4.	35
4.3.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 2.	35
4.4.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba.	36
4.5.	Comparación entre las curvas de los estados “sin fuga” y “fuga leve” en distintas iteraciones.	36
4.6.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador.	37
4.7.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador.	37

4.8.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador.	38
4.9.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula.	38
4.10.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba.	39
4.11.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador.	39
4.12.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema.	40
4.13.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador.	41
4.14.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 2.	42
4.15.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 5.	43
4.16.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la fuga en la bomba.	44
4.17.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador.	45
4.18.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema.	46
4.19.	Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases del estado del enfriador.	47
4.20.	Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases del estado de la válvula.	48
4.21.	Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases de la fuga en la bomba.	49
4.22.	Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases del estado del acumulador.	50
4.23.	Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases de la estabilidad del sistema.	51
4.24.	Gráficos del RMS obtenido según las clases del estado del enfriador.	52
4.25.	Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.	53
4.26.	Gráficos de la media obtenida según las clases del estado del enfriador.	53
4.27.	Gráficos de la varianza obtenida según las clases del estado del enfriador.	54
4.28.	Gráficos del RMS obtenido según las clases del estado de la válvula.	54
4.29.	Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.	55
4.30.	Gráficos de la media obtenida según las clases del estado de la válvula.	55
4.31.	Gráficos de la varianza obtenida según las clases del estado de la válvula.	56
4.32.	Gráficos del RMS obtenido según las clases de la fuga en la bomba.	57
4.33.	Gráficos de la varianza obtenida según las clases de la fuga en la bomba.	57
4.34.	Gráficos del RMS obtenido según las clases del estado del acumulador.	58
4.35.	Gráficos de la varianza obtenida según las clases del estado del acumulador.	58
4.36.	Gráficos del RMS obtenido según las clases del estado del enfriador, usando 2 datos por ciclo.	59

4.37.	Gráficos del RMS obtenido según las clases del estado del enfriador, usando 6 datos por ciclo.	60
4.38.	Gráficos del RMS obtenido según las clases del estado de la válvula, usando 2 datos por ciclo.	60
4.39.	Gráficos del RMS obtenido según las clases del estado de la válvula, usando 6 datos por ciclo.	61
4.40.	Valores de accuracy para modelos LDA.	62
4.41.	Valores de accuracy para modelos LDA.	63
4.42.	Valores de accuracy para modelos LDA.	63
4.43.	Valores de accuracy para modelos LDA.	64
4.44.	Valores de accuracy para modelos LDA.	65
4.45.	Valores de accuracy obtenidos de modelos RF.	66
4.46.	Valores de accuracy para la clasificación del estado del enfriador a partir de modelos RF usando datos de varianza.	67
4.47.	Valores de accuracy obtenidos de modelos RF.	67
4.48.	Valores de accuracy obtenidos de modelos RF.	68
4.49.	Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.48	69
4.50.	Valores de accuracy obtenidos de modelos RF.	69
4.51.	Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.50.	70
4.52.	Valores de accuracy obtenidos de modelos RF.	71
4.53.	Valores de accuracy obtenidos de modelos KNN.	72
4.54.	Valores de accuracy obtenidos de modelos KNN.	73
4.55.	Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.54.	73
4.56.	Valores de accuracy obtenidos de modelos KNN.	74
4.57.	Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.57.	75
4.58.	Valores de accuracy obtenidos de modelos KNN.	75
4.59.	Valores de accuracy obtenidos de modelos KNN.	76
4.60.	Valores de accuracy obtenidos de modelos SVM.	77
4.61.	Valores de accuracy obtenidos de modelos SVM.	78
4.62.	Valores de accuracy obtenidos de modelos SVM.	78
4.63.	Valores de accuracy obtenidos de modelos SVM.	78
4.64.	Valores de accuracy obtenidos de modelos SVM.	79
4.65.	Heatmap con la variación del accuracy en función de los parámetros c y gamma.	79
4.66.	Heatmap con la variación del accuracy en función de los parámetros c y gamma.	80
4.67.	Heatmap con la variación del accuracy en función de los parámetros c y gamma.	81
4.68.	Heatmap con la variación del accuracy en función de los parámetros c y gamma.	82
4.69.	Heatmap con la variación del accuracy en función de los parámetros c y gamma.	83
4.70.	Accuracy obtenido a partir de modelos MLP.	84
4.71.	Accuracy obtenido a partir de modelos MLP.	85
4.72.	Accuracy obtenido a partir de modelos MLP.	86
4.73.	Accuracy obtenido a partir de modelos MLP.	87
4.74.	Accuracy obtenido a partir de modelos MLP.	88

4.75.	Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.	90
4.76.	Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.	91
4.77.	Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.	92
4.78.	Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.	93
4.79.	Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.	94
5.1.	Comparación entre gráficos con igual parámetro de tiempo y clasificación, pero diferentes ventanas de tiempo.	96
5.2.	Accuracy de modelos KNN obtenido según las clases del estado de la válvula, usando RMS.	97
5.3.	Comparación entre los parámetros: Media, RMS y valor peak to peak, usando el mismo largo de ventana de tiempo y los mismos 3 sensores.	98
5.4.	Comparación entre gráficos con igual parámetro de tiempo y clasificación, pero diferentes ventanas de tiempo.	99
A.1.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado enfriador, iteración 1.	108
A.2.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 2.	109
A.3.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 3.	109
A.4.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 4.	110
A.5.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 5.	110
A.6.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 1.	111
A.7.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 2.	111
A.8.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 3.	112
A.9.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 4.	112
A.10.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 5.	113
A.11.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 1.	113
A.12.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 2.	114
A.13.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 3.	114
A.14.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 4.	115

A.15.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 5.	115
A.16.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 1.	116
A.17.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 2.	116
A.18.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 3.	117
A.19.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 4.	117
A.20.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 5.	118
A.21.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 1.	118
A.22.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 2.	119
A.23.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 3.	119
A.24.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 4.	120
A.25.	Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 5.	120
A.26.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 1.	121
A.27.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 2.	121
A.28.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 3.	122
A.29.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 4.	122
A.30.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 5.	123
A.31.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 1.	123
A.32.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 2.	124
A.33.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 3.	124
A.34.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 4.	125
A.35.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 5.	125
A.36.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 1.	126
A.37.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 2.	126

A.38.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 3.	127
A.39.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 4.	127
A.40.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 5.	128
A.41.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 1.	128
A.42.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 2.	129
A.43.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 3.	129
A.44.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 4.	130
A.45.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 5.	130
A.46.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 1.	131
A.47.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 2.	131
A.48.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 3.	132
A.49.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 4.	132
A.50.	Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 5.	133
A.51.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 1.	134
A.52.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 2.	135
A.53.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 3.	136
A.54.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 4.	137
A.55.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 5.	138
A.56.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 1.	139
A.57.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 2.	140
A.58.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 3.	141
A.59.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 4.	142
A.60.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 5.	143

A.61.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 1.	144
A.62.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 2.	145
A.63.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 3.	146
A.64.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 4.	147
A.65.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 5.	148
A.66.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 1.	149
A.67.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 2.	150
A.68.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 3.	151
A.69.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 4.	152
A.70.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 5.	153
A.71.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 1.	154
A.72.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 2.	155
A.73.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 3.	156
A.74.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 4.	157
A.75.	Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 5.	158
A.76.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 1.	159
A.77.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 2.	160
A.78.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 3.	161
A.79.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 4.	162
A.80.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 5.	163

A.81.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 1.	164
A.82.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 2.	165
A.83.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 3.	166
A.84.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 4.	167
A.85.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 5.	168
A.86.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 1.	169
A.87.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 2.	170
A.88.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 3.	171
A.89.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 4.	172
A.90.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 5.	173
A.91.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 1.	174
A.92.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 2.	175
A.93.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 3.	176
A.94.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 4.	177
A.95.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 5.	178

A.96.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 1.	179
A.97.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 2.	180
A.98.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 3.	181
A.99.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 4.	182
A.100.	Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 5.	183
A.101.	Gráficos del RMS obtenido según las clases del estado del enfriador.	184
A.102.	Gráficos de la varianza obtenida según las clases del estado del enfriador.	185
A.103.	Gráficos de la media obtenida según las clases del estado del enfriador.	186
A.104.	Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.	187
A.105.	Gráficos del RMS obtenido según las clases del estado del enfriador.	188
A.106.	Gráficos de la varianza obtenida según las clases del estado del enfriador.	189
A.107.	Gráficos de la media obtenida según las clases del estado del enfriador.	190
A.108.	Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.	191
A.109.	Gráficos del RMS obtenido según las clases del estado del enfriador.	192
A.110.	Gráficos de la varianza obtenida según las clases del estado del enfriador.	193
A.111.	Gráficos de la media obtenida según las clases del estado del enfriador.	194
A.112.	Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.	195
A.113.	Gráficos del RMS obtenido según las clases del estado de la válvula.	196
A.114.	Gráficos de la varianza obtenida según las clases del estado de la válvula.	197
A.115.	Gráficos de la media obtenida según las clases del estado de la válvula.	198
A.116.	Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.	199
A.117.	Gráficos del RMS obtenido según las clases del estado de la válvula.	200
A.118.	Gráficos de la varianza obtenida según las clases del estado de la válvula.	201
A.119.	Gráficos de la media obtenida según las clases del estado de la válvula.	202
A.120.	Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.	203
A.121.	Gráficos del RMS obtenido según las clases del estado de la válvula.	204
A.122.	Gráficos de la varianza obtenida según las clases del estado de la válvula.	205
A.123.	Gráficos de la media obtenida según las clases del estado de la válvula.	206
A.124.	Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.	207
A.125.	Gráficos del RMS obtenido según las clases del estado de la bomba.	208
A.126.	Gráficos de la varianza obtenida según las clases del estado de la bomba.	209
A.127.	Gráficos de la media obtenida según las clases del estado de la bomba.	210
A.128.	Gráficos del valor peak to peak obtenido según las clases del estado de la bomba.	211
A.129.	Gráficos del RMS obtenido según las clases del estado de la bomba.	212
A.130.	Gráficos de la varianza obtenida según las clases del estado de la bomba.	213
A.131.	Gráficos de la media obtenida según las clases del estado de la bomba.	214
A.132.	Gráficos del valor peak to peak obtenido según las clases del estado de la bomba.	215

A.133. Gráficos del RMS obtenido según las clases del estado de la bomba.	216
A.134. Gráficos de la varianza obtenida según las clases del estado de la bomba.	217
A.135. Gráficos de la media obtenida según las clases del estado de la bomba.	218
A.136. Gráficos del valor peak to peak obtenido según las clases del estado de la bomba.	219
A.137. Gráficos del RMS obtenido según las clases del estado del acumulador.	220
A.138. Gráficos de la varianza obtenida según las clases del estado del acumulador.	221
A.139. Gráficos de la media obtenida según las clases del estado del acumulador.	222
A.140. Gráficos del valor peak to peak obtenido según las clases del estado del acumulador.	223
A.141. Gráficos del RMS obtenido según las clases del estado del acumulador.	224
A.142. Gráficos de la varianza obtenida según las clases del estado del acumulador.	225
A.143. Gráficos de la media obtenida según las clases del estado del acumulador.	226
A.144. Gráficos del valor peak to peak obtenido según las clases del estado del acumulador.	227
A.145. Gráficos del RMS obtenido según las clases del estado del acumulador.	228
A.146. Gráficos de la varianza obtenida según las clases del estado del acumulador.	229
A.147. Gráficos de la media obtenida según las clases del estado del acumulador.	230
A.148. Gráficos del valor peak to peak obtenido según las clases del estado del acumulador.	231
A.149. Gráficos del RMS obtenido según las clases de la estabilidad del sistema.	232
A.150. Gráficos de la varianza obtenida según las clases de la estabilidad del sistema.	233
A.151. Gráficos de la media obtenida según las clases de la estabilidad del sistema.	234
A.152. Gráficos del valor peak to peak obtenido según las clases de la estabilidad del sistema.	235
A.153. Gráficos del RMS obtenido según las clases de la estabilidad del sistema.	236
A.154. Gráficos de la varianza obtenida según las clases de la estabilidad del sistema.	237
A.155. Gráficos de la media obtenida según las clases de la estabilidad del sistema.	238
A.156. Gráficos del valor peak to peak obtenido según las clases de la estabilidad del sistema.	239
A.157. Gráficos del RMS obtenido según las clases de la estabilidad del sistema.	240
A.158. Gráficos de la varianza obtenida según las clases de la estabilidad del sistema.	241
A.159. Gráficos de la media obtenida según las clases de la estabilidad del sistema.	242
A.160. Gráficos del valor peak to peak obtenido según las clases de la estabilidad del sistema.	243
A.161. Valores de accuracy obtenidos de modelos RF.	244
A.162. Valores de accuracy obtenidos de modelos RF.	245
A.163. Valores de accuracy obtenidos de modelos RF.	246
A.164. Valores de accuracy obtenidos de modelos RF.	247
A.165. Valores de accuracy obtenidos de modelos RF.	248
A.166. Valores de accuracy obtenidos de modelos KNN.	249
A.167. Valores de accuracy obtenidos de modelos KNN.	250
A.168. Valores de accuracy obtenidos de modelos KNN.	251
A.169. Valores de accuracy obtenidos de modelos KNN.	252
A.170. Valores de accuracy obtenidos de modelos KNN.	253
A.171. Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.	254
A.172. Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.	255
A.173. Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.	256

A.174. Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.	257
A.175. Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.	258
A.176. Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel ‘rbf’ y 1 dato extraído por ciclo.	259
A.177. Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel ‘rbf’ y 1 dato extraído por ciclo.	260
A.178. Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel ‘rbf’ y 1 dato extraído por ciclo.	261
A.179. Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel ‘rbf’ y 1 dato extraído por ciclo.	262
A.180. Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel ‘rbf’ y 1 dato extraído por ciclo.	263

Capítulo 1

Contextualización del trabajo de título

1.1. Introducción

A lo largo del tiempo, y en conjunto al desarrollo tecnológico, las estrategias de mantenimiento y gestión de activos físicos han evolucionado, tomando como punto de partida el mantenimiento correctivo, para llegar a estrategias que incluyen trabajos planificados tales como el mantenimiento preventivo, predictivo y proactivo. Esta evolución le ha permitido a las industrias mantener con mayor eficiencia las etapas de sus procesos productivos, ya que cada vez se hace posible debido a las herramientas tecnológicas, realizar trabajos de mantenimiento minimizando el tiempo de interrupción de la producción.

En este contexto, además se persigue el objetivo de maximizar el tiempo de operación de los componentes, sujetos a mantenimiento, cuando aún no han culminado su vida útil, para optimizar el uso de recursos. Es por esto que uno de los grandes desafíos en los procesos productivos actuales, es evitar las fallas en los diversos componentes que operan en las distintas etapas de un proceso, para así solucionar problemas altamente recurrentes tales como los sobrecostos, y en consecuencia la reducción de las utilidades de las empresas a cargo, debido a una alta tasa de reemplazo de componentes o las detenciones imprevistas que se traducen en una pérdida de productividad o incluso en accidentes fatales producto de las fallas que ocasionan estas detenciones.

Sobre la base de lo anteriormente mencionado, se ha evolucionado hacia la implementación de estrategias de mantenimiento predictivo, el cual utiliza monitoreos continuos de los equipos para predecir y detectar fallas, con lo cual se planifican y ejecutan los trabajos de mantenimiento de una forma más eficiente en cuanto al uso de los tiempos de producción. Esta clase de mantenimiento se basa en el monitoreo de las condiciones operacionales de los equipos y componentes de las distintas etapas de un proceso, para así estimar el tiempo en el que se producirán las fallas, permitiendo obtener un mayor control sobre los estados de salud de los activos. Para esto se utilizan diversas, y cada vez más sofisticadas, técnicas que permiten realizar estos monitoreos, entre las cuales se encuentran técnicas tales como análisis de vibraciones, análisis de aceite, mediciones con ultrasonido, termografías, entre otras.

Gracias al constante desarrollo de tecnologías, es posible estudiar la información obtenida de las técnicas mencionadas anteriormente, mediante métodos de procesamiento de datos y análisis provenientes del Machine Learning y Deep Learning, las cuales son áreas de las cien-

cias de la computación que permiten que los computadores generalicen o “aprendan” de los datos que se les entrega, otorgando así un análisis mucho mas profundo de los estados de salud de los activos para decidir, de formas cada vez mas eficientes, los planes de mantenimiento de un proceso productivo.

Dentro del contexto actual de mantenimiento, una de las tareas más utilizadas, para lo cual se implementan técnicas de Machine Learning y Deep Learning, es predecir mediante diversos algoritmos, la clase a la cual pertenecen un grupo de datos medidos, basándose en datos anteriores. Para el trabajo de título actual, se utilizarán dos tipos de algoritmos de clasificación de datos: Algoritmos de árboles de decisión y algoritmos de redes neuronales.

1.2. Antecedentes generales

1.2.1. Antecedentes de los datos a utilizar

Los datos que serán utilizados para el trabajo de título fueron generados por un estudio previo de Helwig et al.[1] a partir de un banco de pruebas de un sistema hidráulico (ver figura 1.1) compuesto por un sistema primario y un sistema secundario. El objetivo del banco de pruebas, fue obtener un dataset que sirviese para entrenar y testear algoritmos de machine learning destinados a clasificar los distintos estados de salud de algunos componentes del sistema, midiendo los cambios en ciertas variables de operación. Para generar los datos, se realizaron 2205 ciclos de operación en el sistema hidráulico, donde durante cada ciclo se simularon diferentes estados de salud reversibles en 4 componentes y en el sistema. Para medir los cambios en las distintas variables de operación del sistema como temperatura, presión y vibración se usaron 17 sensores instalados en el banco de pruebas.

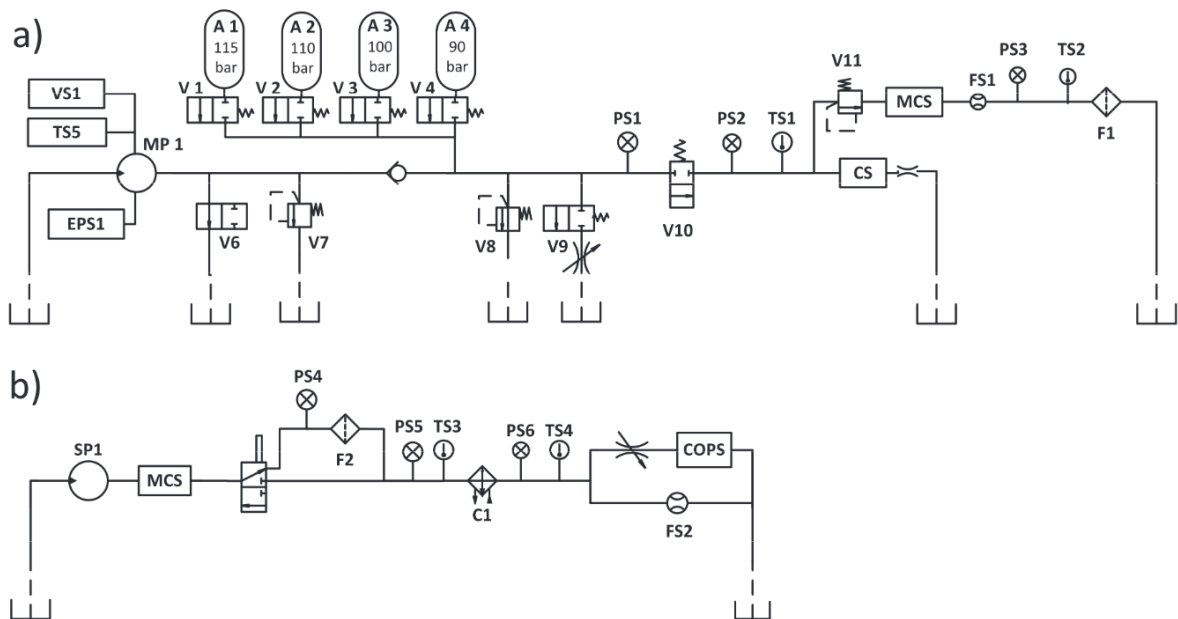


Figura 1.1: Layout del sistema hidráulico. Fuente [1]

1.2.2. Antecedentes Machine Learning

El machine learning (ML) es una área de la Inteligencia Artificial(IA) que permite a un sistema generalizar o aprender de los datos en lugar de aprender mediante la programación explícita. Un modelo de ML es el resultado de ajustar los parámetros de un algoritmo de ML usando un subconjunto de datos llamado conjunto de entrenamiento, tras lo cual el modelo debiese ser capaz de predecir el comportamiento de nuevos datos. Entre las principales tareas para lo cual se utiliza el ML es predecir la clase a la que pertenecen ciertos datos, y en particular en este trabajo de título, se utilizará para clasificar los diferentes estados de salud que presenta el sistema hidráulico y sus componentes a lo largo de las pruebas realizadas.

1.3. Objetivos

1.3.1. Objetivo General

Implementar una metodología de aprendizaje de máquinas para predecir el estado de salud en un sistema hidráulico.

1.3.2. Objetivos Específicos

1. Generar diferentes sets de entrenamiento y testeo para los modelos preprocesando el dataset con diferentes tamaños de ventanas temporales y diferentes parámetros de tiempos (RMS, varianza, etc.)
2. Probar diferentes modelos de Machine Learning y seleccionar las mejores combinaciones de features (sensores) para cada caso.
3. Probar diferentes hiperparámetros para los modelos y sus respectivos features seleccionados.
4. Seleccionar las combinaciones de modelos con los mejores resultados y compararlos con los resultados obtenidos en el paper de Helwig[1] y con otros benchmarks sobre el mismo dataset.

1.4. Alcances

- Uso de modelos: Linear Discriminant Analysis, Random Forest, K Nearest Neighbors, Support Vector Machines y Multi Layer Perceptron.
- Obtención de múltiples resultados para las 5 clasificaciones de estado de salud obtenidos a partir de las diversas combinaciones de modelos.
- Comparación de resultados con otros benchmarks del mismo dataset.

Capítulo 2

Antecedentes

2.1. Antecedentes del dataset

Como se adelantó en el capítulo anterior, el dataset con el cual se trabajará se obtuvo a partir de un montaje experimental. El montaje consiste en un banco de pruebas hidráulico que a su vez consta de un circuito primario de trabajo (ver figura 2.1.a) y uno secundario de enfriamiento-filtrado (ver figura 2.1.b) que están conectados a través del tanque de aceite. El sistema repite cíclicamente ciclos de carga constante (duración 60 segundos) y mide valores de proceso como presiones, caudales volumétricos y temperaturas, mientras que la condición de cuatro componentes hidráulicos (enfriador, válvula, bomba y acumulador) varía cuantitativamente. Los componentes del sistema se distribuyen como sigue:

- **MP:** Bomba principal.
- **A:** Acumulador hidráulico.
- **VS:** Sensor de vibración.
- **TS:** Sensor de temperatura.
- **EPS:** Sensor de potencia eléctrica.
- **V:** Válvula.
- **PS:** Sensor de presión.
- **MCS:** Sensor de contaminación de partículas.
- **CS:** Sensor de contaminación de partículas.
- **FS:** Sensor de caudal.
- **F:** No especificado.
- **SP:** No especificado.
- **C:** Enfriador.
- **COPS:** Monitoreo de aceite.

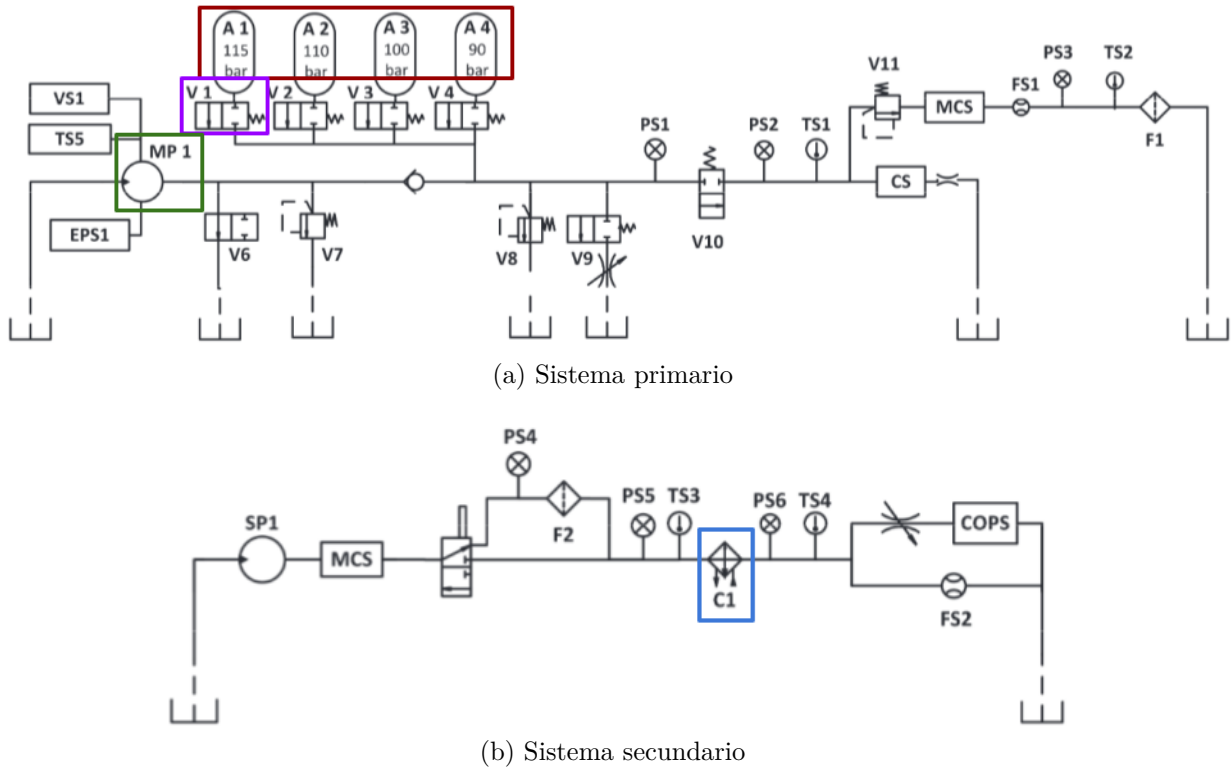


Figura 2.1: Layout del sistema hidráulico por partes. Fuente [1]

En la tabla 2.1 se resumen los componentes, su TAG en el layout, una breve descripción de su estado de salud y su variable de control:

Tabla 2.1: Descripción de los componentes que se clasificaran y sus variables de control.

Componente	TAG	Condición	Variable de control
Enfriador	C1	Potencia decreciente	Ciclo de trabajo del ventilador del enfriador
Valvula	V10	Degradación del switch	Corriente de V10
Bomba	MP1	Fuga interna	Orificios de derivación conmutables
Acumulador	A1-A4	Fuga de gas	Presiones en cada acumulador

En la tabla 2.2 se detallan las diferentes clases que corresponden a los estados de salud de los componentes del sistema y del mismo sistema. En cuanto a la adquisición de los datos, en el sistema se instalaron 17 sensores para medir diferentes variables de operación, estos sensores funcionaron con una tasa de muestreo de 1, 10 o 100 [Hz], con lo cual considerando un ciclo de 60 segundos, se obtuvieron 60, 600 o 6000 datos respectivamente. Los sensores midieron durante 2205 ciclos y la clase de cada uno de los componentes del sistema se mantuvo fija a lo largo de cada ciclo. En la tabla 2.3 se detallan la notación de cada sensor usada en el diagrama del layout, su respectiva tasa de muestreo y su respectiva unidad de medida.

Los datos usados se encuentran disponibles en: <https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems>. Los datos de obtenidos por los sensores se encuentran separados por sensor en arreglos con dimensión de (2205, 60 *tasa de muestreo), mientras que las etiquetas se encuentran almacenadas en un solo archivo con dimensión (2205,5) correspondiente a las 5 clasificaciones. Todos los archivos están en formato .txt.

Tabla 2.2: Clases correspondientes a las clasificaciones del estado de salud de los componentes y del sistema

Clasificacion	Clases	Ciclos por clase
Estado del enfriador	Cerca de la falla total	732
	Eficiencia reducida	732
	Eficiencia total	741
Fuga en la bomba	Sin fuga	1221
	Poca fuga	492
	Cerca de la falla total	492
Estado de la valvula	Comportamiento optimo	1125
	Poco retraso	360
	Retraso severo	360
	Cerca de la falla total	360
Estado del acumulador	Presion optima	599
	Presion ligeramente reducida	399
	Presion severamente reducida	399
	Cerca de la falla total	808
Estabilidad del sistema	Sistema estable	1449
	Estabilidad no alcanzada	756

Tabla 2.3: Resumen de los sensores utilizados para adquirir los datos.

Nombre del sensor	Abreviación	Tasa de muestreo [Hz]	Unidad de medida
Sensor de temperatura 1	TS1	1	C
Sensor de temperatura 2	TS2	1	C
Sensor de temperatura 3	TS3	1	C
Sensor de temperatura 4	TS3	1	C
Sensor de vibración 1	VS1	1	mm/s
Eficiencia del enfriador	CE*	1	%
Potencia del enfriador	CP	1	kW
Factor de eficiencia	SE*	1	%
Sensor de flujo 1	FS1	10	L/min
Sensor de flujo 2	FS2	10	L/min
Sensor de presión 1	PS1	100	bar
Sensor de presión 2	PS2	100	bar
Sensor de presión 3	PS3	100	bar
Sensor de presión 4	PS4	100	bar
Sensor de presión 5	PS5	100	bar
Sensor de presión 6	PS6	100	bar
Potencia del motor	EPS1	100	W

*: Estos son sensores virtuales, cuyos datos provienen de modelos físicos que usaron datos de otros sensores.

2.2. Conceptos generales machine learning

Como ya se mencionó anteriormente, el ML será la herramienta que se utilizará para procesar los datos provenientes de los sensores y así realizar las clasificaciones de los estados de salud del sistema y sus componentes. En esta sección se abordan conceptos generales acerca del ML para dar contexto al trabajo realizado.

2.2.1. Aprendizaje supervisado y aprendizaje no supervisado

Una de las principales categorías para clasificar los diferentes algoritmos de ML, es de acuerdo a si los datos ya están asociados previamente a una categoría o valor que se quiera predecir (aprendizaje supervisado) o si no lo están (aprendizaje no supervisado). En la figura 2.2 se muestra un ejemplo ilustrativo donde puede ver como independiente del cluster al cual pertenecen los datos, en el caso supervisado, se tienen previamente identificadas las clases de los datos, mientras que en el caso no supervisado no es así.

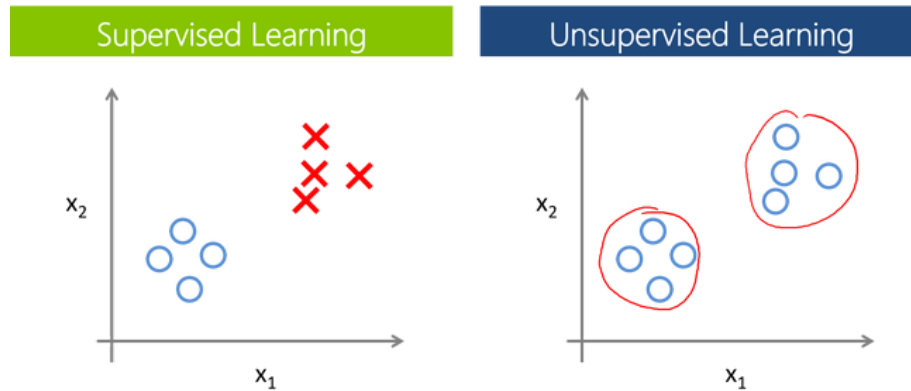


Figura 2.2: Ejemplo de datos etiquetados y datos no etiquetados. Fuente [29]

Aprendizaje Supervisado: El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso de analítica, estos datos tienen características etiquetadas que definen el significado de los datos. Por ejemplo, esta técnica se ha usado para detectar “fake-news”, usando noticias previamente clasificadas como falsas y otras verdaderas para luego usar métodos como k-Nearest Neighbors, Random Forests, Support Vector Machine y otros[2].

Aprendizaje No-Supervisado: Al igual que el aprendizaje supervisado busca encontrar patrones en los datos, pero estos no tienen etiquetas que le den un significado a estos patrones, por lo que lo que se trata de hacer es crear cúmulos de datos (clusters) que tengan datos con patrones similares, pudiendo identificar datos atípicos de esta manera u conjuntos de interés. Por ejemplo, se ha usado para poder detectar y clasificar los sonidos de las ballenas azules[3].

2.2.2. Underfitting y overfitting

Al entrenar modelos predictivos un gran desafío es encontrar los hiperparámetros que nos permitan generalizar el comportamiento de los datos y no ajustarse a un comportamiento en especial que estos puedan presentar. El problema descrito anteriormente se conoce como overfitting y consiste en que el modelo entrenado sobre aprende del conjunto de entrenamiento y luego no consigue ajustarse bien al tratar de predecir con otros conjuntos de datos (ver figura 2.3).

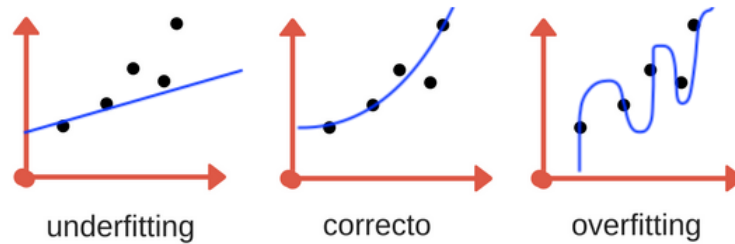


Figura 2.3: Ejemplo de overfitting. Fuente [30]

2.2.3. Métricas de desempeño

Usualmente para evaluar el desempeño de un modelo de clasificación, se utiliza la matriz de confusión, la cual nos muestra gráficamente la cantidad de datos que fueron bien y mal clasificados.

2.2.3.1. Clasificación binaria

Como ejemplo en la figura 2.4 se presenta un matriz de confusión para una clasificación binaria (2 clases):

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figura 2.4: Ejemplo de una matriz de confusión. Fuente [31]

En la matriz se pueden distinguir 2 clases: la clase positiva y la clase negativa, junto con los ejes de clase predicha y clase verdadera. En base a lo anterior se crean 4 conjuntos de datos:

- **True positive (TP):** Son los datos que realmente pertenecían a la clase positiva y fueron bien clasificados como positivos.
- **False positive (FP):** Son los datos que realmente pertenecían a la clase negativa y fueron mal detectados como positivos.

- **True negative (TN):** Son los datos que realmente pertenecían a la clase negativa y fueron bien clasificados como negativos.
- **False negative (FN):** Son los datos que realmente pertenecían a la clase positiva y fueron mal detectados como negativos.

A partir de estos conjuntos, se suelen extraer 3 ratios que evalúan el desempeño de la clasificación:

$$Accuracy = \frac{\text{Casos bien predichos}}{\text{Casos totales}} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Sensitivity = \frac{\text{Casos positivos bien predichos}}{\text{Casos positivos totales}} = \frac{TP}{TP + FN}$$

$$Specificity = \frac{\text{Casos negativos bien predichos}}{\text{Casos negativos totales}} = \frac{TN}{TN + FP}$$

La primera métrica nos indica el desempeño del modelo con respecto a las 2 clases que se están clasificando, mientras que *sensitivity* mide el desempeño al clasificar la clase positiva y *specificity* mide el desempeño al clasificar la clase negativa.

2.2.3.2. Clasificación no binaria

Debido a que en este trabajo de título la mayoría de las clasificaciones son NO binarias (3 o más clases), solo se utilizará la métrica **accuracy**. En este caso, si se usa una matriz de confusión, los casos bien predichos corresponderán a la suma de la diagonal de la matriz.

$$Accuracy = \frac{\text{Casos bien predichos}}{\text{Casos totales}} = \frac{\text{Casos bien predichos}}{\text{Suma diagonal}}$$

2.2.4. Escalar los datos

Escalar los datos se refiere a cambiar el rango de las variables que conforman los datos, para poder compararlas con las otras variables y que los modelo de machine learning puedan comparar de manera justa las variables que conforman los datos. Ya que, por ejemplo, si tenemos datos que tienen variables de temperatura en °C y de presión en bar, los modelos de machine learning ponderarían la diferencia de 1 [°C] igual que la de 1 [bar], aun cuando la temperatura esta en un rango de 0 [°C] a 100 [°C] y la presión de 0.5 [bar] a 4 [bar], lo cual implicaría que la diferencia de 1 [°C] corresponde a una diferencia de 1 % de la temperatura con respecto a los otros datos y la de 1 [bar], a una del 28 %, mucho mas grande que la de la temperatura. Por esto es necesario escalar los datos respecto a los demás datos, así se trabaja con diferencias que son comparables entre variables. Uno de los escalamientos más comunes corresponde al de escala mínima-máxima, la cual posee la siguiente definición[12](Siendo X la variable a escalar):

$$X_{escalado} = \frac{X_{no-escalado} - X_{min}}{X_{max} - X_{min}}$$

La cual cambia el rango original de la variable de $[X_{min}, X_{max}]$ a $[0,1]$.

2.2.5. Ventanas de tiempo y parámetros de tiempo

Ventanas de tiempo y overlap: En los problemas donde se trabaja con datos de series temporales, es común que al procesar los datos se alargue el conjunto de entrenamiento usando overlap. El overlap o superposición, consiste en que las ventanas de tiempo que se toman a lo largo del dataset compartan algunos datos con el fin de aumentar las dimensiones de los conjuntos de datos y así lograr mejores desempeños de los modelos. Pensemos por ejemplo en un arreglo de números Xt :

$$Xt = X1, X2, X3, \dots, X14.$$

Si nos quedamos con este arreglo tal como esta, tendríamos un total de 14 datos, sin embargo si por ejemplo tomásemos ventanas de 8 datos con un overlap de 5 datos (ver tabla 2.4) y concatenamos estos arreglos, terminaríamos con un arreglo de largo 24.

Tabla 2.4: Ejemplo de ventanas de tiempo con overlap

	Dato 1	Dato 2	Dato 3	Dato 4	Dato 5	Dato 6	Dato 7	Dato 8
Ventana 1	X1	X2	X3	X4	X5	X6	X7	X8
Ventana 2	X4	X5	X6	X7	X8	X9	X10	X11
Ventana 3	X7	X8	X9	X10	X11	X12	X13	X14

Parámetros de tiempo: Otra práctica común con los datos temporales, es extraer parámetros de tiempo a partir de estos. Los parámetros de tiempo usados fueron:

Sea Xt un vector de tiempo de largo n :

$$Xt = x_1, x_2, x_3, \dots, x_n$$

- Root mean square (RMS)

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$$

- Valor peak to peak ($P2P$)

$$P2P = \max(Xt) - \min(Xt)$$

- Media (μ)

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

- Varianza (σ)

$$\sigma = \frac{1}{n} \sum_{i=1}^n (\mu - x_i)^2$$

Particularmente en el caso de este trabajo de título, se aplicarán los parámetros de tiempo sobre ventanas de igual tiempo, pero distinto largo, ya que los sensores entregan una cantidad diferente de datos en un mismo tiempo al tener tasas de muestreo diferentes.

2.3. Dimensionalidad del dataset

En algunas ocasiones, y en particular en el problema que se aborda en este trabajo de título, se dispone de una gran cantidad de features (sensores en nuestro caso) y resulta conveniente reducir la dimensionalidad del problema con el fin de **mejorar el desempeño del modelo**, reducir los tiempo de computación, o bien con el fin de poder visualizar de mejor forma los datos [4]

2.3.1. La maldición de la dimensionalidad

Generalmente el aumentar la cantidad de datos que se disponen trae consigo mejoras en el desempeño de los modelos, sin embargo, como se ilustra en la figura 2.5 el aumentar la cantidad de features en un dataset muchas veces es contraproducente para obtener buenos modelos predictivos. Lo anterior es conocido como “the curse of dimensionality”, o la maldición de la dimensionalidad y como solución a esto existen algunos métodos que reducen la cantidad de features.

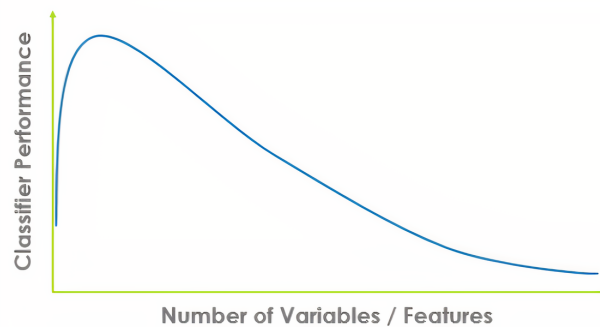


Figura 2.5: Evolución del desempeño de un modelo al aumentar la dimensionalidad del dataset. Fuente [32]

2.3.2. Métodos de reducción de dimensionalidad

Los métodos de reducción de dimensionalidad consisten en aplicar transformaciones matemáticas a los datos y así llevarlos a espacios de menor dimensión. Algunos de los métodos de reducción de dimensionalidad son:

- **Principal components analysis (PCA)**
- **Kernel principal components analysis (KPCA)**
- **Linear discriminant analysis (LDA)**
- **Locally Linear Embedding (LLE)**

Para este trabajo de título, solo se utilizará Linear discriminant analysis (LDA)

2.3.2.1. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) es una proyección lineal con la que se busca encontrar las direcciones donde se maximice la separación entre clases y se minimice la dispersión intraclase, como se nota en la figura 2.6.

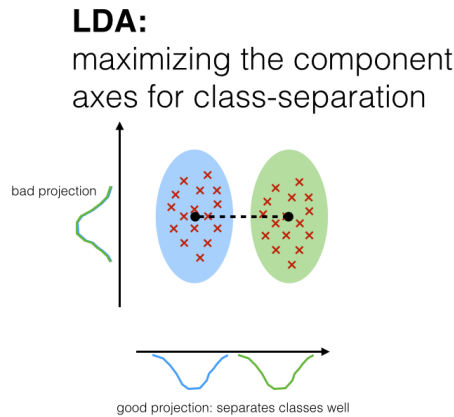


Figura 2.6: Ejemplo de LDA. Fuente [13]

Los pasos para realizar esta metodología[13] consisten principalmente en 6:

1. Se toma cada clase del dataset, con sus n-dimensiones (features).
2. Se calcula el vector promedio para cada dimensión y clase.
3. Se calculan las matrices de dispersión.
4. Se calculan los vectores y valores propios de las matrices.
5. Se eligen k vectores propios, tal que sus valores propios sean los k mayores, y se ordenan por sus valores propios de mayor a menor, con estos se forma una matriz W de $n \times k$ dimensiones.
6. Se usa W , para transformar cada dato al espacio de k-dimensiones.

2.3.3. Selección hacia adelante

Selección hacia adelante (Forward Selection), es una metodología de selección de variables, en la cual se empieza con un modelo vacío y se van agregando variable una por una, con tal de ir mejorando los resultados del modelo[14]. Si se llega a un punto en que en el modelo no mejora al agregar cualquiera de las variables restantes, estas son descartadas. Este método garantiza que las dimensiones del dataset final son iguales o menores al original.

2.4. Modelos a usar

2.4.1. Random Forests

Un árbol de decisiones es un concepto un tanto intuitivo y que se ilustra en la figura 2.7, en donde se tiene un nodo inicial(raíz) que se enlaza a un nodo de decisión, el cual genera ramificaciones, las cuales pueden llegar a un nodo terminal(hoja) o a otro nodo de decisión, el cual se vuelve a ramificar, y en el caso de la figura 2.7, cada rama lleva a un nodo terminal.

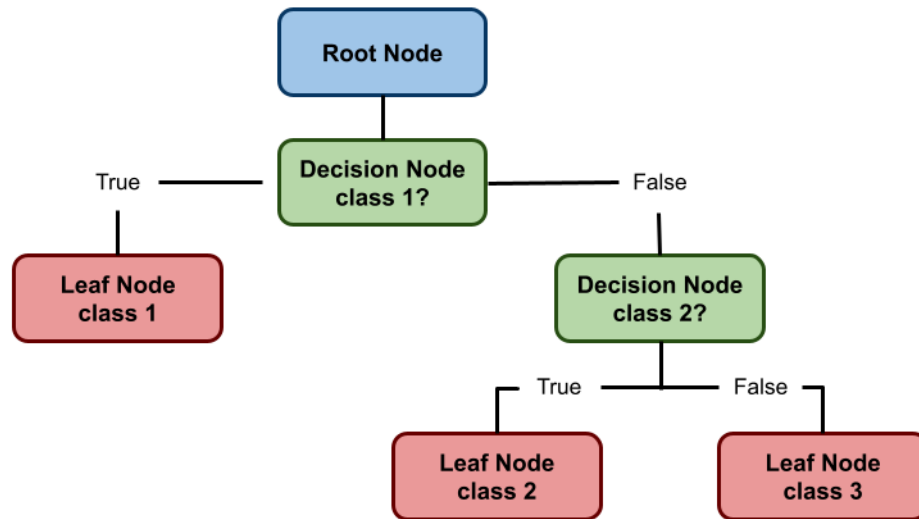


Figura 2.7: Ejemplo de un árbol de decisión. Elaboración propia.

Con tal de crear nodos de decisión precisos, existen distintos algoritmos de decisión. Pero todos funcionan mas o menos de la misma manera, primero se exploran las variables y los posibles criterios de decisión, para luego elegir aquel que posea la mejor métrica de decisión.

Las métricas de decisión mas usadas, corresponden a[16]:

- **Índice Gini**, a mayor valor de índice Gini, mejor, esta se calcula como:
 $Gini = p^2 + q^2$. Siendo p y q la probabilidad de acierto y falla, respectivamente.
- **Ganancia de la información**, a menor la entropía, mejor, esta se calcula como:
 $Entropía = -p \cdot \log_2(p) - q \cdot \log_2(q)$. Siendo p y q la probabilidad de acierto y falla, respectivamente.

Un Random Forest es un conjunto de árboles de decisión combinados con empaquetado, lo cual permite que distintos árboles vean distintas porciones de los datos, logrando que cada árbol se entrene con distintas muestras de datos para un mismo problema[15], evitando así el overfitting y proporcionando mayor flexibilidad, figura 2.8. Con esto, al combinar los resultados de cada árbol, podemos determinar el resultado del bosque completo por medio de votación. Como se ilustra en la figura 2.9.

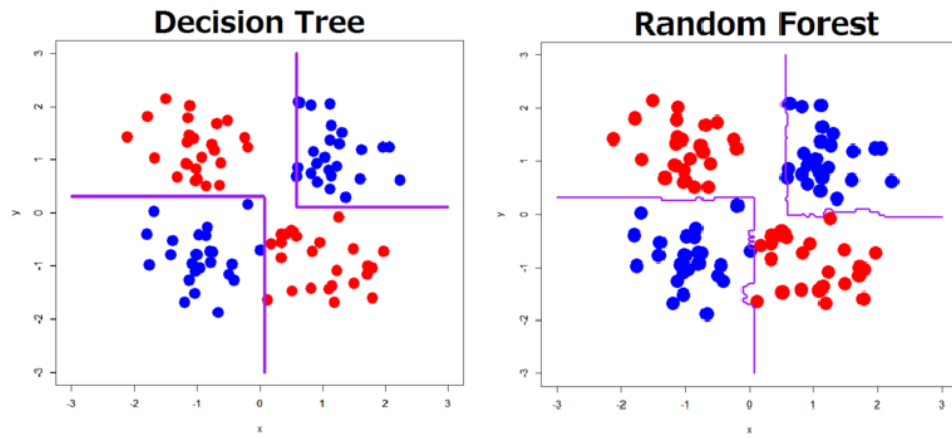


Figura 2.8: Árbol de decisión vs Random Forest. Fuente [34]

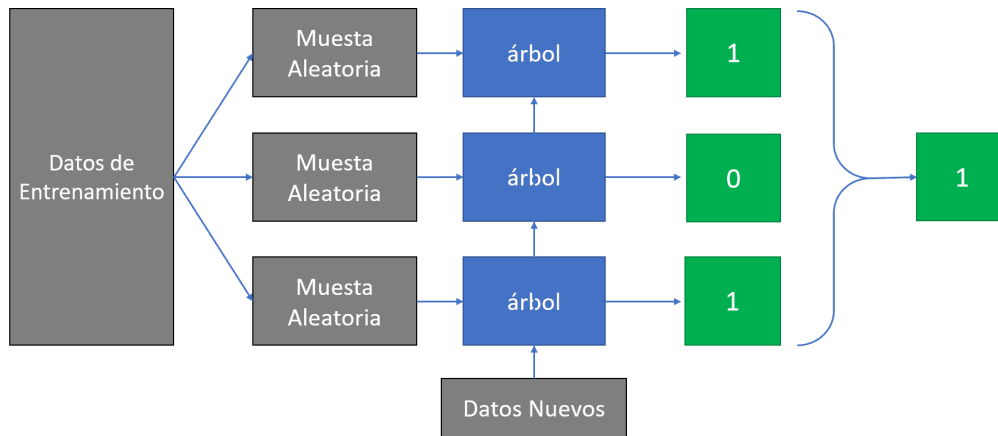


Figura 2.9: Esquema de funcionamiento de un Random Forest. Fuente [15]

2.4.2. K-Nearest Neighbors

K-Nearest Neighbors(k-vecinos más cercanos) es un modelo de clasificación, en la que se generan clusters(grupos) con los datos de entrenamiento y luego al evaluar un dato nuevo, se toman en cuenta los k elementos mas cercano, y se escoge su clasificación según cual sea la clase con más elementos en los k vecinos del nuevo dato[17], esto se puede notar en la figura 2.10.

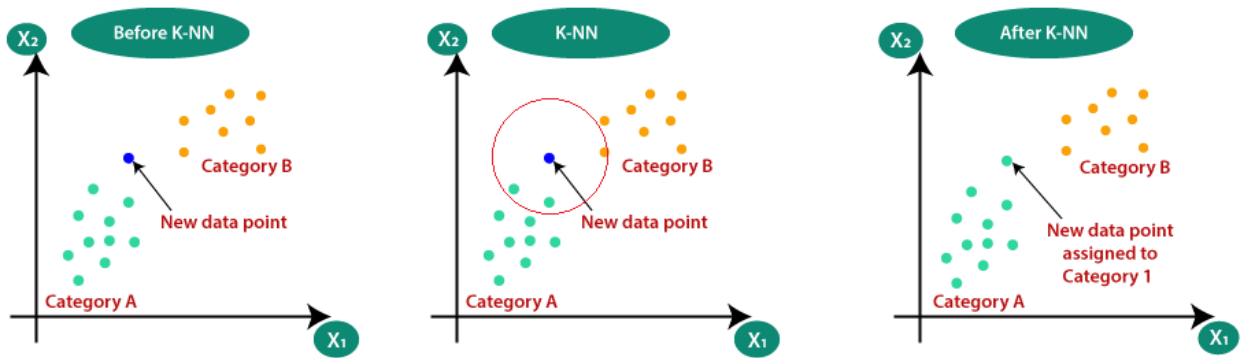


Figura 2.10: Esquema del algoritmo de clasificación KNN, con $k=3$. Fuente [17]

Por defecto, cada k elemento tiene un peso uniforme en la decisión de la clasificación, pero con tal de mejorar la robustez de las clasificaciones, se suele usar un peso basado en la distancia, el cual suele ser el inverso distancia[18], así, si un vecino esta mas cerca que otro, tiene mayor peso en la decisión de la clasificación que el otro que esta mas lejos.

2.4.3. Support Vector Machines

Support Vector Machines (SVM) es un algoritmo de que calcula el hiperplano óptimo que separa las clases de los datos en el espacio n-dimensional de los datos[19].

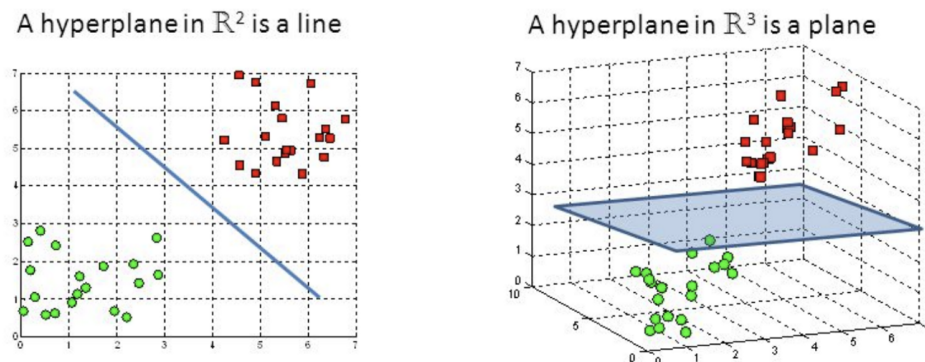


Figura 2.11: Izquierda, hiperplanos dividiendo un espacio bidimensional, derecha, hiperplano dividiendo un espacio tridimensional. Fuente [19]

Debido a que pueden existir infinitos hiperplanos que dividen a las clases, se define como al óptimo como aquel que maximice los márgenes o la separación entre los clusters de datos, tal como se ve en la figura 2.12.

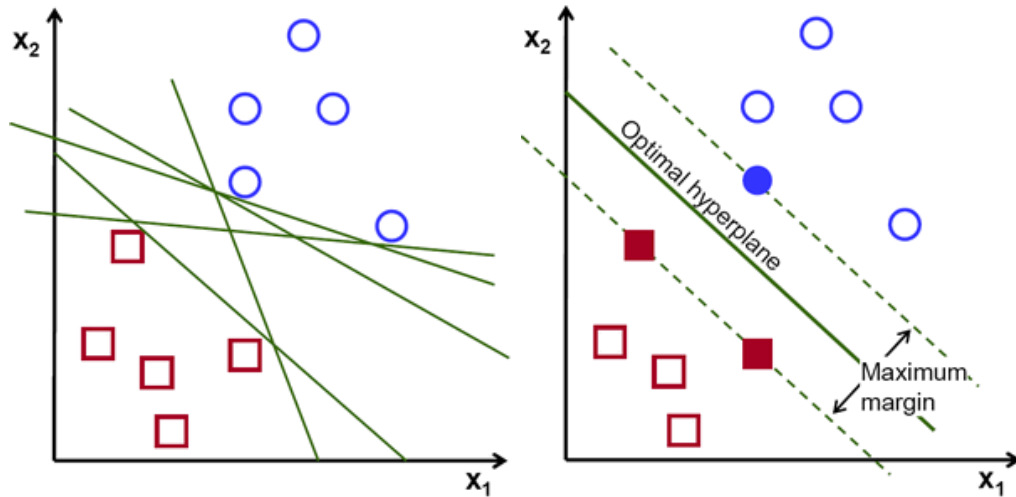


Figura 2.12: Izquierda, posibles hiperplanos, derecha, hiperplano óptimo.
Fuente [19]

Estos márgenes están relacionados con lo estrictos que estamos siendo al dividir los datos, ya que pueden existir datos que pertenezcan a una clase, pero se encuentren más cercanos a otra clase. Lo anterior se aplica usando el parámetro **Cost**[21] (**C**), el cual determina el costo por traspasar el hiperplano, si este es bajo, los hiperplanos serán mas flexibles en la división de los datos, por lo que los márgenes crecerían, y si es muy alto, el algoritmo intentará determinar todos los datos de manera correcta, y los márgenes tenderían a disminuir.

Ya que un hiperplano es una separación lineal de las clases, lo cual no es siempre posible en el espacio dimensional original, se suele aumentar la dimensionalidad, con tal de que en el espacio de mayor dimensionalidad, sea posible un separación lineal[20], figura 2.13, luego estos hiperplanos encontrados en el espacio de dimensiones mayores se proyecta al espacio original, resultando en una separación no lineal.

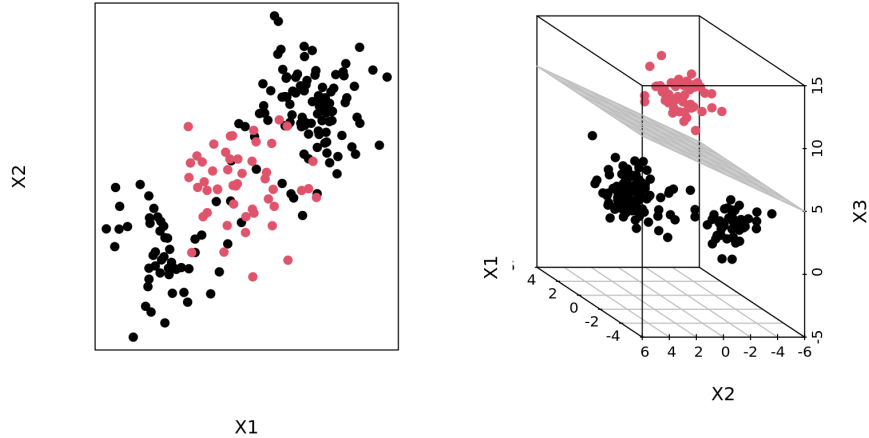


Figura 2.13: En este caso en la izquierda un hiperplano no es capaz de dividir los datos en el espacio bidimensional, pero si se agrega una dimensión, derecha, un hiperplano es capaz de separar los datos. Fuente [20]

Este aumento de dimensionalidad, la llevan a cabo los Kernels, los cuales corresponden [20][22] a una función: $K(\mathbf{x}) \in R^{>n}$, con $x \in R^n$, siendo X los elementos con su dimensionalidad original. Los mas usados corresponden a:

- Kernel lineal $K(\mathbf{x}) = \mathbf{x} \cdot \mathbf{x}'$
- Kernel polinómico $K(\mathbf{x}, \mathbf{r}, \mathbf{d}) = (\mathbf{x} \cdot \mathbf{x}' + r)^d$
 - d: Corresponde al grado del polinomio.
 - r: Corresponde al valor de la constante 0 del polinomio.

- Gaussian Kernel (RBF) $K(\mathbf{x}, \gamma) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$

γ [21]: Corresponde a la influencia que tiene cada dato del dataset, por lo que si el γ de un dato es alto, los demás datos deben estar mas cerca para afectar a la decisión del hiperplano.

- Kernel Sigmoide $K(\mathbf{x}, \gamma, r) = \tanh(\gamma \cdot \mathbf{x} \cdot \mathbf{x}' + r)$

γ : Corresponde a la influencia que tiene cada dato del dataset, por lo que si el γ de un dato es alto, los demás datos deben estar mas cerca para afectar a la decisión del hiperplano.

r: Corresponde al valor de desfase de la tangente hiperbólica.

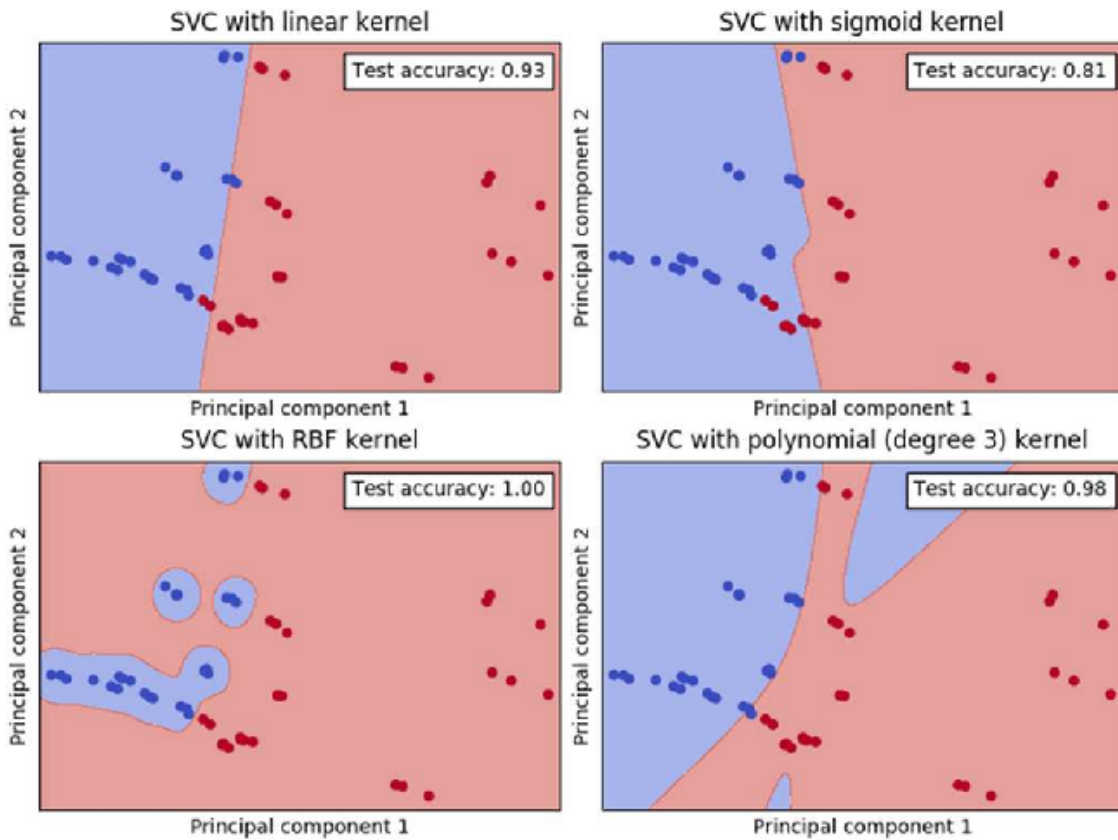


Figura 2.14: Hiperplano computado usando un Kernel Gausiano. Fuente [35]

2.4.4. Redes neuronales

El otro tipo de algoritmos que se usarán en el trabajo de título serán las redes neuronales, las cuales funcionan optimizando una función de pérdida que toma el resultado de la red neuronal y el esperado, por esto, este tipo de algoritmos son conocidos como aproximadores no lineales de funciones. Y son capaces de segmentar datos en distribuciones complejas, un ejemplo se ve en la figura 2.15.

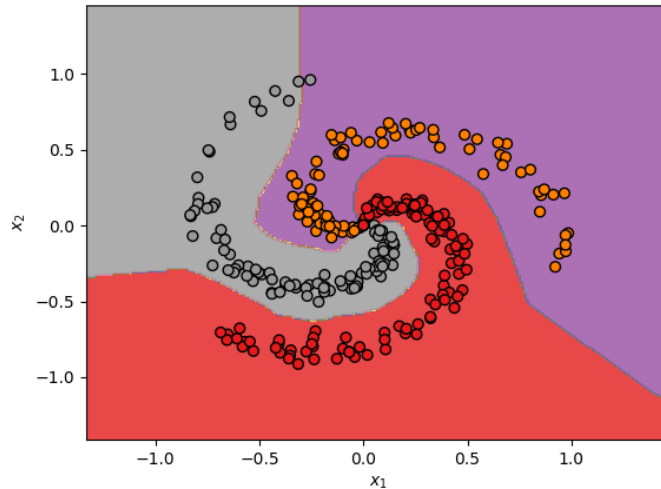


Figura 2.15: Resultados de una red neuronal. Fuente [25]

Las redes neuronales son, tal como lo indica su nombre, arquitecturas basadas en unidades llamadas neuronas artificiales. El concepto de la neurona se puede ver en la figura 2.16, a la cual ingresan varios inputs que se ponderan por su peso correspondiente, donde el output es un función de activación aplicada a la suma de los inputs ponderados.

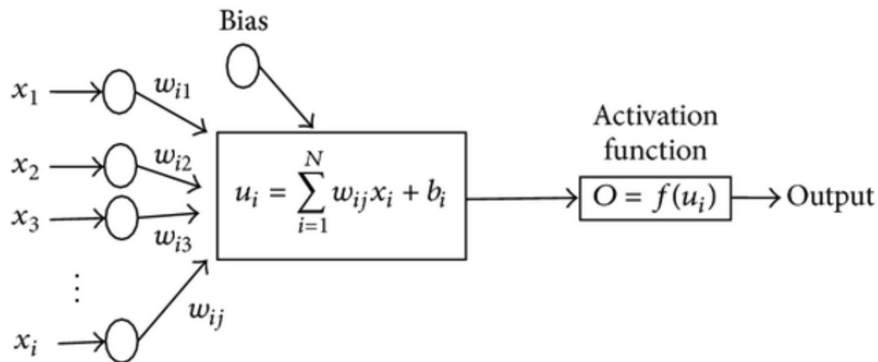


Figura 2.16: Esquema de una neurona artificial. Fuente [36]

Luego una red neuronal, se puede dividir en capas, en donde en cada capa hay una cantidad definida de neuronas, y según su orden dentro de esta red, las capas reciben el nombre[25] de:

- Capa de entrada: Siendo esta la primera, corresponde a las features de un dato del dataset, entonces si se tienen k features por datos, se tiene que esta capa tendría k dimensiones.
- Capa oculta: Son las capas que siguen a las de entrada, y están conformadas por N_i neuronas, siendo N_i un hiperparámetro, en estas cada neurona tiene sus propios pesos y bias, y función de activación, otro hiperparámetro, generando un output por cada

neurona, siendo el resultado de esta capa un vector con cada output de las neuronas y por ende de dimensión N_i .

- **Capa de salida:** Corresponde a la ultima capa, la cual toma el output de la ultima capa oculta y genera el resultado final de la red neuronal, usando nuevamente neuronas, las dimensiones de esta capa corresponden con las dimensiones del resultado que se busca. Por ejemplo, si se esta clasificando a n clases distintas de datos, esta capa tendrá n neuronas, una por cada clase.

En la figura 2.17, se puede ver una red neuronal con una capa de entrada de 3 features, los cuales pasan a la primera capa oculta de 4 neuronas, en donde cada neurona toma los datos de entrada y genera un output, luego estos cuatro output son los datos que pasan a la siguiente capa oculta, la cual posee 4 neuronas también, las cuales al igual que con la primera capa oculta, toman los datos y generan un output cada una, para finalmente pasar estos cuatro outputs a la capa de salida, en la cual se encuentra solo una neurona, la cual genera la predicción o regresión que se busca.

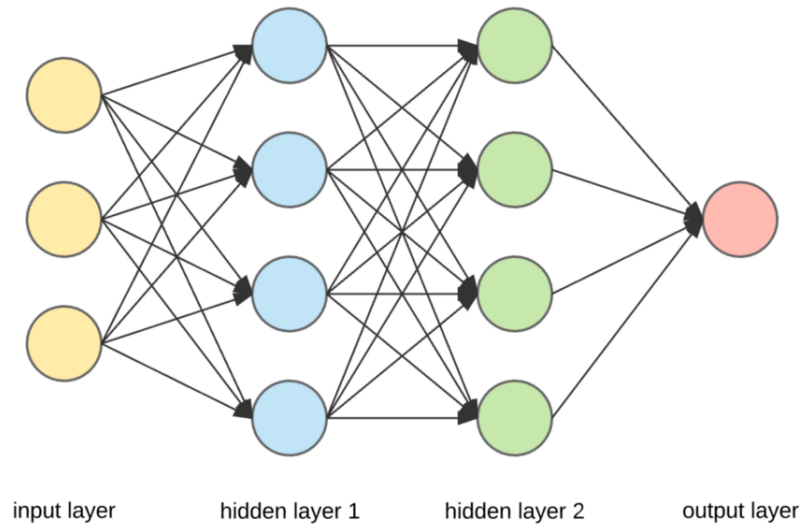


Figura 2.17: Esquema de una red neuronal. Fuente [37]

2.4.4.1. Hiperparámetros

Para una red neuronal densa, los hiperparámetros[23][24] son:

- N_i , numero de neuronas en la capa oculta i .
- **Función de activación**, son aquellas que transforman el resultado de la neurona a un rango de valores definidos, generando el output de la neurona, tal como se ve en la figura 2.16. Algunas funciones de activación[26] se pueden ver en la figura 2.18 y se describen a continuación:
 - ReLU (Rectified Linear Unit):
 $f(u) = \max(0, u)$ en donde $f : R \rightarrow R_0^+$.
 - Sigmoid:
 $f(u) = \frac{1}{1+\exp(-u)}$ en donde $f : R \rightarrow [0, 1]$

- Tanh:
 $f(u) = \tanh(u)$ en donde $f : \mathbb{R} \rightarrow [-1, 1]$

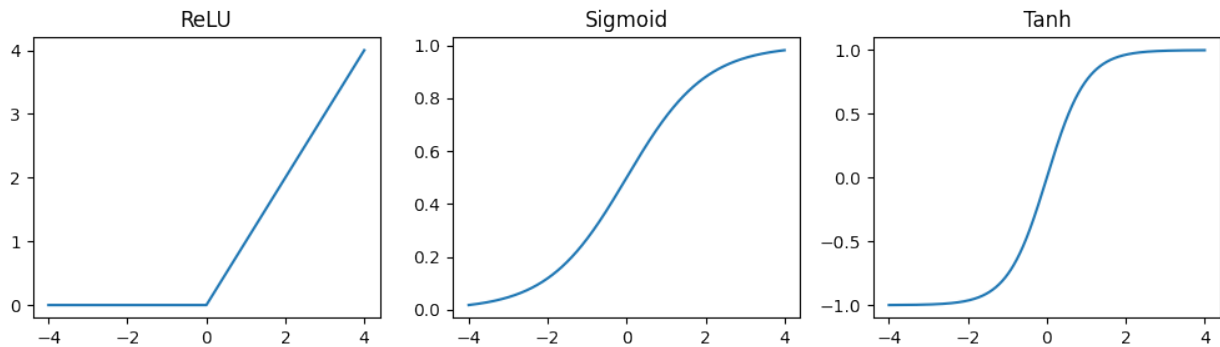


Figura 2.18: Funciones de activación. Elaboración propia

- **Optimizador** corresponde al algoritmo de actualización de los pesos y bias, y es el que permite entrenar las redes neuronales. La elección de un optimizador u otro, trae consigo la elección de hiperparámetros propios de cada optimizador. Pero un hiperparámetro común entre ellos corresponde:
 - **Learning Rate**, el cual corresponde a que tanta influencia tiene cada actualización. De este modo si este valor es muy grande, la siguiente actualización de los pesos, los terminara modificando mucho, lo cual puede evitar que el modelo converja. Algunos optimizadores, van modificando este valor a medida que van entrenando con tal de converger mas rápido.
- α , corresponde al factor de regulación el cual afecta a la función de perdida, ya que este factor determina que tan penalizante es la complejidad de los pesos para el cálculo de la pérdida. Entre mas grande, mas grande es la penalización.

Capítulo 3

Metodología

En este capítulo se detallan los procedimientos y decisiones tomadas a lo largo de la semanas de trabajo para llegar a los resultados finales. Los pasos a seguir se pueden dividir en 4 grandes etapas:

1. Recopilación de antecedentes y exploración del dataset.
2. Preprocesamiento de los datos.
3. Selección de features y de hiperparámetros.
4. Análisis de resultados.

Todos los códigos generados para cumplir las tareas descritas a continuación fueron escritos en el lenguaje de programación **python**, usando principalmente las librerías *numpy*, *pandas*, *matplotlib* y *scikit-learn*. La computación de los códigos se llevo a cabo principalmente en un computador propiedad del estudiante, aunque en ciertas ocasiones también se utilizó el servicio de cloud computing *google collab*. También se uso el sistema de control de versiones *GitHub*¹ para, valga la redundancia, mantener un control de las versiones del código y también tener respaldo de los códigos.

3.1. Recopilación de antecedentes y exploración del dataset

Durante las primeras semanas de trabajo, se procedió a resolver dudas respecto al estudio donde se generó el dataset, junto consultar trabajos relacionados a este para entender mejor el contexto del problema. Con un buen entendimiento del problema a resolver se procedió a la exploración del dataset para entender mejor el camino a seguir en la etapa del preprocesamiento.

En primer lugar se descargó el archivo `data.zip` desde el siguiente link: <https://archive.ics.uci.edu/ml/machine-learning-databases/00447/>. Al descomprimir el archivo descargado,

¹ Para ver el GitHub utilizado en el trabajo de título visitar: https://github.com/IgnacioHC/Trabajo_de_titulo

se encontraron 20 archivos en formato .txt, 2 archivos nombrados description.txt y document.txt que contenían información del dataset, 1 archivo de nombre profile.txt que contenía las etiquetas de los datos y 17 archivos correspondientes a los datos medidos por cada sensor con nombre SENSOR.txt, donde SENSOR es el nombre de cada sensor especificado en la tabla 2.3.

Al cargar en el editor de código los archivos correspondientes a los sensores, y a las etiquetas, resultaron arreglos con las siguientes dimensiones:

Tabla 3.1: Dimensiones de los archivos cargados.

Nombre archivo	Dimensiones del arreglo
profile.txt	(2205, 5)
TS1.txt	(2205, 60)
TS2.txt	(2205, 60)
TS3.txt	(2205, 60)
TS4.txt	(2205, 60)
VS1.txt	(2205, 60)
CE.txt	(2205, 60)
CP.txt	(2205, 60)
SE.txt	(2205, 60)
FS1.txt	(2205, 600)
FS2.txt	(2205, 600)
PS1.txt	(2205, 6000)
PS2.txt	(2205, 6000)
PS3.txt	(2205, 6000)
PS4.txt	(2205, 6000)
PS5.txt	(2205, 6000)
PS6.txt	(2205, 6000)
EPS1.txt	(2205, 6000)

Con respecto a los archivos correspondientes a los sensores , las dimensiones corresponden a **(ciclos, datos por ciclo)**, donde los valores coinciden con lo esperado según las tasas de muestreo de la tabla 2.3. En cuanto al archivo profile.txt, sus dimensiones corresponden a **(ciclos, clasificación)**, donde clasificación corresponde a las 5 clasificaciones de estados de salud descritas en la tabla 2.2. Es importante mencionar que en base a esto notamos que el estado de salud o **la clase de cada componente es constante a lo largo de cada ciclo de proceso**, y esta solo cambia al variar las condiciones de cada ciclo. También cabe destacar que los 60, 600 o 6000 datos que entregue cada sensor por 1 ciclo de operación, corresponden simultáneamente a 1 clase por cada una de las 5 clasificaciones, es decir **a un ciclo de cada sensor le corresponden 5 etiquetas distintas**.

El siguiente paso fue poder visualizar los datos sin procesar para poder tener una mejor idea del comportamiento de estos. Debido a que el dataset contiene datos de repeticiones de un mismo ciclo, pero en diferentes condiciones, se decidió visualizar los comportamientos de

las distintas variables de proceso **a lo largo de 1 ciclo**. Para lograr lo anterior se llevaron a cabo los siguientes pasos:

1. Se elige 1 clasificación (válvula, enfriador, etc.).
2. A partir de esta clasificación, por cada estado de salud (o clase) que le corresponda, se selecciona de forma aleatoria 1 ciclo correspondiente a dicha clase (usando las etiquetas de los datos).
3. Se seleccionan los datos correspondientes a los 17 sensores o un subconjunto de ellos.
4. Para cada sensor en el conjunto escogido, se selecciona los datos de los ciclos correspondientes a los índices obtenidos anteriormente.
5. Se crea una figura con un gráfico por cada sensor escogido, donde cada gráfico presenta N curvas a lo largo de los 60 segundos de duración de un ciclo, con N igual a la cantidad de clases distintas que tenga el componente elegido.

Por ejemplo, se desea revisar el comportamiento de los sensores de temperatura al intentar identificar los estados de salud del estado del enfriador, para lo cual selecciono lo siguiente:

- **Sensores:** TS1, TS2, TS3, TS4.
- **Clasificación:** Estado del enfriador.

Con esto, la función programada buscará un índice de forma aleatoria para cada clase resultando:

- Cerca de la falla total: índice clase 1.
- Eficiencia reducida: índice clase 2.
- Eficiencia total: índice clase 3.

Y finalmente se crea una figura que contendrá 4 gráficos (uno para cada sensor) y 3 curvas por cada gráfico correspondientes a las 3 clases del estado del enfriador.

Con los gráficos descritos anteriormente se pudo revisar el comportamiento de las diferentes variables de proceso y como estas se comportaban en función de los diferentes estados de salud de los componentes y el sistema.

3.2. Preprocesamiento de los datos

En esta etapa se detallan los procedimientos seguidos para preparar los datos antes de ser usados con los modelos de ML, desde que se cargan los datos hasta obtener los conjuntos de entrenamiento y testeo. El primer cambio aplicado a los datos originales fue separar el arreglo con las etiquetas a 5 arreglos de dimensión (2205,) correspondientes a las 5 clasificaciones, para después realizar los procesos que se explican a continuación:

3.2.1. Separación en conjuntos de entrenamiento y testeo

Para entrenar un modelo de ML y luego evaluar su desempeño, es necesario separar los datos y las etiquetas en conjunto de entrenamiento y conjunto testeo. En el caso de este trabajo de título, y como es usual al trabajar con modelos de ML, se separaron los datos usando la función de la librería scikit-learn `train_test_split`[5]. Esta función toma una determinada proporción de datos de entrenamiento para dividir los conjuntos tomando datos y etiquetas de forma aleatoria y finalmente reordenarlos. La separación de los datos se llevo a cabo por sensor, y los parámetros usados fueron los siguientes:

Tabla 3.2: Parámetros usados para la función `train_test_split`

Nombre parámetro	Parámetro	Tipo de dato
X	datos sensor	array (2205, 60*tasa de muestreo)
y	etiquetas clasificación	array (2205,)
<code>train_size</code>	0.7	float
<code>random_state</code>	19	int

El parámetro `train_size` le entrega al modelo la proporción del conjunto de entrenamiento (70% en este caso) con respecto al total de datos y el parámetro `random_state` genera la combinación de datos que serán tomados pseudo aleatoriamente para que esta sea repetible. Un aspecto importante a tener en consideración es la decisión de dividir los datos en conjuntos de entrenamiento y testeo antes de aplicar los parámetros de tiempo, lo que se hizo para evitar que datos de parámetros de tiempo extraídos de un mismo ciclo vayan a parar a ambos conjuntos de datos (entrenamiento y testeo), o dicho más simple, evitar que los modelos se entrenen con datos que luego evaluaran su desempeño.

3.2.2. Obtención de parámetros de tiempo

Como ya se ha adelantado, a partir de los datos sin procesar se extrajeron parámetros de tiempo sobre ventanas de igual duración en tiempo, pero diferente cantidad de datos según las tasas de muestreo de cada sensor. Los parámetros de tiempo usados fueron: **RMS**, **media**, **valor peak to peak** y **varianza**, cuyas formulas matemáticas fueron detalladas en el capítulo de antecedentes. Con el fin de estudiar el efecto en el desempeño de los modelos, se probaron distintas combinaciones de largo de ventana y overlap sobre las cuales se aplicaron los parámetros de tiempo. En la tabla 3.3 se detallan estas combinaciones de largo de ventanas y largos de overlap, junto con la cantidad de datos que se generan al aplicar un parámetro de tiempo sobre la ventana de tiempo segun la siguiente ecuación:

$$Datos\ por\ ciclo = ParteEntera \left(\frac{Duración\ ciclo - Largo\ overlap}{Largo\ ventana - Largo\ overlap} \right)$$

- **Datos por ciclo** : Cantidad de valores de un parámetro de tiempo que se extraen por ciclo
- **Duración ciclo = 60 segundos** : Duración total del ciclo de operación.
- **Largo ventana [segundos]** : Largo de la ventana de tiempo sobre la cual se aplica el parámetro de tiempo.
- **Largo overlap [segundos]** : Largo de tiempo que comparten las ventanas de un mismo ciclo.

Tabla 3.3: Combinaciones de largos de ventana y overlap usados para generar los conjuntos de entrenamiento y testeo.

Largo ventana [segundos]	Largo overlap [segundos]	Datos por ciclo extraídos por el parámetro de tiempo
60	0	1
30	0	2
20	0	3
22	10	4
20	10	5
18	10	6
15	8	7

3.2.3. Escalamiento de los datos

Otra etapa necesaria dentro del preprocesamiento es escalar los datos como se explicó en el capítulo de antecedentes, para esto se usó la función de scikit-learn `MinMaxScaler`[6], usando los parámetros por defecto, es decir se escalaron los datos entre 0 y 1. Este proceso se llevó a cabo para los datos de cada sensor y además, se cercioró de que el *scaler* se entrenara con todos los datos de cada sensor.

3.2.4. Cambio en las dimensiones de las etiquetas

Una vez que se realiza el proceso de extraer parámetros de tiempo, los datos de todos los sensores pasan a tener dimensiones de (2205, datos extraídos por ciclo) mientras que las dimensiones de los 5 arreglos de las etiquetas se mantuvieron en (2205,), por lo cual fue necesario crear una función que redimensionara las etiquetas a (2205, datos extraídos por ciclo) para hacerlas compatibles con los datos de los sensores.

3.3. Selección de features y de hiperparámetros

3.3.1. Selección de features

Como se explicó anteriormente, usar todos o una gran cantidad de features (sensores en este caso) muy probablemente reducirá el desempeño de los modelos, por lo cual se usó el

método de **selección hacia adelante** para determinar los mejores subconjuntos de sensores a usar en cada caso. Este proceso de selección fue bastante extenso, ya que para cada modelo escogido, era necesario entrenar varios subconjuntos de sensores, durante varias iteraciones para escoger al mejor subconjunto tal como se explicó en el subcapítulo 2.3.3.

En la tabla 3.4 se ven los 18 modelos diferentes usados para realizar la selección, los cuales se probaron usando datos provenientes de los 4 parámetros de tiempo, para cada uno de los 7 largos de ventana y para cada una de las 5 clasificaciones, resultando un total de $5 \cdot 7 \cdot 4 \cdot 18 = 2520$ combinaciones distintas de clasificación - parámetro de tiempo - largo de ventana - modelo. Una vez terminadas las iteraciones de la selección hacia adelante, el accuracy obtenido por la combinación, junto con el conjunto de sensores seleccionados se guardaron en archivos .csv y .json respectivamente para su posterior uso. Inicialmente todos los modelos se probaron variando solo 1 parámetro dejando los demás por defecto, salvo por el MLPClassifier, cuyos parámetros se especifican en la tabla 3.5

NOTA: Los modelos MLPClassifier corresponden a los modelos basados en redes neuronales.

Tabla 3.4: Modelos usados para realizar la selección de sensores.

Nombre modelo	Parámetro usado
RandomForestClassifier[7]	n_estimators = 40
	n_estimators = 60
	n_estimators = 80
	n_estimators = 100
	n_estimators = 120
KNeighborsClassifier[8]	n_neighbors = 1
	n_neighbors = 2
	n_neighbors = 3
	n_neighbors = 4
	n_neighbors = 5
	n_neighbors = 6
SVC[9]	kernel = 'rbf'
	kernel = 'linear'
	kernel = 'sigmoid'
LinearDiscriminantAnalysis[10]	n_components = 1
MLPClassifier[11]	activation = 'tanh'
	activation = 'relu'
	activation = 'logistic'

Tabla 3.5: Parámetros usados en la selección de sensores para los modelos MLPClassifier.

Nombre parámetro	Parámetro
hidden_layer_sizes	(20, 20, 20, 20, 20)
learning_rate	'constant'
early_stopping	True
max_iter	1200

3.3.2. Selección de parámetros

Finalmente, se utilizaron los conjuntos de sensores con mejor desempeño asociados a las distintas combinaciones para variar algunos parámetros de los modelos. Particularmente para el modelo correspondiente a LDA no se realizó una variación de sus parámetros, guardando solo los resultados de la selección de features.

Random Forest: Para los modelos RF se probó su desempeño al cambiar su criterio de gini a entropía de la información.

K Nearest Neighbors Para los modelos KNN se varió el peso de los vecinos, pasando de un peso uniforme a un peso basado en la distancia.

Support Vector Classifier: En el caso de los modelos SVC, se buscó el mejor kernel y la mejor cantidad de datos por ciclos para ser fijados en los modelos y luego variar los parámetros “C” y “gamma”, tomando los siguientes valores:

- $gamma = 10, 100, 200, 500, 1000, 1500, 2000$ y 3000
- $C = 10, 10^2, 10^3, 10^4, 10^5, 10^6$ y 10^7

Multi Layer Perceptron Classifier: El caso del MLPClassifier fue bastante similar al SVC, ya que se estudio en primer lugar el accuracy para diferentes largos de ventana y para diferentes funciones de activación, para luego tomar las mejores combinaciones por cada clasificacion y variar el learning rate y el alpha como sigue:

- $alpha = 0.0001, 0.001, 0.01$ y 0.1
- $learningrate = 0.0001, 0.001, 0.01$ y 0.1

3.4. Análisis de resultados

El último gran paso realizado durante el trabajo de título, consistió en determinar la mejor forma de representar gráficamente los resultados obtenidos junto con los criterios para comparar los diferentes desempeños obtenidos. Los criterios que se usaron finalmente para analizar, son descritos a continuación y tienen relación con estudiar el cambio en el desempeño de los diferentes modelos al variar las combinaciones descritas en la subsección 3.3.1. Se destaca nuevamente que para medir el desempeño de los diferentes modelos, solo se usó el accuracy, por lo cual los análisis descritos a continuación se enfocan en estudio de la variación de dicha métrica.

1. **Efecto del largo de las ventanas de tiempo:** Este efecto se estudió en los resultados finales del modelo, donde se graficó la evolución del accuracy en función del largo de ventana y del parámetro principal de cada modelo.
2. **Diferencias entre parámetros de tiempo:** En cuanto a los parámetros de tiempo, en primer lugar, se graficaron los resultados obtenidos a partir de gran parte de los sensores, usando las diferentes ventanas de tiempo descritas en la tabla 3.3, con el fin de notar diferencias entre los sensores y entre los largos de ventana, y así tener una idea a priori de los resultados a futuro. Finalmente, en las figuras de los resultados de todos los modelos, los resultados se dividieron en diferentes gráficos correspondientes a los diferentes parámetros de tiempo
3. **Efecto del cambio de los parámetros de los modelos:** Se estudió la evolución del desempeño de cada modelo a medida que se iban variando los parámetros descritos en la tabla 3.4. Para todos los modelos se analiza inicialmente el cambio del accuracy en función de un parámetro principal, y para los casos del SVM y MLP se analizan adicionalmente los cambios al variar otros 2 parámetros.
4. **Diferencias entre los sensores:** Para los diferentes tipos de sensores mencionados en la tabla 2.3, se analizaron los gráficos con la data raw descritos en la sección 3.1, buscando a simple vista para tener una idea a priori acerca de qué variables de operación y qué clasificaciones entregarían los mejores resultados. Tras el preprocesamiento cada combinación de modelo - ventana-parámetro de tiempo - clasificación eligió un set de sensores diferente en la selección hacia adelante, lo que no permitió poder analizar el accuracy obtenido usando diferentes combinaciones de sensores.
5. **Resultados generales:** Finalmente se compararon los resultados obtenidos por el estudiante con los resultados obtenidos por el estudio original. Se compararon los mejores desempeños por clasificación y por modelo.

Capítulo 4

Resultados

En este capítulo se muestran de forma gráfica los resultados obtenidos por las diferentes combinaciones de modelos, sensores, parámetros de tiempo, ventanas de tiempo e hiperparámetros de los modelos.

Se debe tener en cuenta que debido a la gran cantidad de combinaciones distintas **NO es posible presentar todos los resultados** obtenidos, sin embargo, se seleccionaron cuidadosamente las figuras para entregar la mayor cantidad de información posible. Para consultar los gráficos de más las combinaciones descritas en la metodología, revisar el anexo A.

También es importante recordar para los subcapítulos 4.1 y 4.2 que los datos de un sensor ya sea TS1, PS6, FS2, o cualquier otro, presentan simultáneamente 5 estados de salud distintos, por lo cual para un mismo sensor variando la clasificación o el parámetro de tiempo, es probable encontrar comportamientos muy similares.

4.1. Gráficos de los datos sin procesar

Los primeros resultados a mostrar, son los gráficos de las variables de operación captadas por los diferentes sensores sin ningún tipo de preprocesamiento. Como se detalló en el capítulo de metodología, en cada figura se usan datos provenientes de alguna de las clasificaciones (componentes o sistema), donde cada gráfico representa los valores obtenidos por un determinado sensor a lo largo de los 60 segundos de operación de 1 ciclo. En cuanto a las diferentes curvas que se aprecian en cada gráfico, estas corresponden a las clases o estados de salud que puede presentar cada componente que está siendo monitoreado. Cabe destacar que cada curva representa solo 1 ciclo de operación **elegido aleatoriamente** de un total de 2205 ciclos medidos. Particularmente para este tipo de gráficos, al ser obtenidos de forma aleatoria, se decidió obtener 5 iteraciones de cada uno y mostrar en este capítulo las figuras mas representativas.

4.1.1. Gráficos de la temperatura

Los sensores de temperatura (TS1, TS2, TS3 y TS4), representados en las figuras 4.1 a 4.7 entregan valores en $^{\circ}C$, fueron muestreados a $1[Hz]$ y según el layout del sistema

hidráulico (ver figura 1.1), los dos primeros se encuentran en el sistema primario, mientras que los dos segundos en el sistema secundario. Sin importar el tipo de clasificación, estos sensores demuestran ser sumamente útiles para obtener datos que sirvan para el monitoreo del sistema de condición, ya que sin ningún tipo de procesamiento se puede ver como las curvas de cada clase se encuentran muy bien separadas. También se destaca que independiente de las ubicaciones tan distintas que presenta cada sensor en el sistema hidráulico, a simple vista no se notan grandes diferencias en el comportamiento de ninguno de ellos.

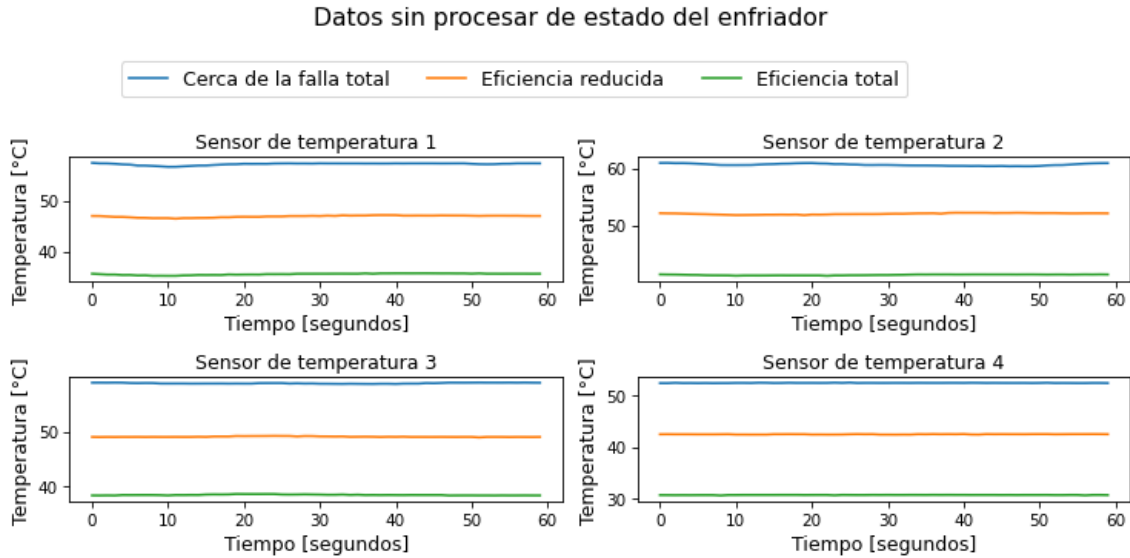


Figura 4.1: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado enfriador.

Con respecto a los gráficos del **enfriador** (figura 4.1), notamos que a simple vista parecen ser los que mayor diferencia presentan entre los valores que toma la temperatura para las diferentes clases. Se espera que los modelos que usen los datos de sensores de temperatura para predecir el estado de salud del enfriador presenten los mejores desempeños.

Para el caso de la **válvula**, se obtuvieron resultados bastante distintos a lo largo de las 5 iteraciones. Se tiene por ejemplo el caso de los gráficos obtenidos en la iteración 4 (figura 4.2) donde se distingue muy bien las diferencia entre las curvas, y por otra parte la iteración 2 (figura 4.3), donde se solapan 2 curvas por un lado y 2 por otro. A priori no podemos decir que la clasificación de la válvula tendrá un buen o un mal desempeño usando estos sensores, pero si podemos suponer que su accuracy será menor que para la clasificación del estado del enfriador.

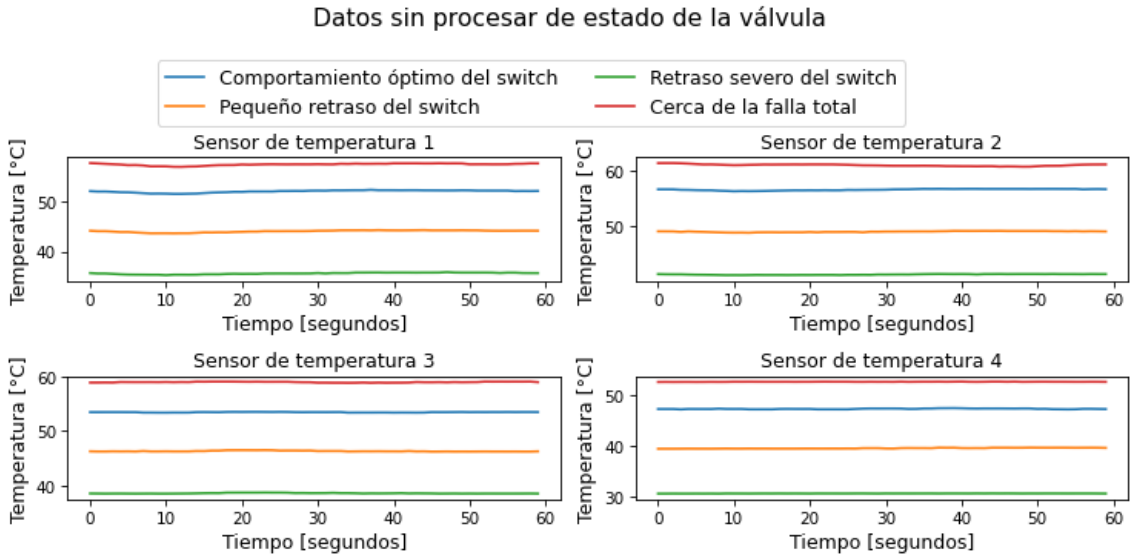


Figura 4.2: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 4.

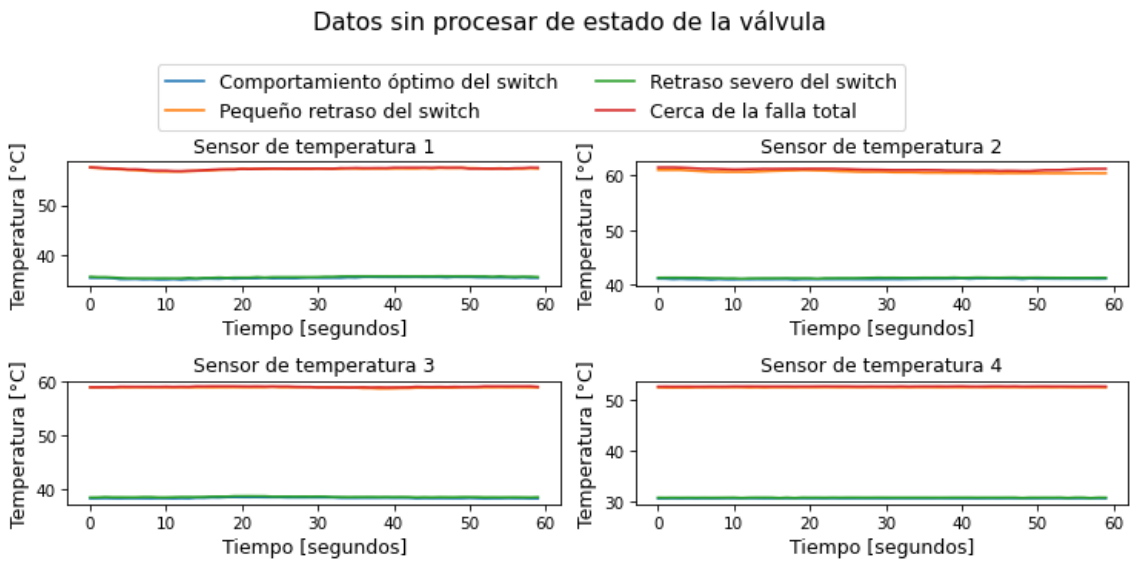


Figura 4.3: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 2.

Para el caso de la **bomba** se obtuvieron gráficos bastante buenos, donde en general se distinguían bien las curvas de estados de salud. En la figura 4.4 se puede ver como las figuras presentan una separación muy buena, similar a la presentada en los gráficos del estado del enfriador. Si bien en la mayoría de los casos las curvas de la válvula se separan bien, también hay casos donde estas curvas se juntan bastante como se puede apreciar en la figura 4.5

Datos sin procesar de fuga en la bomba

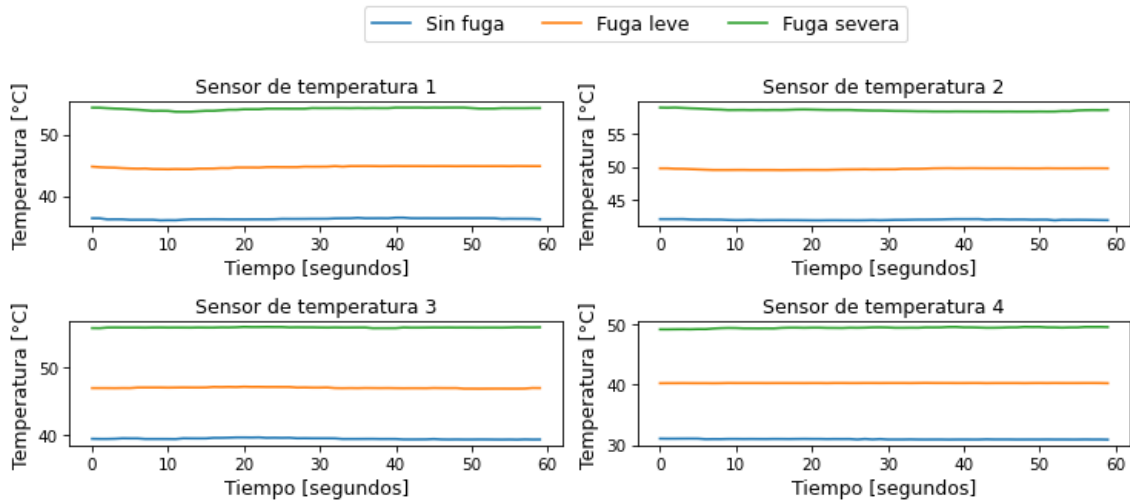


Figura 4.4: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba.

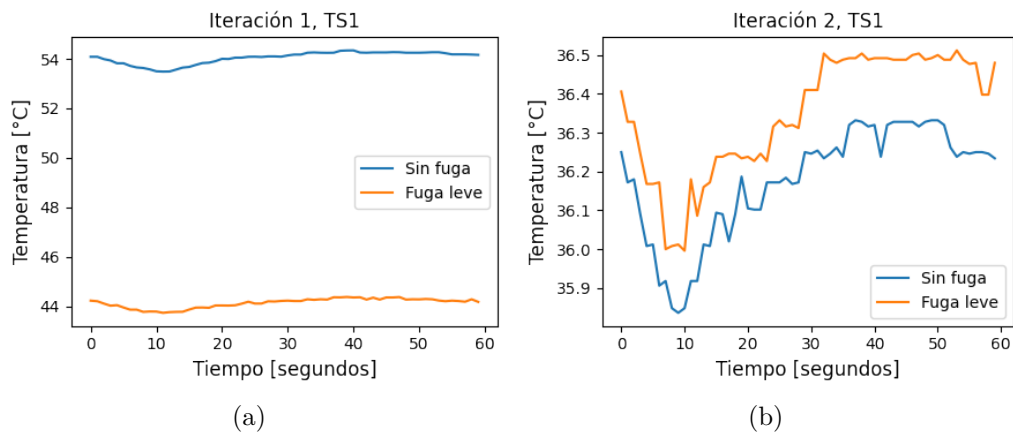


Figura 4.5: Comparación entre las curvas de los estados “sin fuga” y “fuga leve” en distintas iteraciones.

El caso del **acumulador** es similar al de la válvula, si bien es posible distinguir las curvas de sus estados de salud (ver figura 4.6), estas tienden a juntarse, lo cual probablemente entregue desempeños más bajos en comparación al enfriador o la bomba.

Datos sin procesar de estado del acumulador

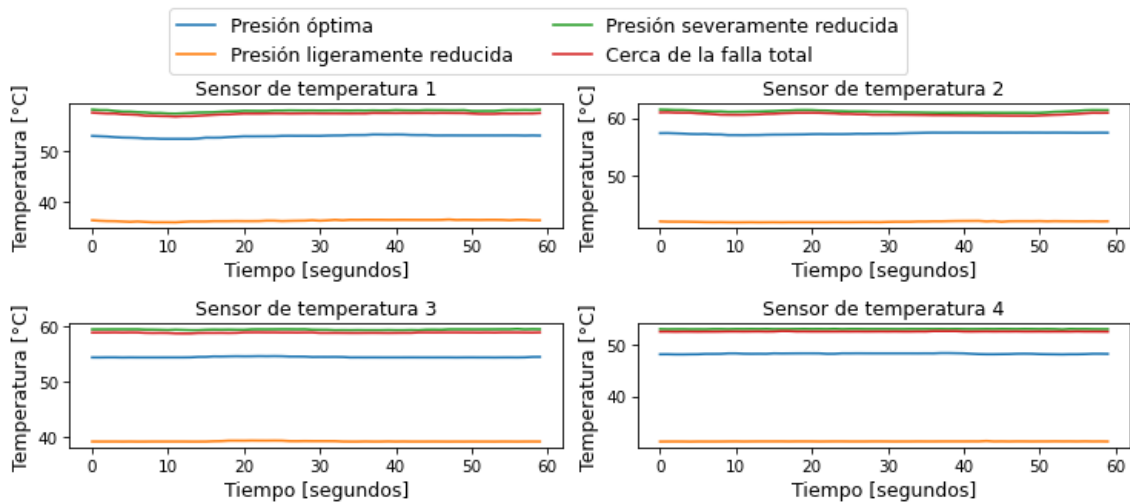


Figura 4.6: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador.

Finalmente, para el caso de la **estabilidad del sistema**, como se puede apreciar en la figura 4.7, sus curvas se distinguen a simple vista, siendo uno de los mejores casos junto con la clasificación del estado del enfriador.

Datos sin procesar de estabilidad del sistema

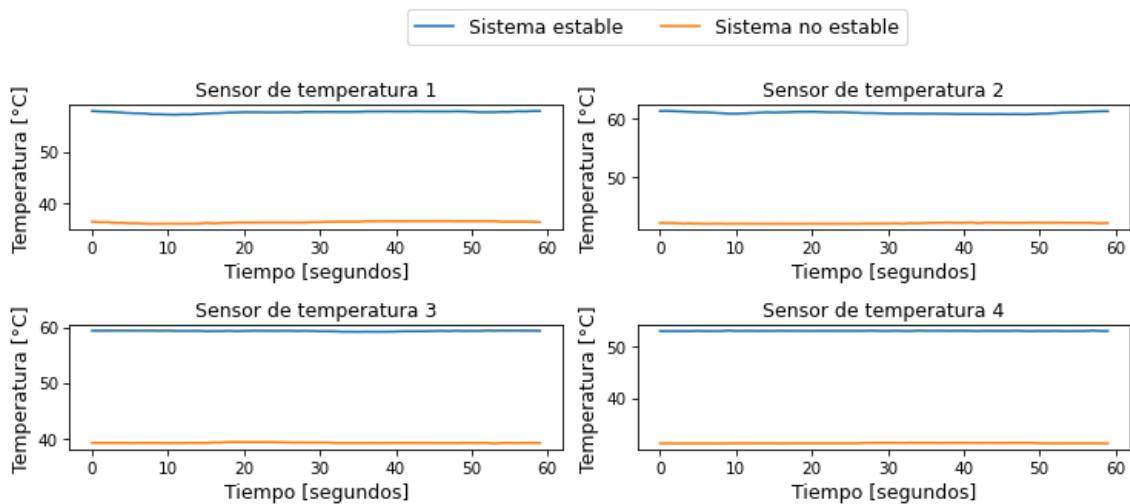


Figura 4.7: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador.

4.1.2. Gráficos de Flujo

Los sensores FS1 y FS2, usaron una tasa de muestreo de $10[Hz]$, y midieron el flujo en $[L/min]$. El sensor FS1 se ubica en el sistema primario, mientras que el sensor de flujo FS2 se ubica en el sistema secundario. De forma general, con respecto a las figuras 4.8 a 4.11 podemos decir que para el caso de FS1, no se distingue a simple vista las diferencias entre clases por

lo cual queda como candidato para ser descartado más adelante o bien de ser utilizado, es probable que los modelos le otorguen poco peso. Para los curvas correspondientes al sensor FS2, la situación es un poco mejor, ya que en ciertos ciclos es posible distinguir algunos comportamientos de las diferentes clases, sin embargo es notoria la diferencia con respecto a las curvas obtenidas de los sensores de temperatura.

Al observar la figura 4.8 con clases correspondientes al **enfriador**, se puede notar que en el gráfico correspondiente a FS1 las clases no se distinguen mucho entre si, sin embargo hay una leve mejora en un tramo donde al menos se distingue la curva correspondiente a la clase “cerca de la falla total”. Para el gráfico del sensor FS2, podemos decir que a priori es el que mejor separa los tres estados de salud del enfriador.

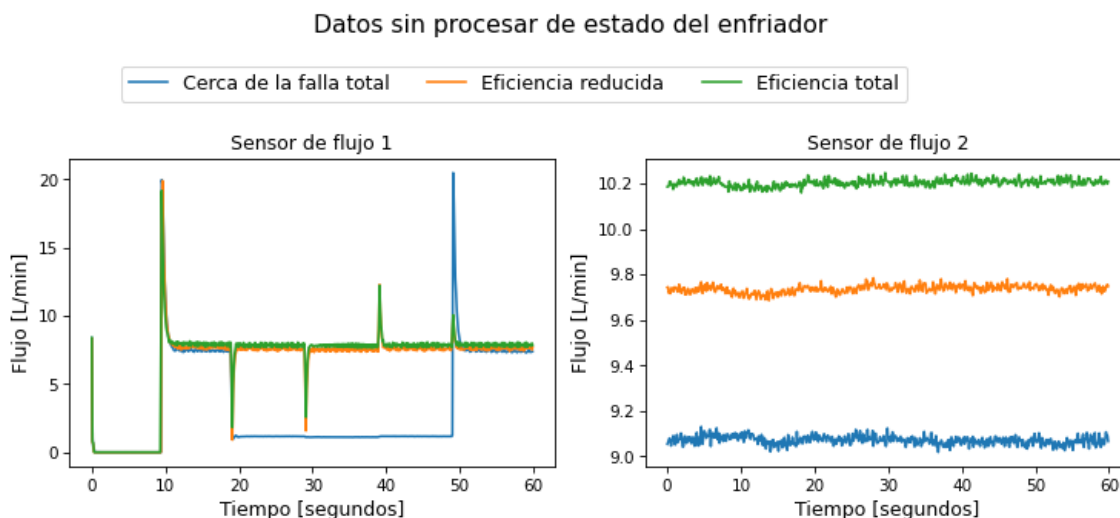


Figura 4.8: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador.

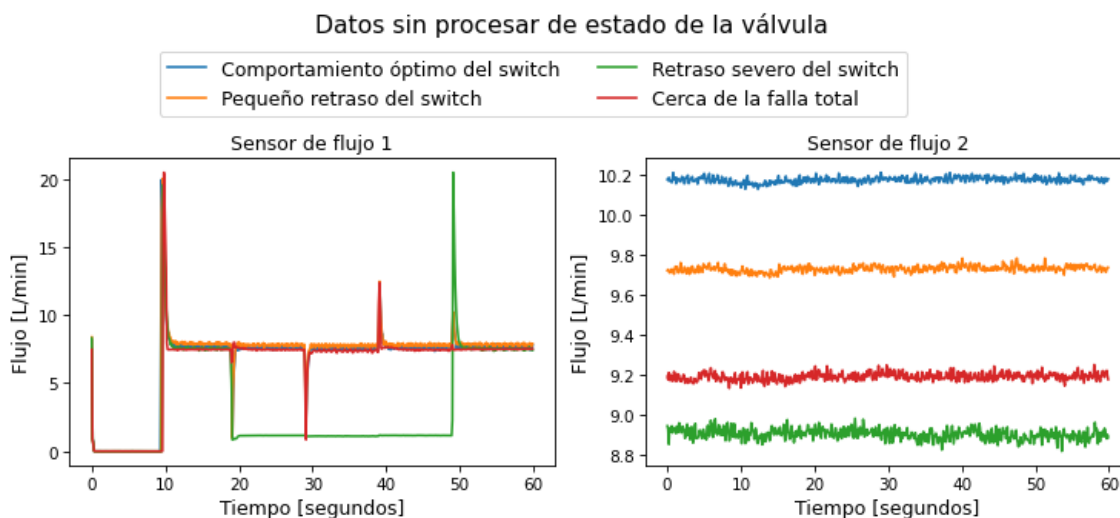


Figura 4.9: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula.

En la figura 4.9 que representa los estados de salud de la **válvula**, las curvas obtenidas a partir del sensor FS1 son prácticamente indistinguibles entre sí, salvo por un tramo donde se separa el estado “retraso severo del switch”. Para el gráfico del sensor FS2, se puede ver que se distinguen bien las clases, sin embargo, esto no fue algo habitual a lo largo de las iteraciones.

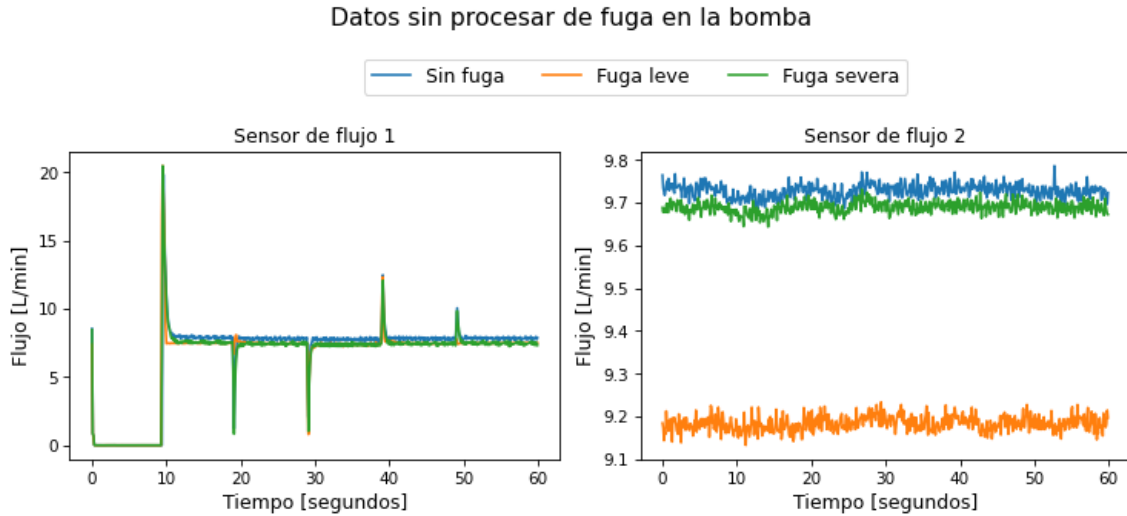


Figura 4.10: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba.

Con respecto a la figura 4.10 correspondiente a la fuga en la **bomba**, se puede ver nuevamente que para el sensor FS1 no es posible distinguir entre los diferentes estados de salud. Para el sensor FS2 notamos que 2 curvas se acercan bastante y otra se aleja, lo que podría inducir a “confundir” al modelo.

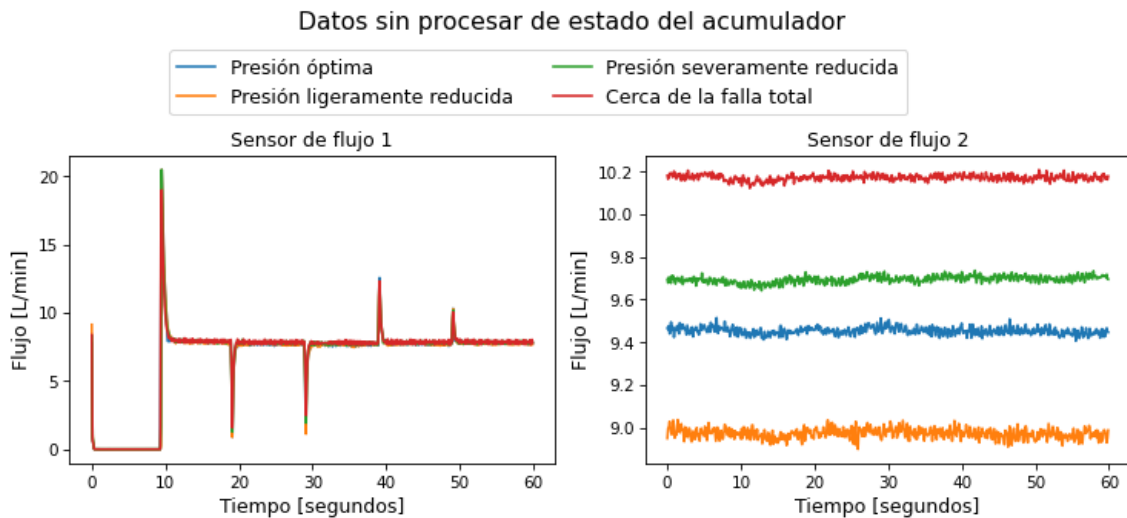


Figura 4.11: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador.

Continuando con los gráficos de los sensores de flujo, se tiene la figura 4.11 con la información correspondiente al estado del **acumulador**. Al igual que en el caso anterior, en el gráfico correspondiente al FS1 no es posible distinguir ninguna clase a simple vista, sin embargo, para el caso del sensor FS2 se pueden distinguir muy bien las clases.

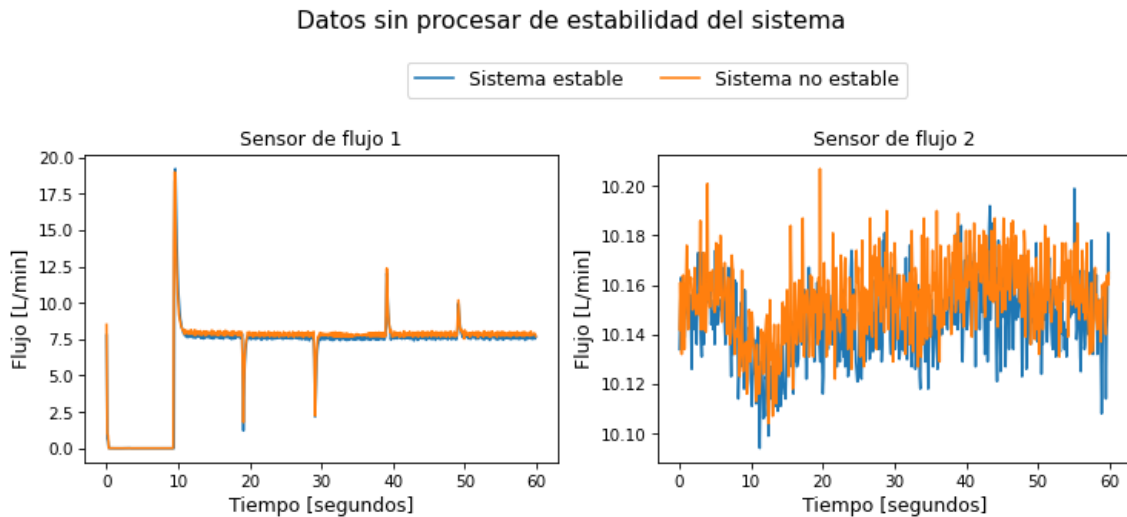


Figura 4.12: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema.

Con respecto a la figura 4.12, nuevamente el sensor FS1 presenta malos resultados, sin embargo, contrario a la mayoría de los gráficos anteriores, aquí el sensor FS2 entregó datos de estados de salud que están sumamente solapados.

4.1.3. Gráficos de Presión

Los gráficos de presión fueron creados a partir de los datos obtenidos mediante los sensores PS1, PS2, PS3, PS4, PS5 y PS6, estos usaron una tasa de muestreo de $100[Hz]$ y midieron la presión en $[bar]$. Los sensores PS1, PS2 y PS3 se encuentran en el sistema primario y por su parte los sensores PS4, PS5 y PS6 en el sistema secundario. Para el caso de los gráficos realizados a partir de los datos obtenidos de los sensores de presión (figura 4.13 a 4.18) podemos decir que en general respecto a los gráficos anteriores, estos entregan resultados mas variados de acuerdo a las diferentes clasificaciones y de acuerdo a los mismo sensores.

Para la figura 4.13 con los estados de salud del **enfriador**, tenemos que en los sensores PS5 y PS6, nos permiten distinguir muy bien entre clases, el PS1 y el PS4 permiten distinguir un poco pero no de gran forma, y de los sensores PS2 y PS3 prácticamente es imposible saber a simple vista a que clase pertenece cada curva.

Datos sin procesar de estado del enfriador

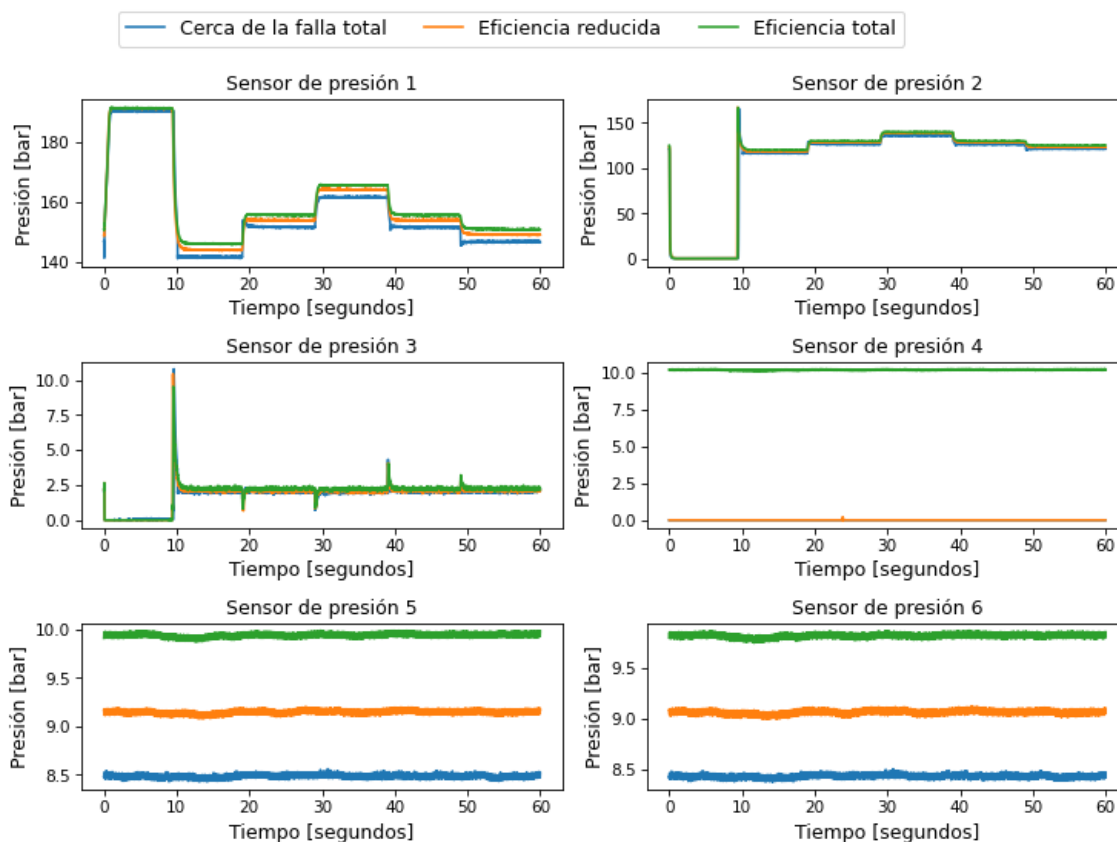


Figura 4.13: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador.

Para el caso de la **válvula**, se presentan comportamientos un tanto distintos (ver figura 4.14), aquí la mayoría de los sensores consiguen separar al menos las clases “retraso severo del switch” y “cerca de la falla total”, sin embargo, salvo por el sensor PS4, las clases mencionadas se sobreponen a las otras 2 clases. En la figura 4.15, también correspondiente a la válvula, se puede ver como nuevamente todos los sensores consiguen separar en cierto grado las diferentes clases, pero se repite la sobreposición entre clases.

Datos sin procesar de estado de la válvula

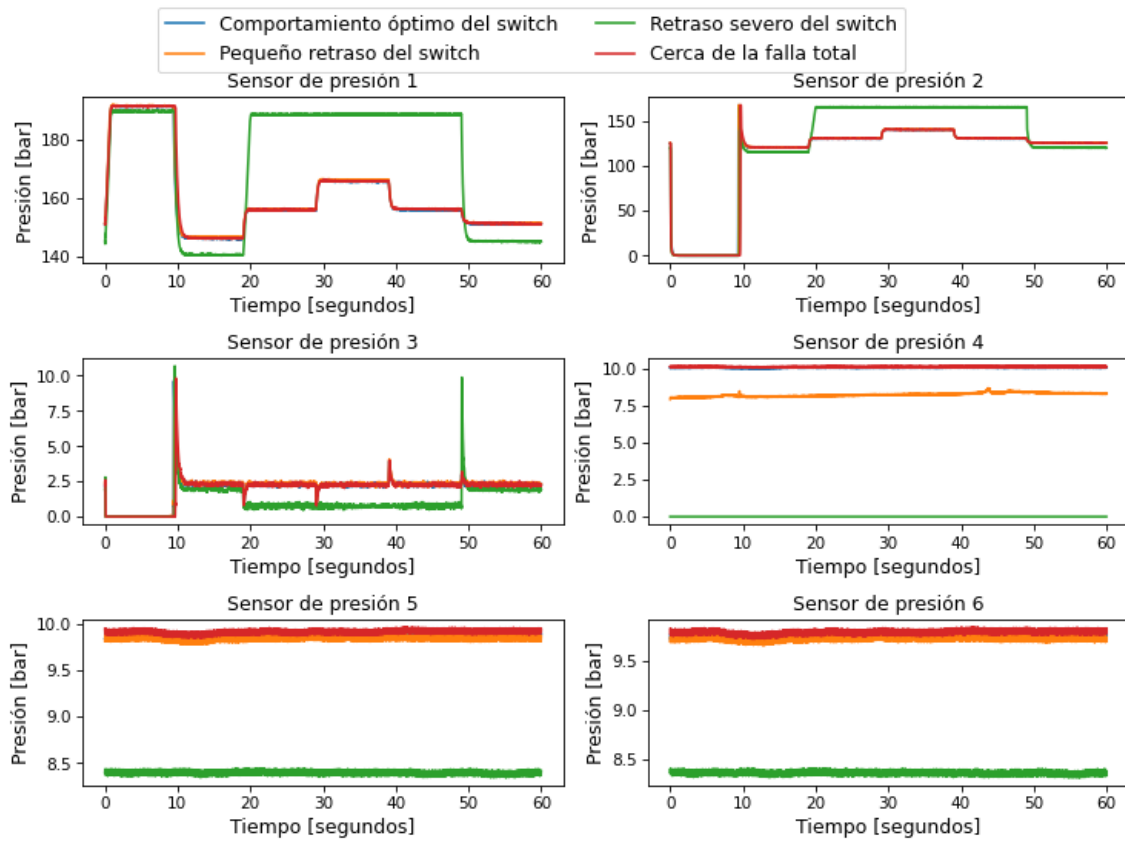


Figura 4.14: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 2.

Datos sin procesar de estado de la válvula

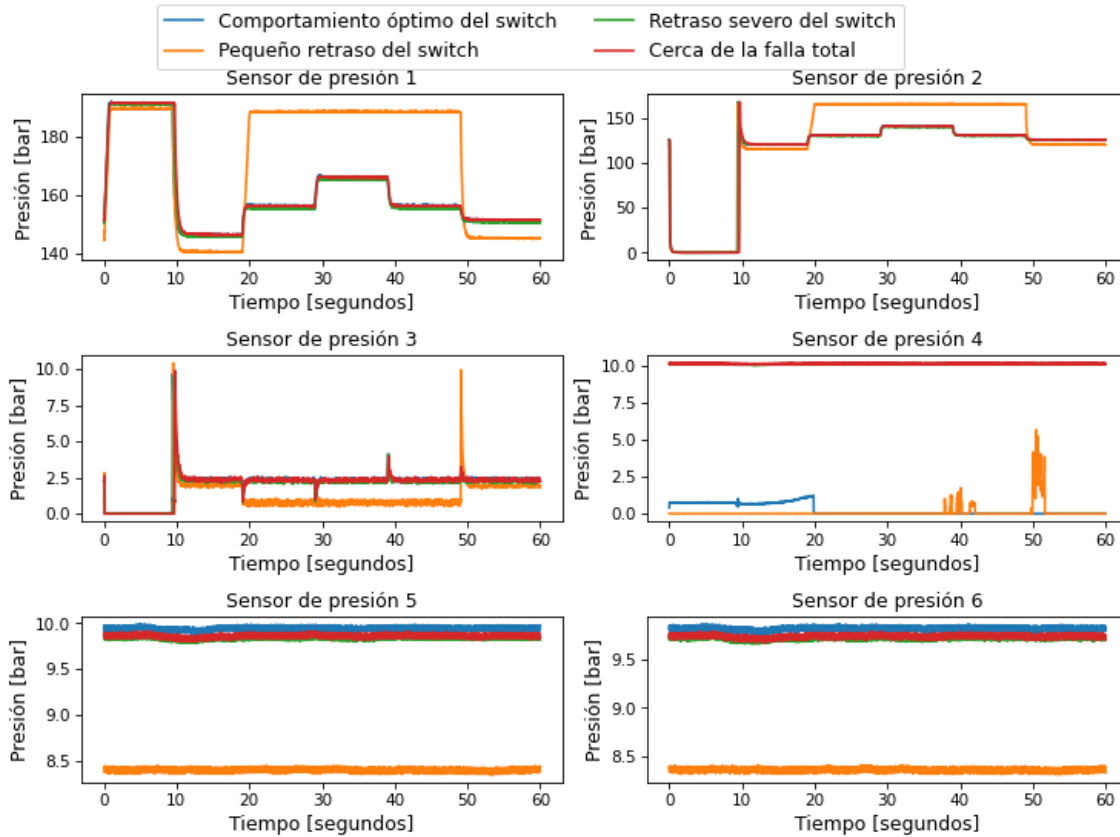


Figura 4.15: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 5.

Para la bomba, cuyos gráficos se encuentran en la figura 4.16, se tienen los mejores resultados entre todos los sensores. A diferencia de las clasificaciones anteriores, en este caso es posible distinguir muy fácilmente las curvas en los sensores PS5 y PS6, en los sensores PS1 y PS3, también es posible pero en menor medida y en los sensores PS2 y PS4 solo se distinguen 2 clases, mientras que la 3ra se solapa a alguna de las primeras.

Datos sin procesar de fuga en la bomba

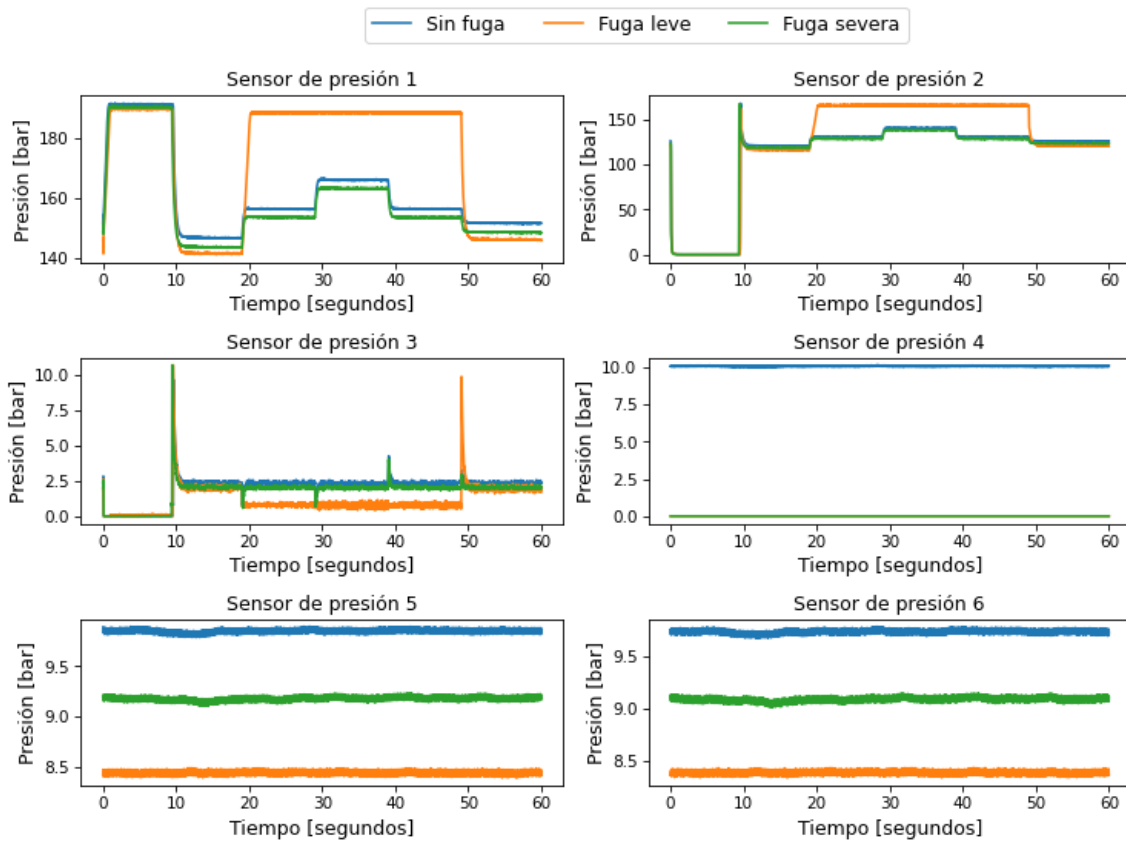


Figura 4.16: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la fuga en la bomba.

En cuanto al estado del **acumulador** (figura 4.17), vemos nuevamente que los sensores PS5 y PS6 consiguen distinguir muy bien entre clases. Por su parte los sensores PS1, PS4 también entregan algún grado de separación entre clases, mientras que en los gráficos de los sensores PS3 y PS2 las clases son casi indistinguibles.

Datos sin procesar de estado del acumulador

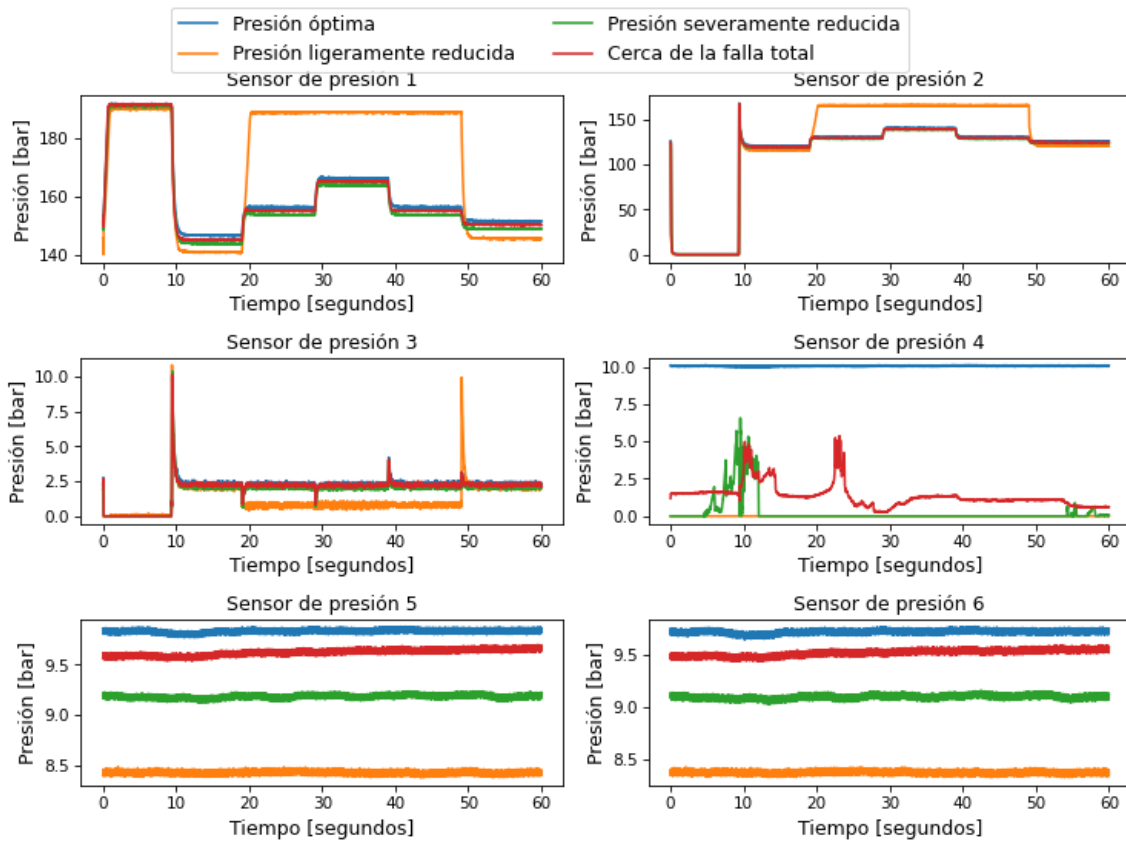


Figura 4.17: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador.

Finalizando con los gráficos de presión, tenemos la **estabilidad del sistema** disponibles en la figura 4.18. Se repiten una vez mas los comportamientos vistos anteriormente: gráficos PS5 y PS6 permiten distinguir muy bien entre clases, mientras que los demás sensores entregan un grado de separación pero en menor medida.

Datos sin procesar de estabilidad del sistema

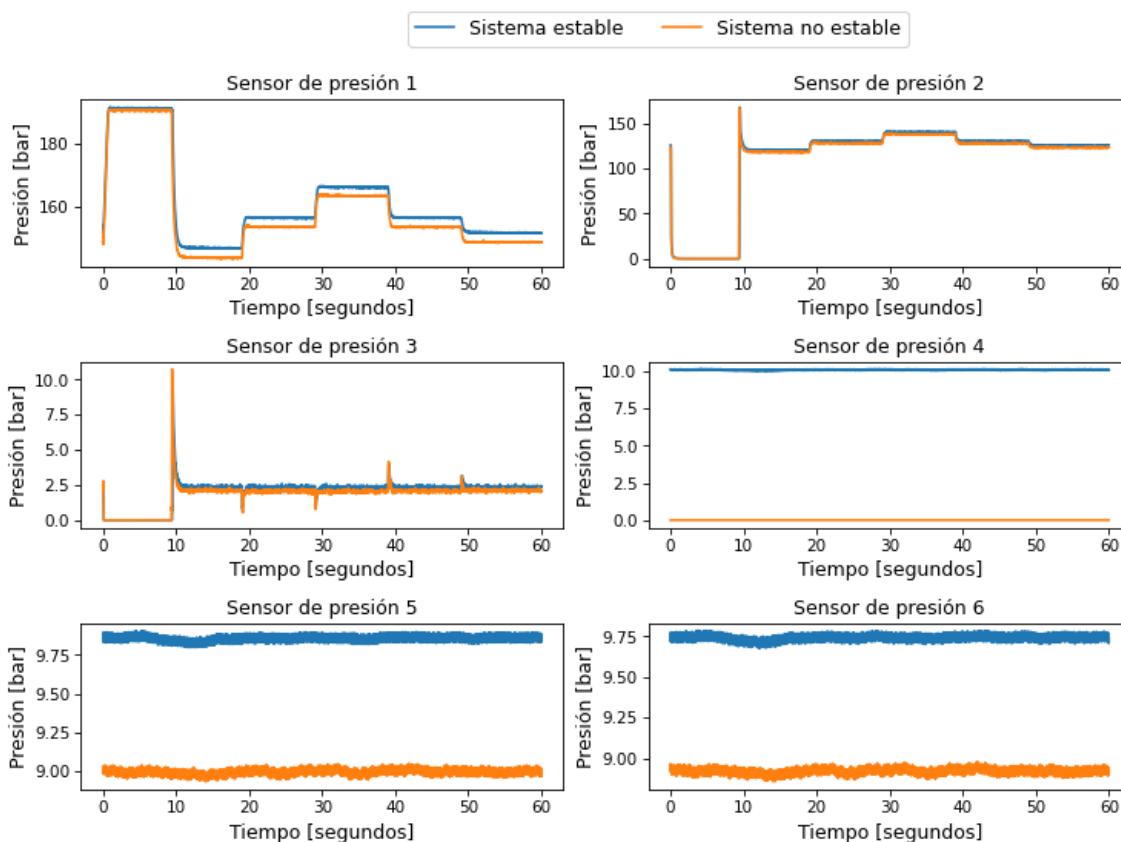


Figura 4.18: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema.

4.1.4. Gráficos de otras variables

Finalmente nos resta analizar un conjunto de sensores que midieron 5 variables distintas y fueron muestreados a diferentes tasas según la tabla 2.3. Debido a que se trata de diferentes variables, entre las curvas de los sensores no se presenta una gran similitud, pero podemos ver que por ejemplo el sensor CP (potencia del enfriador) distingue en general bastante bien las clases mientras que los sensores VS1, SE (factor de eficiencia) y EPS1 (potencia del motor) muestran curvas que se solapan o se acercan mucho.

Datos sin procesar de estado del enfriador

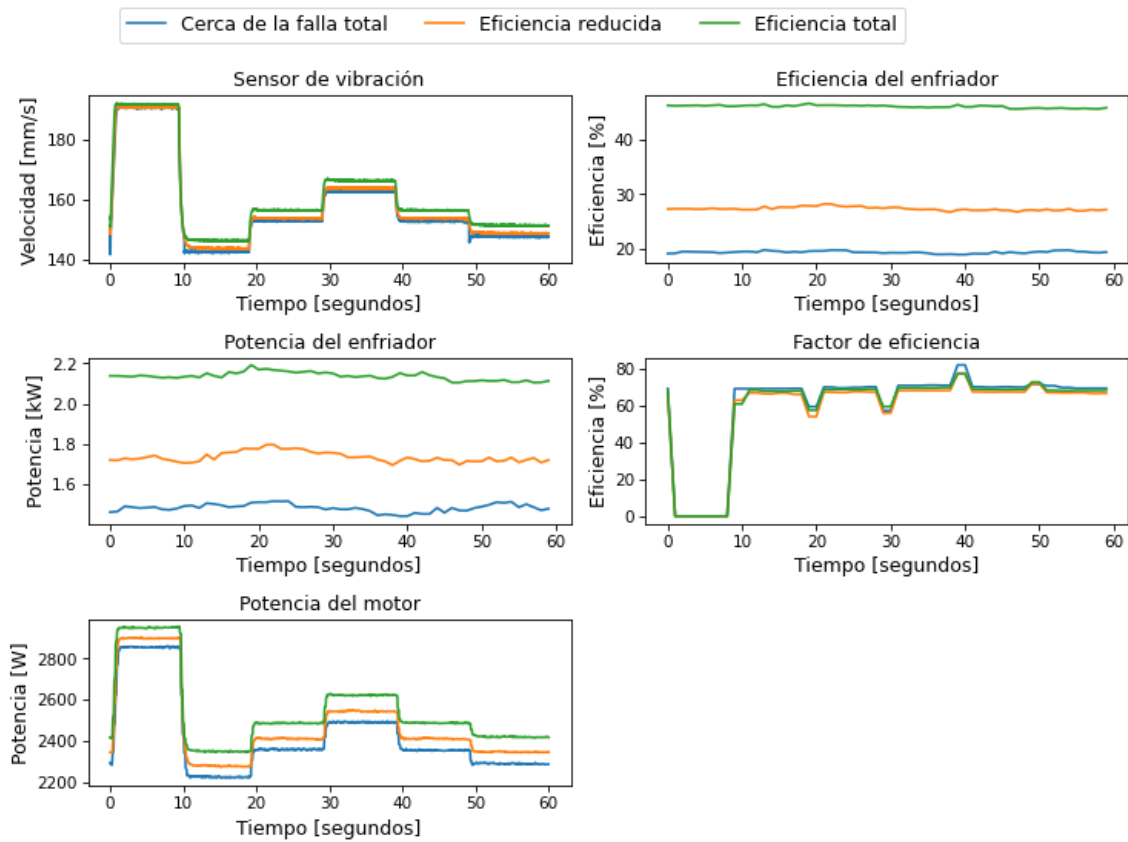


Figura 4.19: Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases del estado del enfriador.

Comenzando con el estado del **enfriador** (figura 4.19), vemos que en todos los sensores salvo en el factor de eficiencia, es posible distinguir las diferentes clases en mayor o menor grado.

Datos sin procesar de estado de la válvula

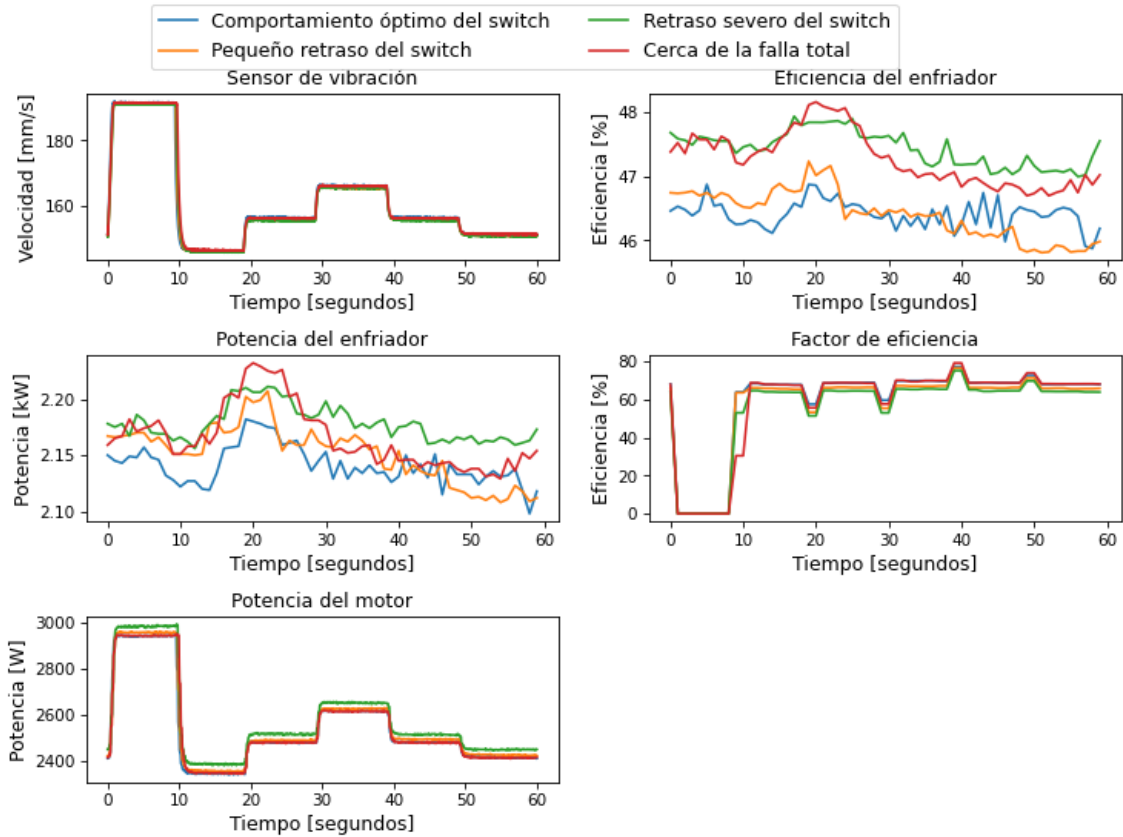


Figura 4.20: Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases del estado de la válvula.

Al observar los gráficos de la **válvula** (figura 4.20) vemos que las clases se encuentran muy solapadas, y que solo los gráficos de potencia y eficiencia del enfriador es posible notara alguna diferencias.

Datos sin procesar de fuga en la bomba

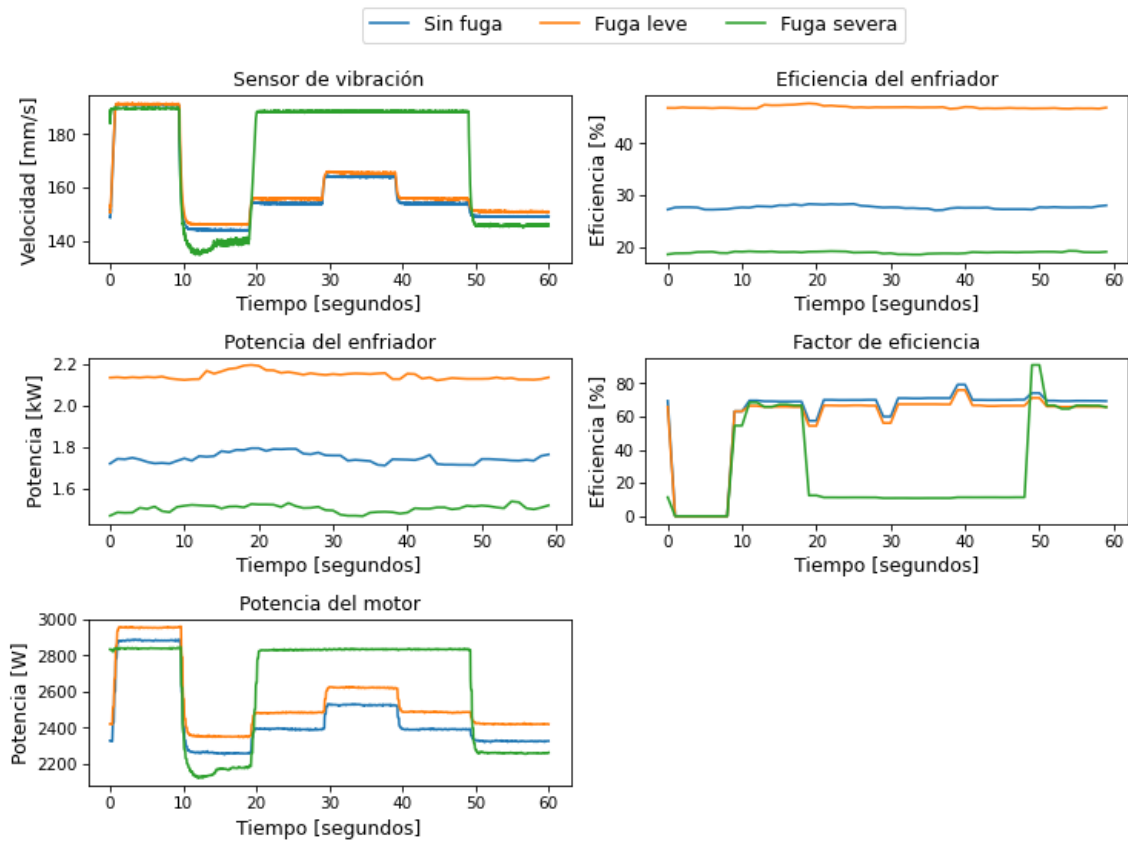


Figura 4.21: Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases de la fuga en la bomba.

Para la **bomba** (figura 4.21) vemos una situación más parecida e incluso un poco mejor que en el caso del enfriador, ya que se distinguen muy bien la clases en todos los sensores.

Datos sin procesar de estado del acumulador

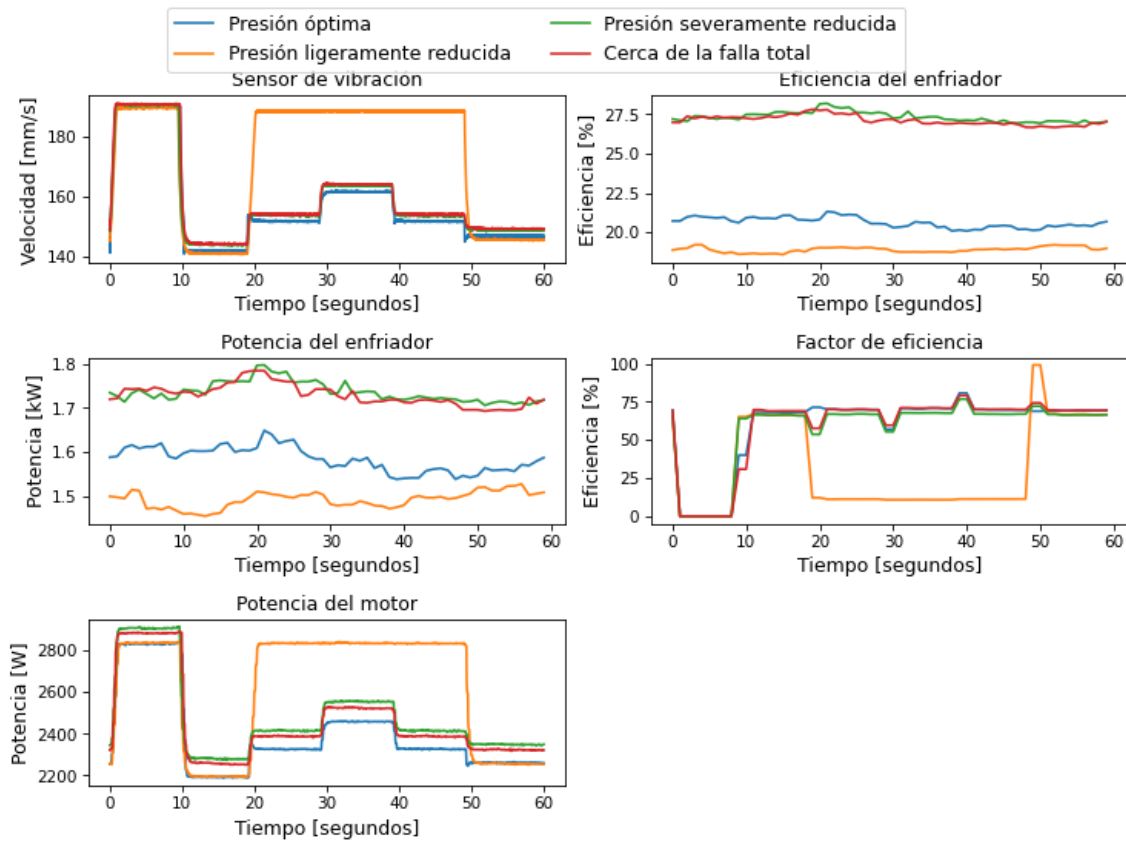


Figura 4.22: Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases del estado del acumulador.

En cuanto al acumulador, este presenta unas curvas un poco mas indistinguibles y parecidas al caso de la válvula, aunque en los sensores de potencia y eficiencia del enfriador si es posible distinguir bien las clases y menor medida en el sensor de potencia del motor.

Datos sin procesar de estabilidad del sistema

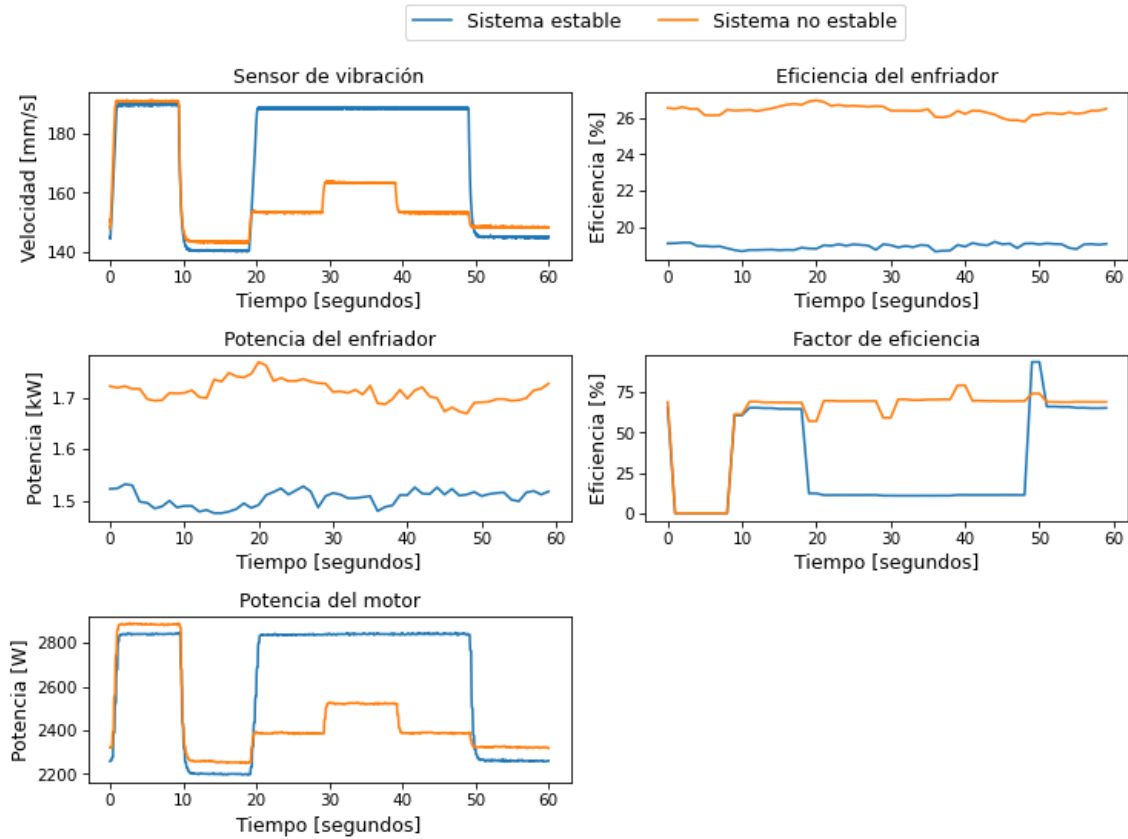


Figura 4.23: Gráficos de los datos sin procesar obtenidos por sensores de diferentes variables, separando según las clases de la estabilidad del sistema.

Para la **estabilidad del sistema** (figura 4.23) tenemos que nuevamente un buen caso donde todos los sensores en mayor o menor medida separan los diferentes estados de salud correspondientes a esta clasificación.

4.2. Gráficos de los parámetros de tiempo

A continuación se presentan figuras con los parámetros de tiempo escogidos para las diferentes clasificaciones de estados de salud, variando además los largos de ventanas sobre los cuales fueron extraídos estos parámetros. Estas figuras muestran gráficos de dispersión, donde el “eje Y” representa el respectivo valor del parámetro de tiempo y el “eje X” representa simplemente el índice que toma cada dato al ser separado en arreglos correspondientes a cada clase. Para mostrar estos valores, se escogieron 6 sensores que se consideraron como el conjunto mas representativo, sin embargo para ver los gráficos con una mayor cantidad de sensores se puede consultar el anexo A.2. El **propósito de analizar estas figuras**, no es revisar sensor a sensor los cambios como en el subcapítulo anterior, sino que ver de forma general como cambia el comportamiento de los parámetros de tiempo.

4.2.1. Diferencias entre los parámetros de tiempo

Para comenzar revisamos las figuras 4.24 a 4.31, donde las 4 primeras representan los parámetros de tiempo para la clasificación del estado del enfriador y las 4 últimas representan los parámetros de tiempo para el estado de la válvula, donde en los 8 casos se usaron ventanas de tiempo de 60 segundos, es decir, se extrajo solo 1 dato por ciclo.

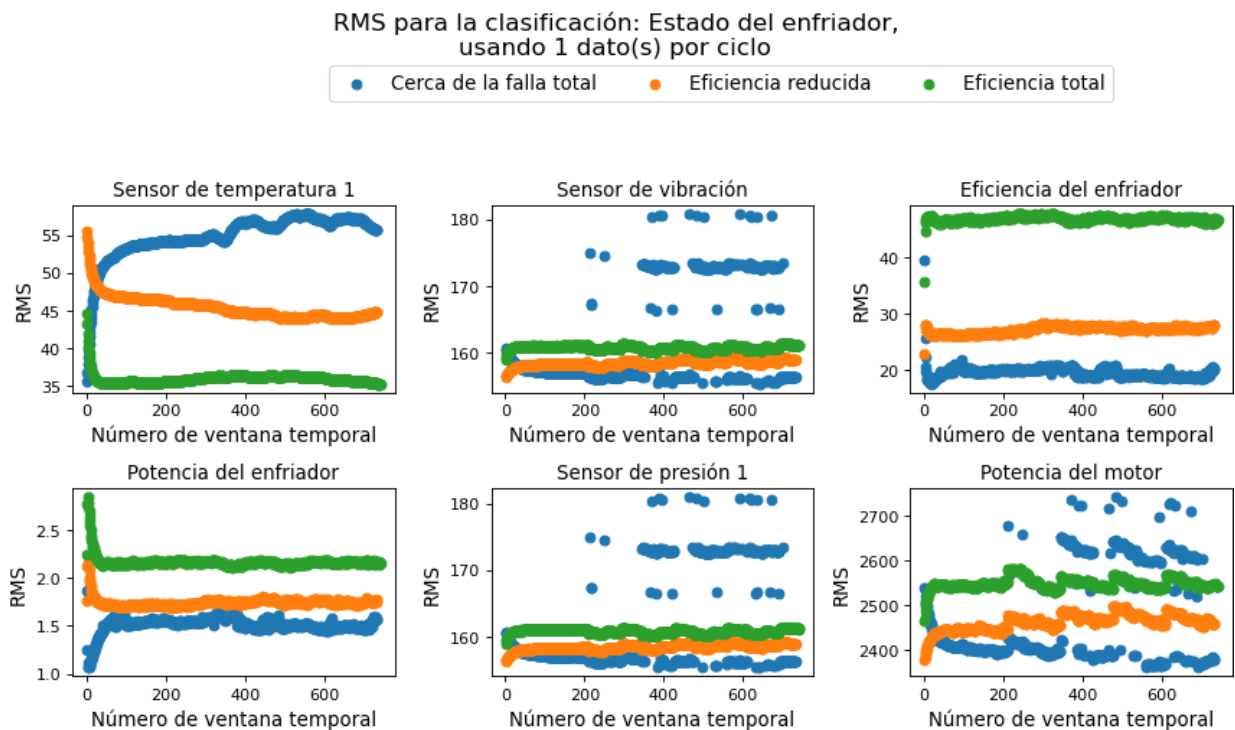


Figura 4.24: Gráficos del RMS obtenido según las clases del estado del enfriador.

Valor peak to peak para la clasificación: Estado del enfriador, usando 1 dato(s) por ciclo

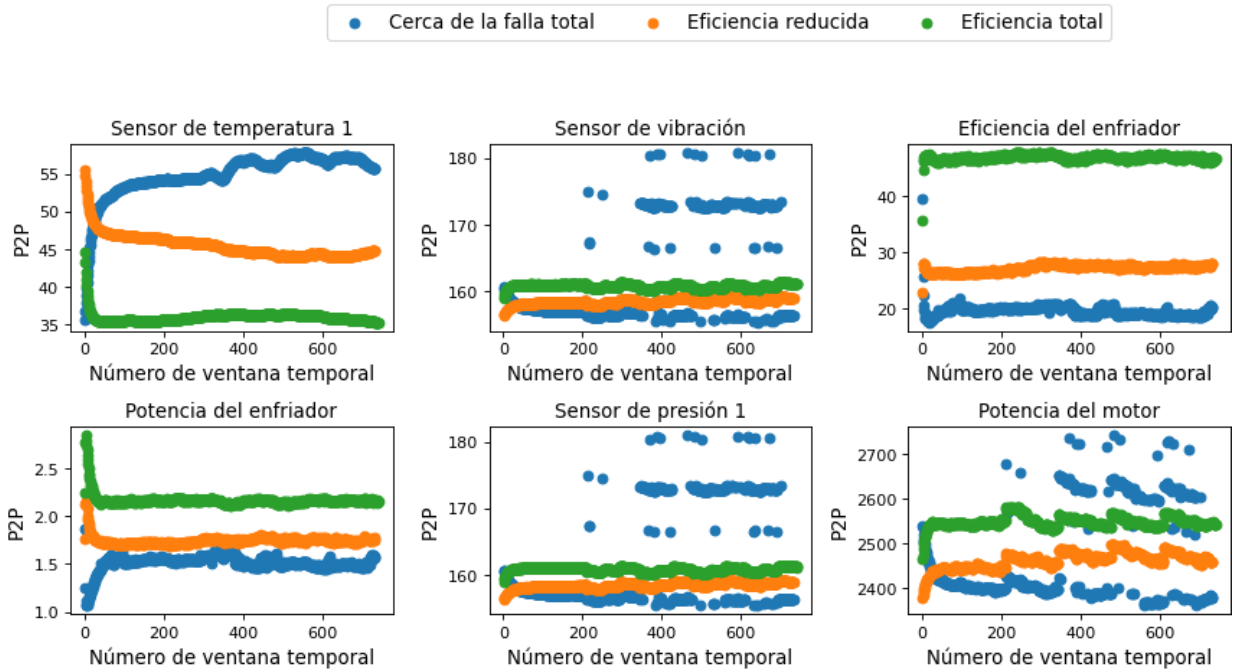


Figura 4.25: Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.

Media para la clasificación: Estado del enfriador, usando 1 dato(s) por ciclo

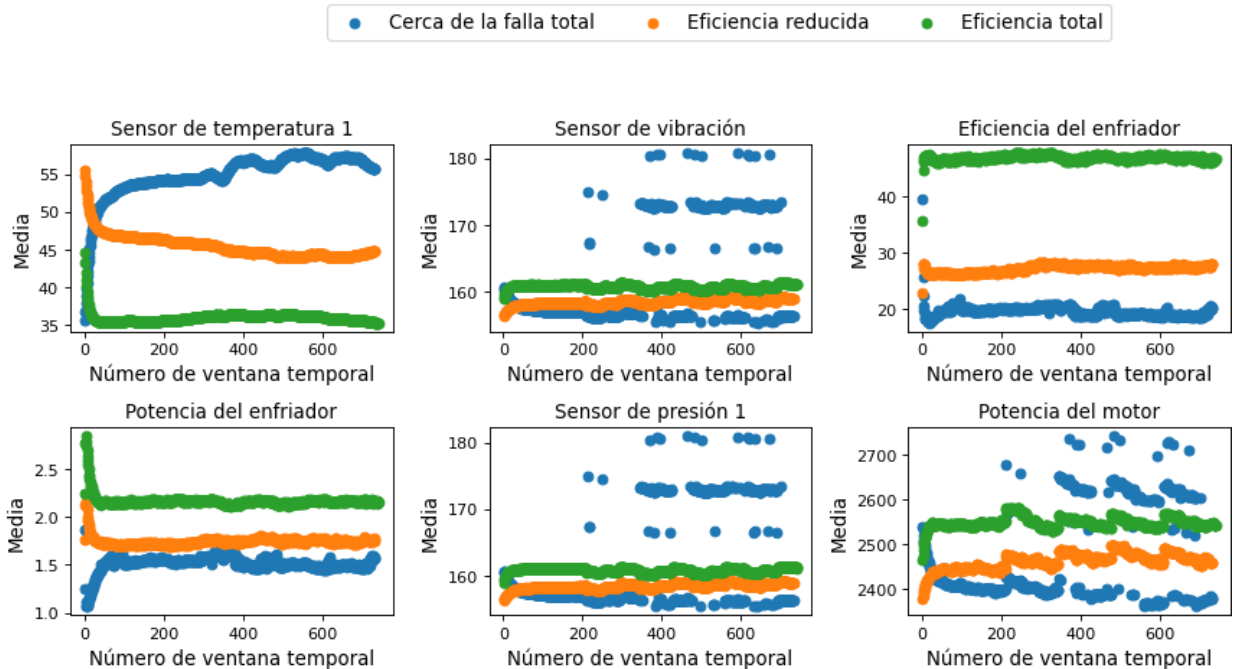


Figura 4.26: Gráficos de la media obtenida según las clases del estado del enfriador.

Varianza para la clasificación: Estado del enfriador,
usando 1 dato(s) por ciclo

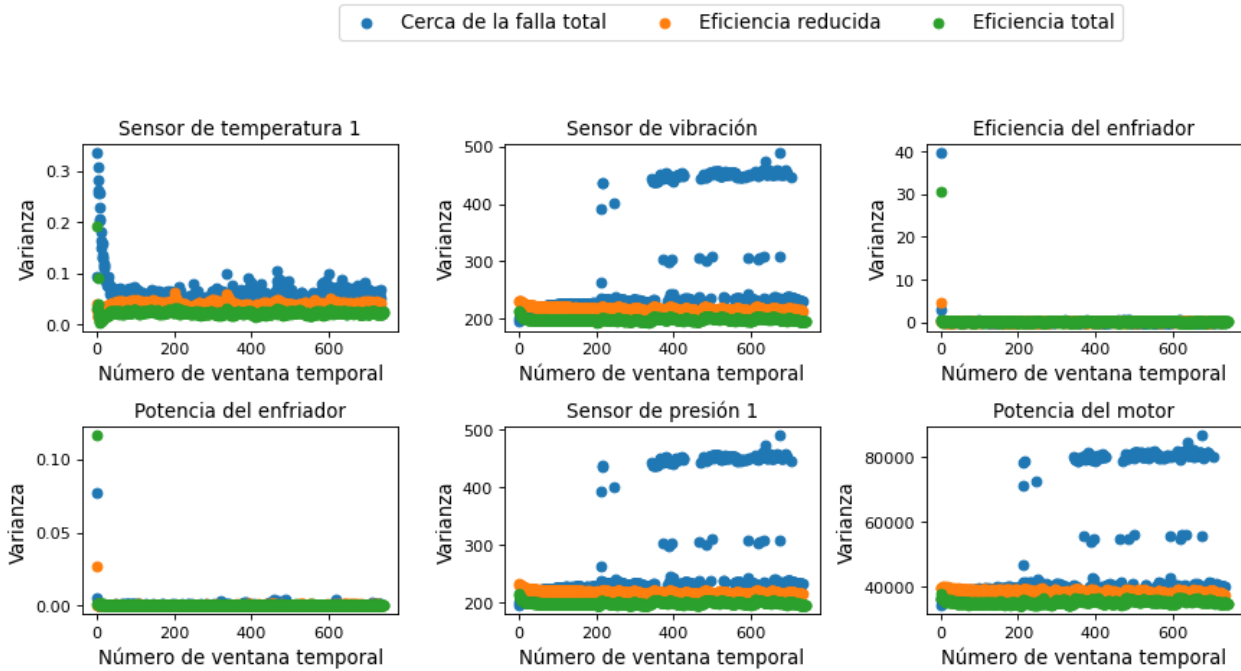


Figura 4.27: Gráficos de la varianza obtenida según las clases del estado del enfriador.

RMS para la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo

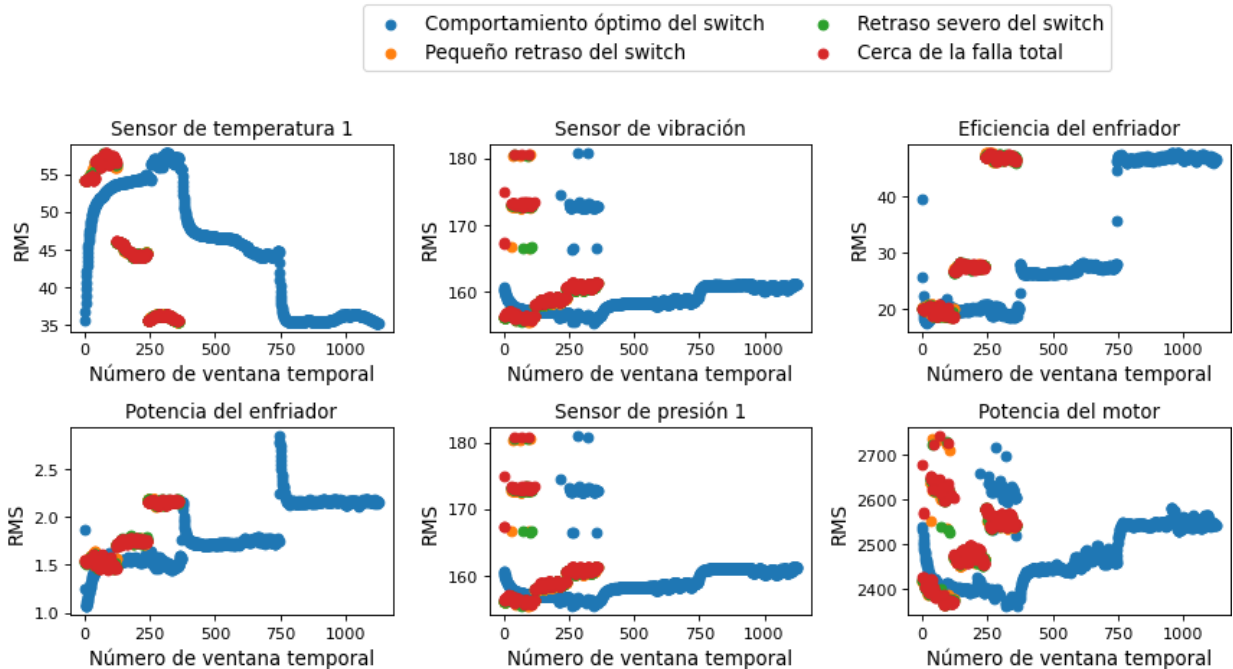


Figura 4.28: Gráficos del RMS obtenido según las clases del estado de la válvula.

Valor peak to peak para la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo

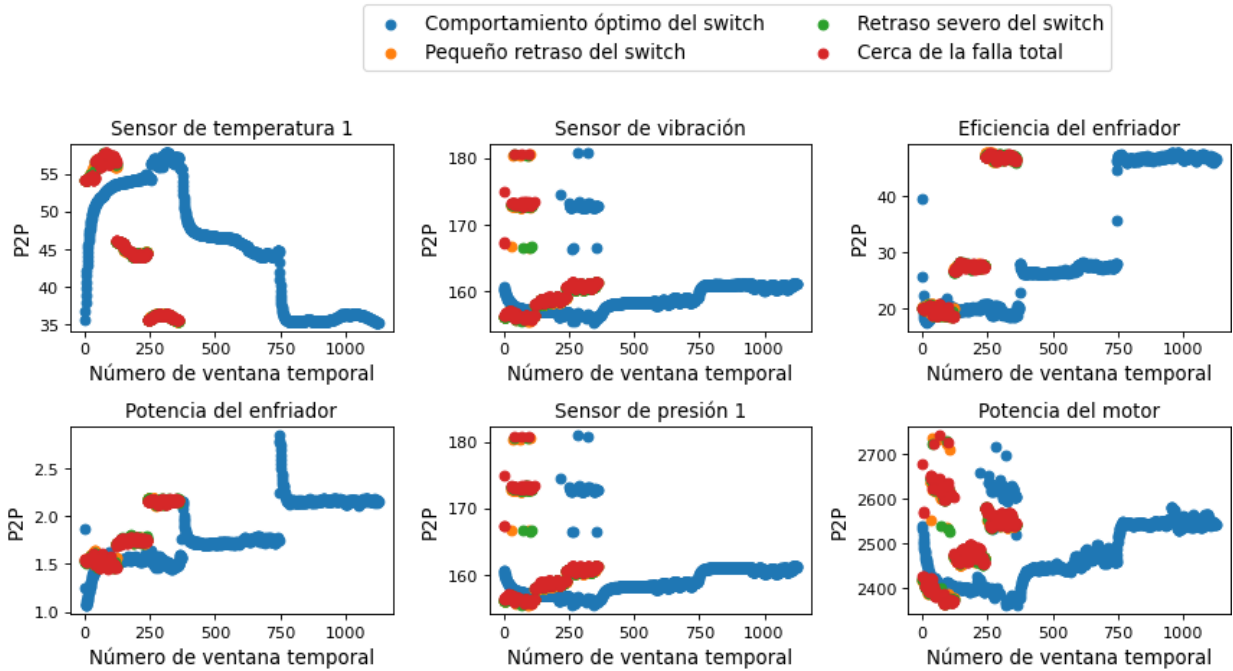


Figura 4.29: Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.

Media para la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo

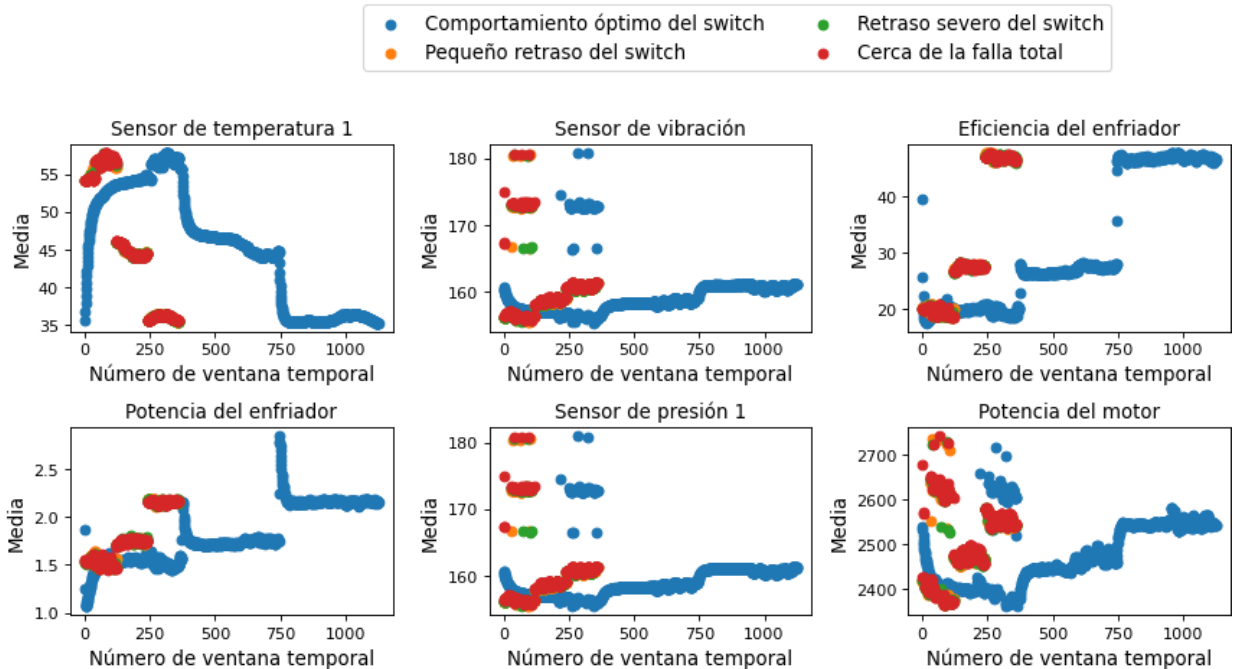


Figura 4.30: Gráficos de la media obtenida según las clases del estado de la válvula.

Varianza para la clasificación: Estado de la válvula, usando 1 dato(s) por ciclo

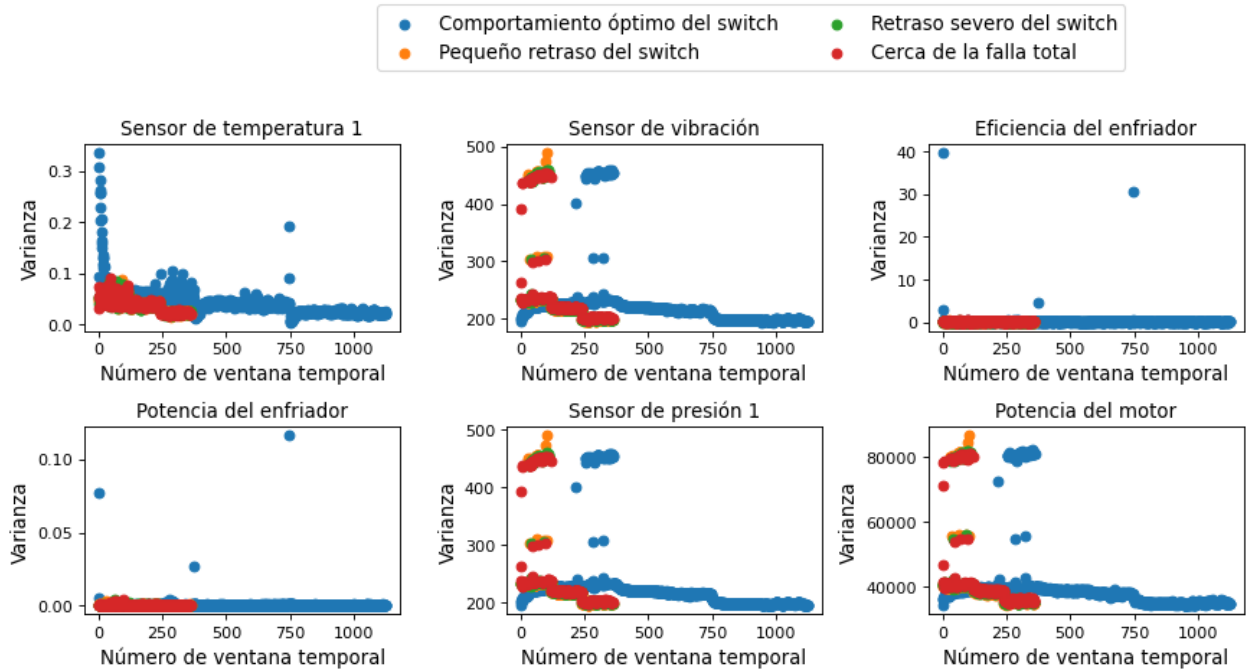


Figura 4.31: Gráficos de la varianza obtenido según las clases del estado de la válvula.

Lo que se observa para ambas clasificaciones (enfriador y válvula), es que los resultados obtenidos por el RMS, el valor peak to peak y la media, son prácticamente iguales, mientras que también en ambos casos la varianza presenta un comportamiento distinto a los demás. Adicionalmente se destaca que a simple vista la separación entre clases mejoró con respecto a los datos sin procesar.

4.2.2. Cambios según las diferentes clasificaciones

Para estudiar la diferencia de los parámetros de tiempo con las distintas calificaciones, se mostrarán solo los resultados de RMS y varianza, ya que como se pudo ver anteriormente y como también es posible de corroborar en los anexos, no se perciben cambios significativos entre el valor peak to peak y la media con respecto al RMS.

RMS para la clasificación: Fuga en la bomba, usando 1 dato(s) por ciclo

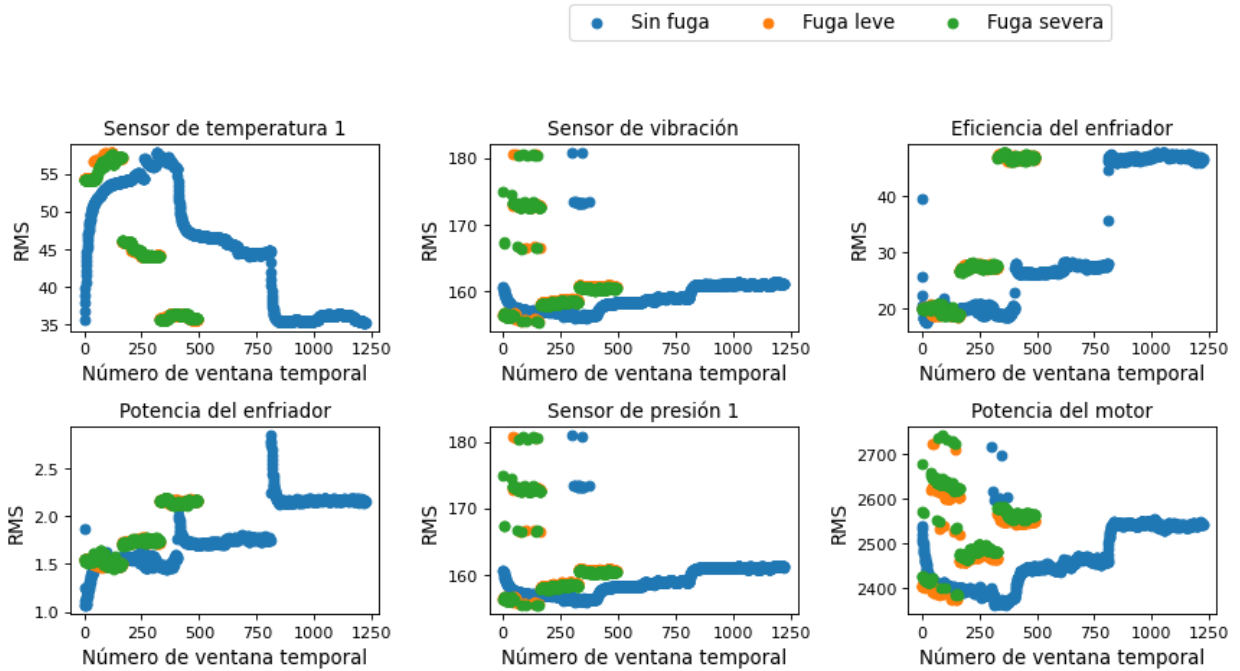


Figura 4.32: Gráficos del RMS obtenido según las clases de la fuga en la bomba.

Varianza para la clasificación: Fuga en la bomba, usando 1 dato(s) por ciclo

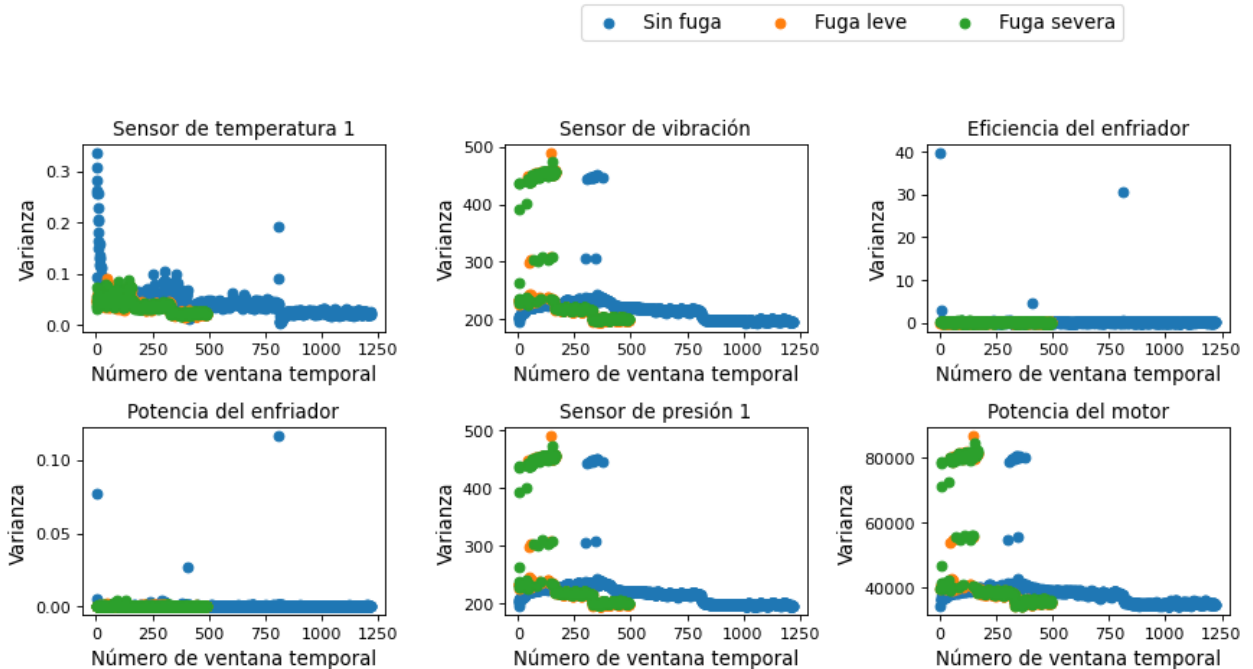


Figura 4.33: Gráficos de la varianza obtenida según las clases de la fuga en la bomba.

RMS para la clasificación: Estado del acumulador, usando 1 dato(s) por ciclo

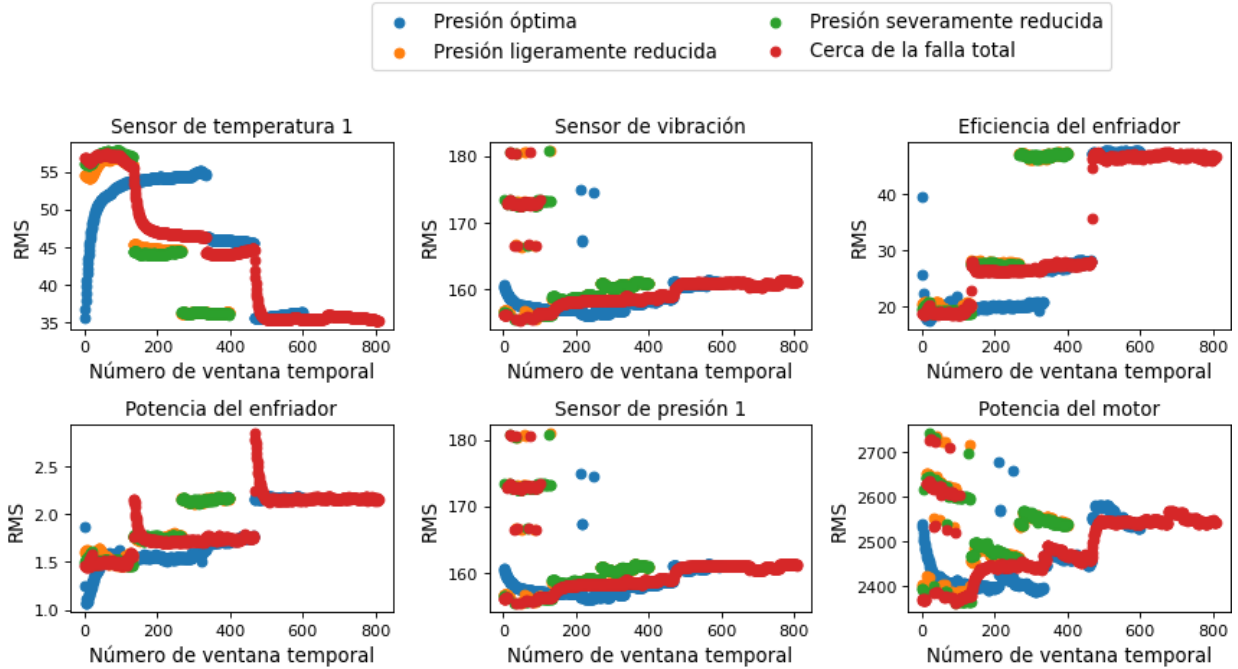


Figura 4.34: Gráficos del RMS obtenido según las clases del estado del acumulador.

Varianza para la clasificación: Estado del acumulador, usando 1 dato(s) por ciclo

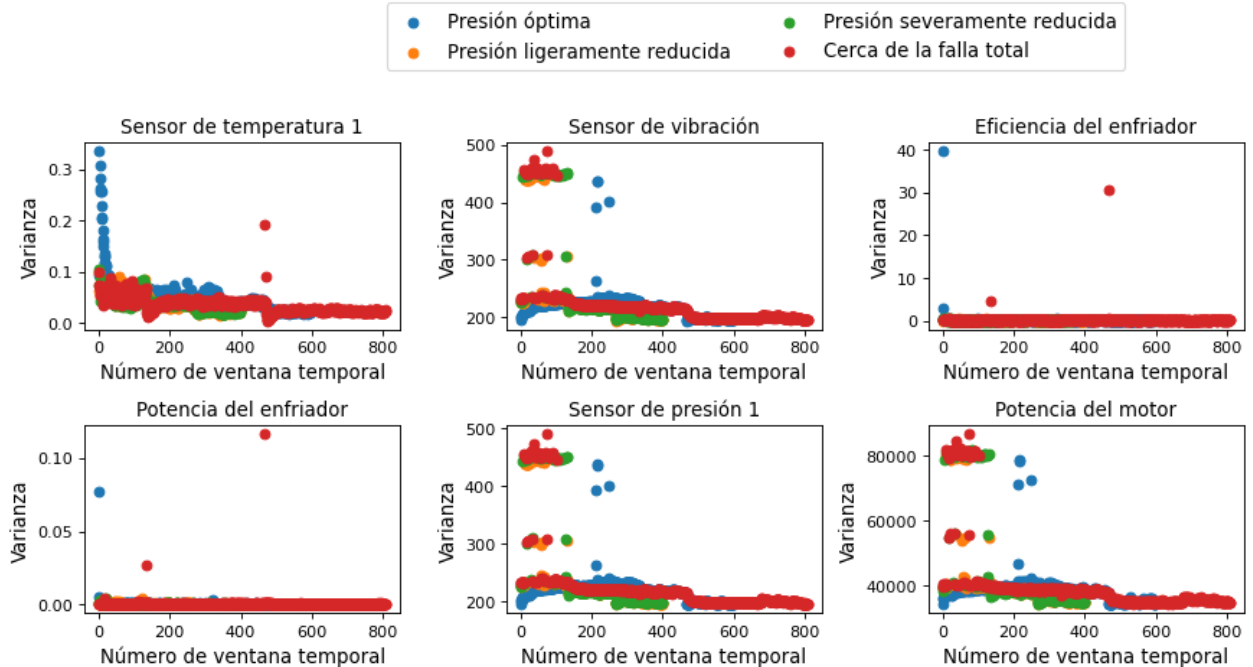


Figura 4.35: Gráficos de la varianza obtenida según las clases del estado del acumulador.

Con esto se puede notar que el RMS logra separar bastante bien las diferentes clases de los componentes, mientras que la varianza en muchas ocasiones solapa los datos. En base a esto podemos esperar que los modelos que utilicen la varianza, presentaran un desempeño más bajo con respecto a los demás parámetros de tiempo.

4.2.3. Cambios diferentes ventanas de tiempo

Finalmente se busca estudiar cambios al extraer los parámetros de tiempo a partir de ventanas de diferente largo. Para lograr lo anterior, se revisaron los resultados al variar la cantidad de datos extraídos por ciclo, en 2 y 6 datos, para una combinación de RMS-enfriador y otra combinación de varianza-válvula.

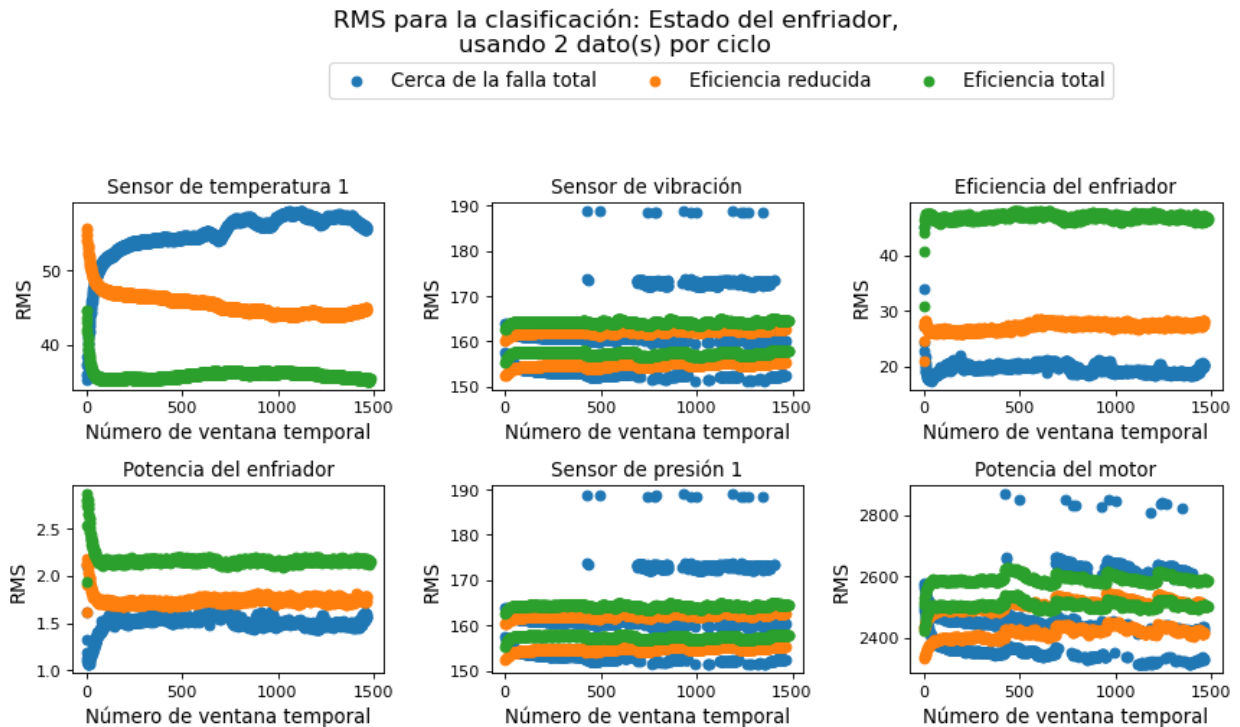


Figura 4.36: Gráficos del RMS obtenido según las clases del estado del enfriador, usando 2 datos por ciclo.

RMS para la clasificación: Estado del enfriador,
usando 6 dato(s) por ciclo

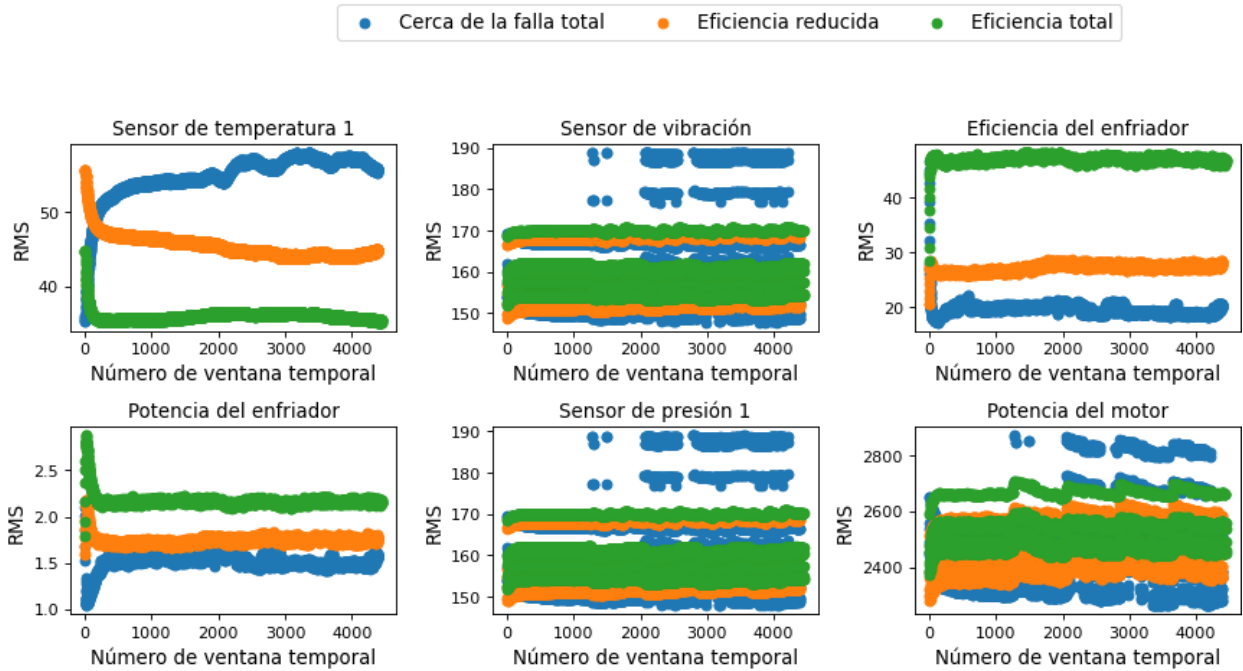


Figura 4.37: Gráficos del RMS obtenido según las clases del estado del enfriador, usando 6 datos por ciclo.

Varianza para la clasificación: Estado de la válvula,
usando 2 dato(s) por ciclo

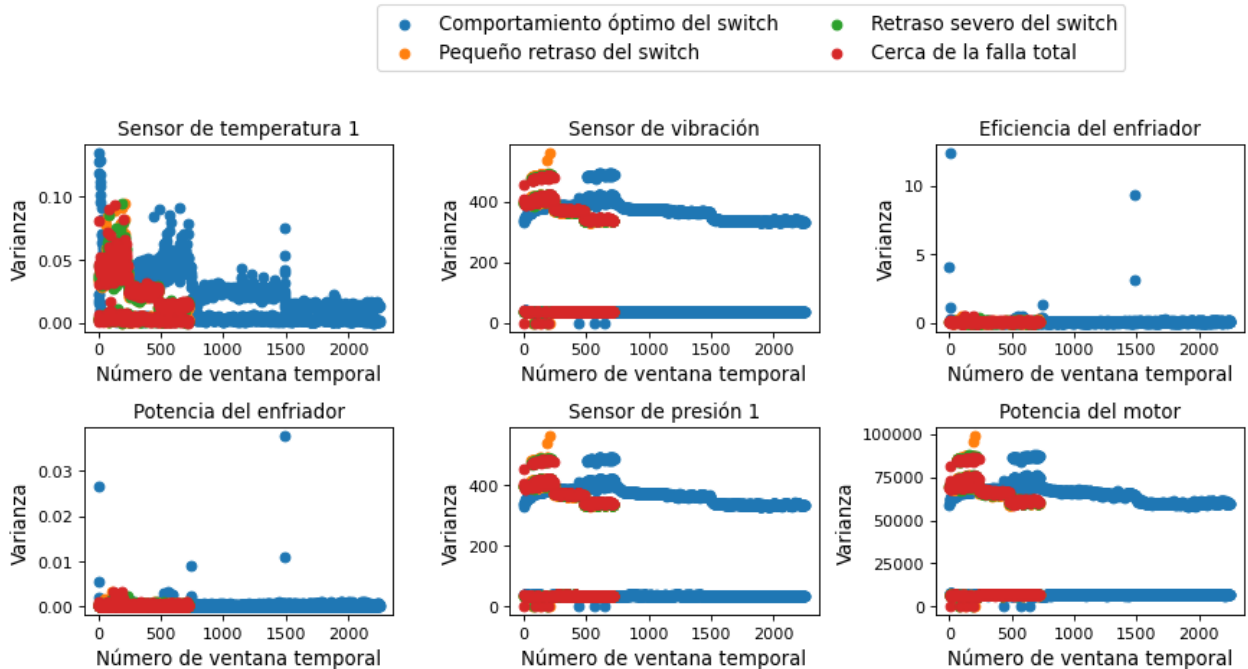


Figura 4.38: Gráficos del RMS obtenido según las clases del estado de la válvula, usando 2 datos por ciclo.

Varianza para la clasificación: Estado de la válvula,
usando 6 dato(s) por ciclo

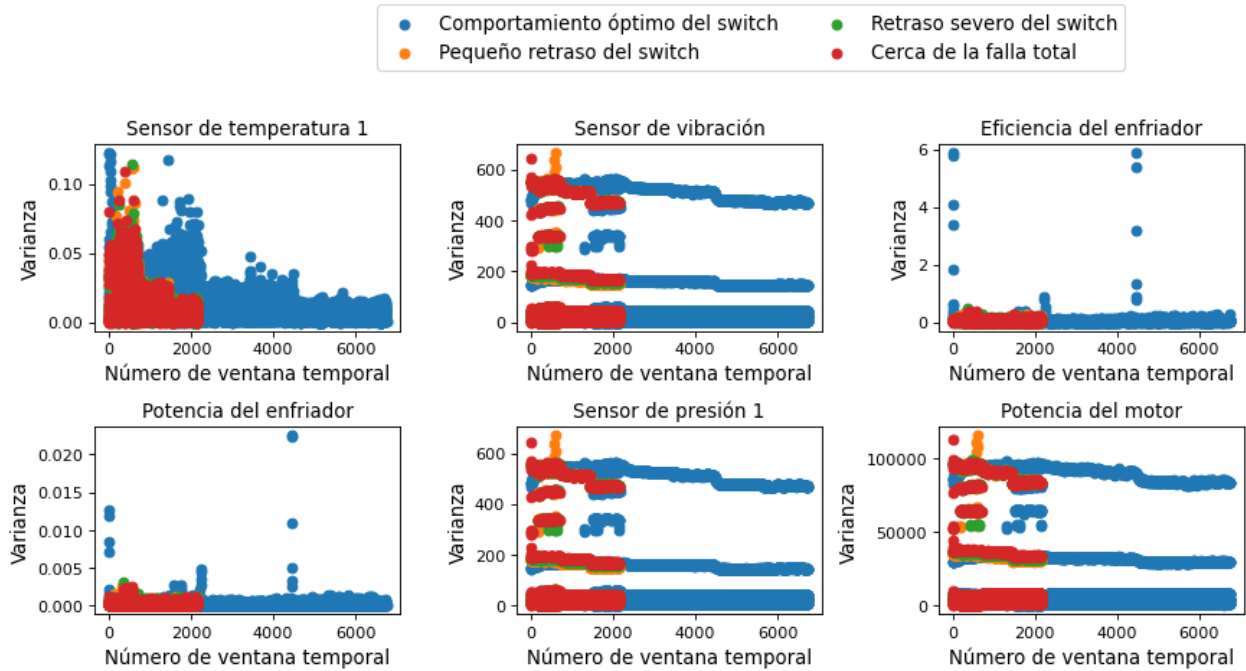


Figura 4.39: Gráficos del RMS obtenido según las clases del estado de la válvula, usando 6 datos por ciclo.

Al comparar las figuras 4.36 y 4.37 con la figura 4.24 y las figuras 4.38 y 4.39 con la figura 4.31, notamos que a grandes rasgos aumentan las aglomeraciones de puntos al aumentar los tamaños de ventana, lo que produce a su vez que tiendan a solaparse los puntos de las diferentes clases.

4.3. Resultados a partir de los modelos LDA

A partir de este capítulo se analiza el desempeño de los modelos al clasificar los estados de salud de los componentes al variar hiperparámetros, ventanas de tiempo y parámetros de tiempo. Particularmente para los modelos LDA se decidió **NO variar hiperparámetros del propio modelo**, lo cual permitió poder mostrar todos los resultados obtenidos en el proceso de selección de features o sensores. Cada una de las figuras 4.40 a 4.44 corresponde a cada una de las clasificaciones del estado de salud, las cuales contienen 4 gráficos correspondientes a al desempeño obtenido por el modelo LDA usando datos obtenidos por los 4 diferentes estados de salud. Cada gráfico en específico muestra la evolución de la accuracy al variar la cantidad de datos por ciclos extraídos.

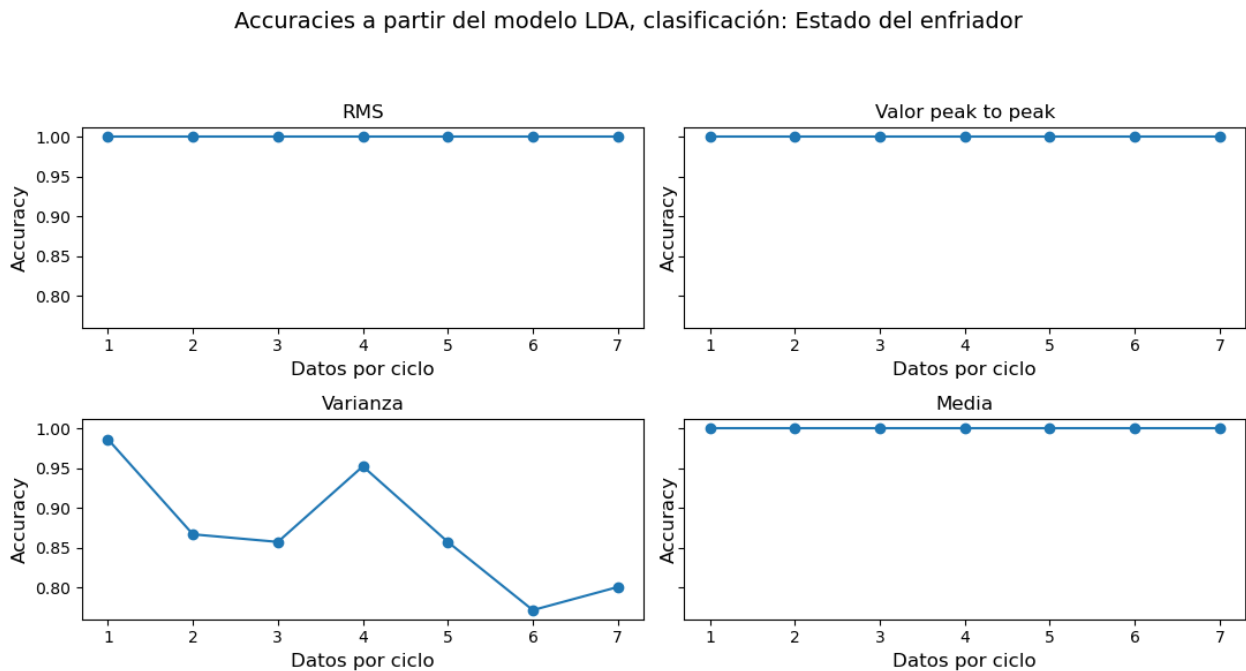


Figura 4.40: Valores de accuracy para modelos LDA.

Como se esperaba de acuerdo a los subcapítulos anteriores, los gráficos correspondientes al **enfriador** (figura 4.40) presentan valores muy buenos de accuracy, incluso superando expectativas, ya que se acercan mucho a una clasificación perfecta. Para el gráfico de la varianza se puede ver que con 1 dato por ciclo se logra un muy buen resultado, para luego mostrar una tendencia a la baja al aumentar la cantidad de datos extraídos.

Para el caso de los resultados de la **válvula**, representados en la figura 4.41, se puede ver que presentan un comportamiento bastante distinto al enfriador. En primer lugar se destacan los valores de accuracy, donde notamos que en el caso del enfriador el accuracy mas bajo es de aproximadamente un 75 %, mientras que para el caso de la válvula no se supera el 65 %, llegando incluso a presentar valores cercanos o bajo el 50 %. También se destaca que esta es la única clasificación donde aparentemente el accuracy sube a medida que se aumenta la cantidad de datos extraídos por ciclos.

Accuracies a partir del modelo LDA, clasificación: Estado de la válvula

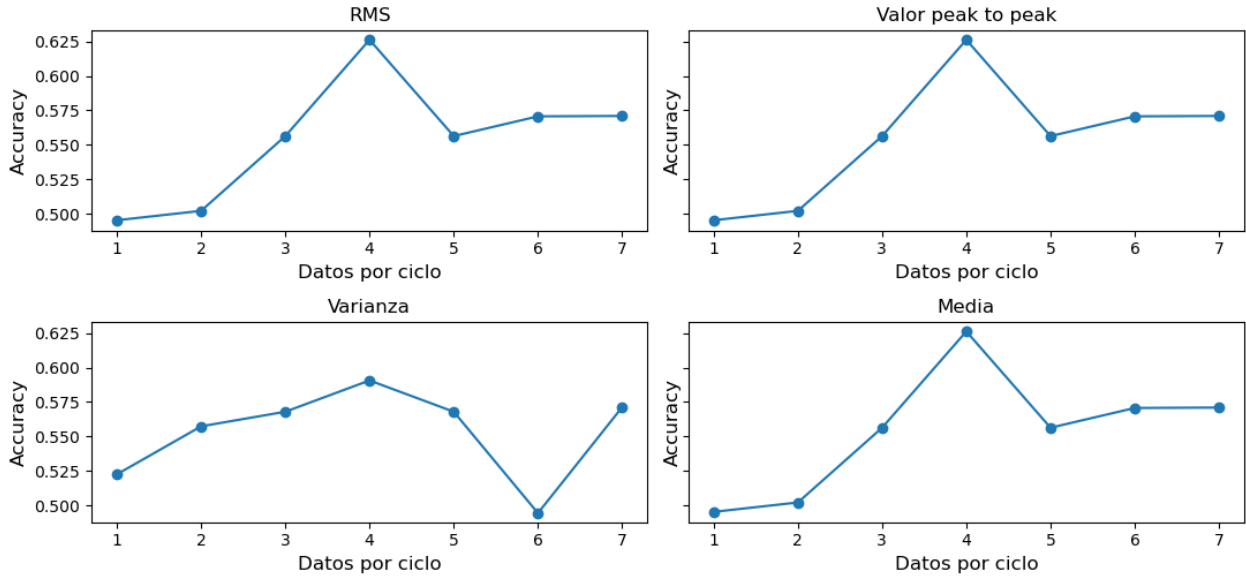


Figura 4.41: Valores de accuracy para modelos LDA.

En cuanto a los gráficos del estado de la **bomba** disponibles en la figura 4.42, se repite el hecho de que las curvas del RMS, la media y el valor peak to peak son prácticamente iguales, mientras que la curva de la varianza se comporta distinto.

Accuracies a partir del modelo LDA, clasificación: Fuga en la bomba

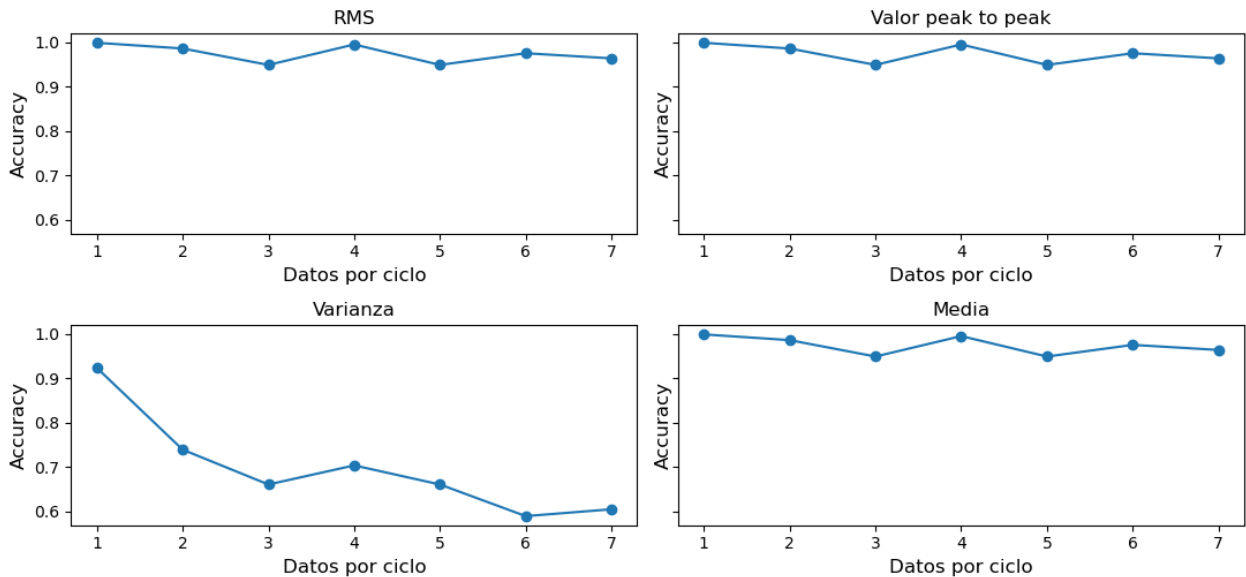


Figura 4.42: Valores de accuracy para modelos LDA.

Para el caso de los primeros 3 parámetros de tiempo mencionados, se ve que alcanzan valores muy altos de accuracy llegando o acercándose mucho al 100 %, donde la cantidad de datos por ciclo no parece afectar de forma significativa el desempeño. Por el lado de la curva de la varianza, se ve que con 1 dato por ciclo se obtiene un accuracy superior al 90 %, pero luego se comienza a reducir abruptamente al aumentar la cantidad de datos extraídos por ciclo llegando a valores cercanos al 60 %

Accuracies a partir del modelo LDA, clasificación: Estado del acumulador

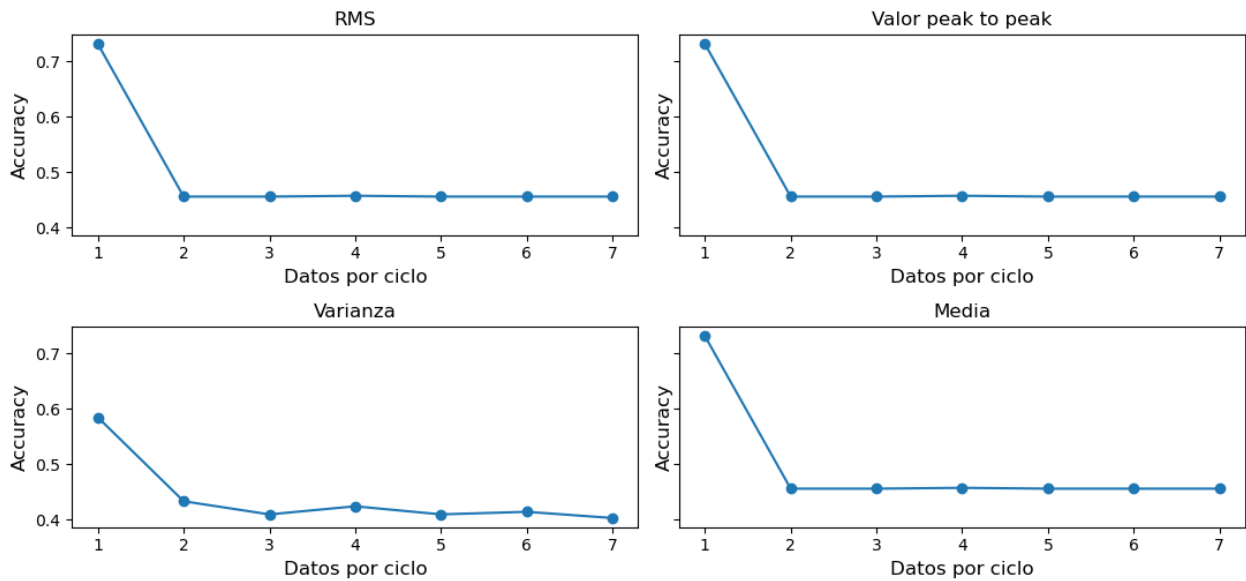


Figura 4.43: Valores de accuracy para modelos LDA.

Para el **acumulador** (figura 4.43) se ve que a diferencia de las figuras anteriores las curvas de los 4 parámetros de tiempo presentan un comportamiento similar entre ellas, partiendo con un valor más alto de accuracy con 1 dato por ciclo para luego mantener un valor constante o aproximadamente constante desde 2 a 7 datos por ciclo. El valor más alto alcanzado por el RMS, la media y el valor peak to peak es cercano al 75 %, similar a lo obtenido por la válvula, sin embargo, en el caso del acumulador el accuracy baja hasta aproximadamente un 45 % para todos los demás puntos, a diferencia de la válvula, que solo baja de 50 %. El caso de la varianza es aún peor, ya que esta no supera el 60 % de accuracy y luego baja a valores cercanos al 40 %.

Accuracies a partir del modelo LDA, clasificación: Estabilidad del sistema

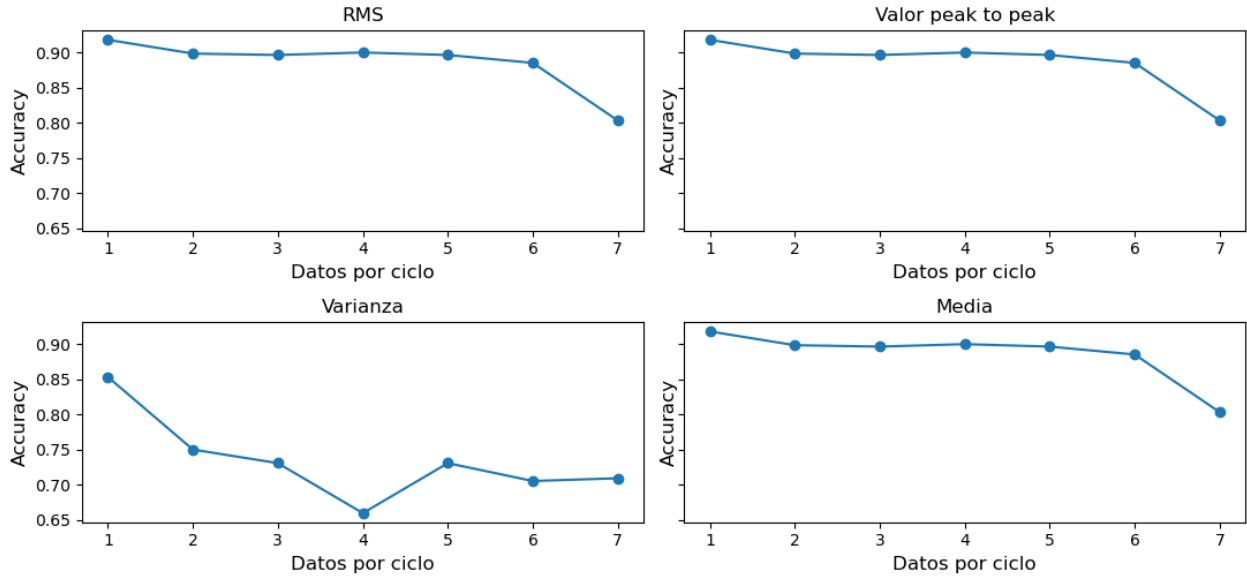


Figura 4.44: Valores de accuracy para modelos LDA.

Para la caso del **estabilidad del sistema** presente en la figura 4.44, notamos que nuevamente se presenta un comportamiento similar entre el RMS, la media y el valor peak to peak. Para las 4 curvas también se repite la tendencia a la baja al aumentar los datos por ciclo, pero esta es mucho leve en los 3 primeros gráficos en comparación a la varianza. En cuanto a los valores de accuracy, se tiene que cada parámetro de tiempo superó a sus similares en la clasificación del estado de la válvula y del acumulador, pero aparentemente se encuentran un poco más bajos al comparar con el estado del enfriador y la fuga en la bomba.

4.4. Resultados a partir de los modelos RF

En este capítulo se muestran los resultados obtenidos a partir de los modelos RF para las 5 clasificaciones. En primer lugar se decidió eliminar la media entre los parámetros de tiempo para no sobrecargar de información las figuras considerando que sus resultados son bastante similares al RMS o al valor peak to peak. Para ver las figuras incluyendo la media se puede consultar el anexo A.3. Estos gráficos fueron elaborados para analizar la evolución del accuracy al variar la cantidad de árboles, la cantidad de datos extraídos por ciclo y la diferencia entre el criterio de gini y el criterio de la entropía de la información. Cada color de curva representa un cantidad de datos por ciclo distinta, las curvas con marcadores circulares corresponden a modelos RF que usaron el criterio de gini, mientras que las curvas con marcadores cuadrados corresponden a modelos que usaron el criterio de entropía de la información.

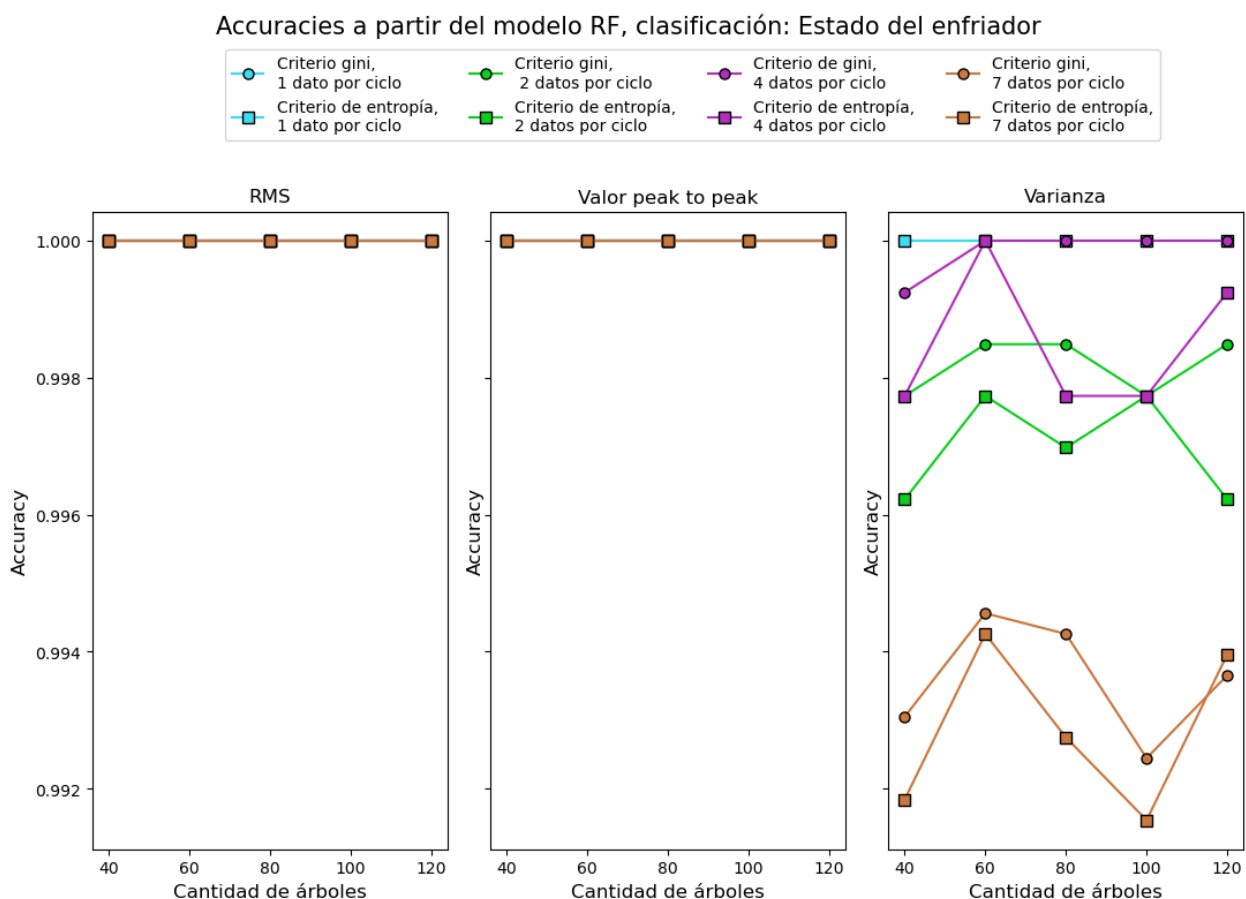


Figura 4.45: Valores de accuracy obtenidos de modelos RF.

En la figura 4.45 se muestran los resultados de la clasificación del estado del **enfriador**. Similar a lo que sucedió anteriormente con el LDA para el enfriador, los modelos que usaron los datos provenientes del RMS y el valor peak to peak, obtuvieron un accuracy de un 100% sin importar la cantidad de datos por ciclo, la cantidad de árboles o el criterio.

Para la varianza vemos en la figura 4.46 que las curvas correspondientes a 1 dato por ciclo para ambos criterios y a la curva correspondiente a 4 datos por ciclo para el criterio gini alcanzan un accuracy del 100%. Aunque vemos que las curvas correspondientes a 4 datos por ciclo presentan mejor desempeño que las curvas de 2 datos por ciclo de forma general podemos decir que también aquí se cumple que el accuracy tiende a disminuir al aumentar los datos extraídos por ciclo.

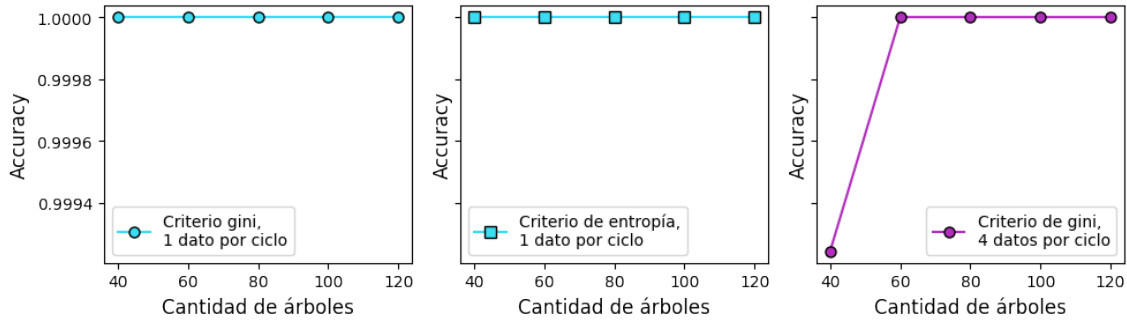


Figura 4.46: Valores de accuracy para la clasificación del estado del enfriador a partir de modelos RF usando datos de varianza.

Accuracias a partir del modelo RF, clasificación: Estado de la válvula

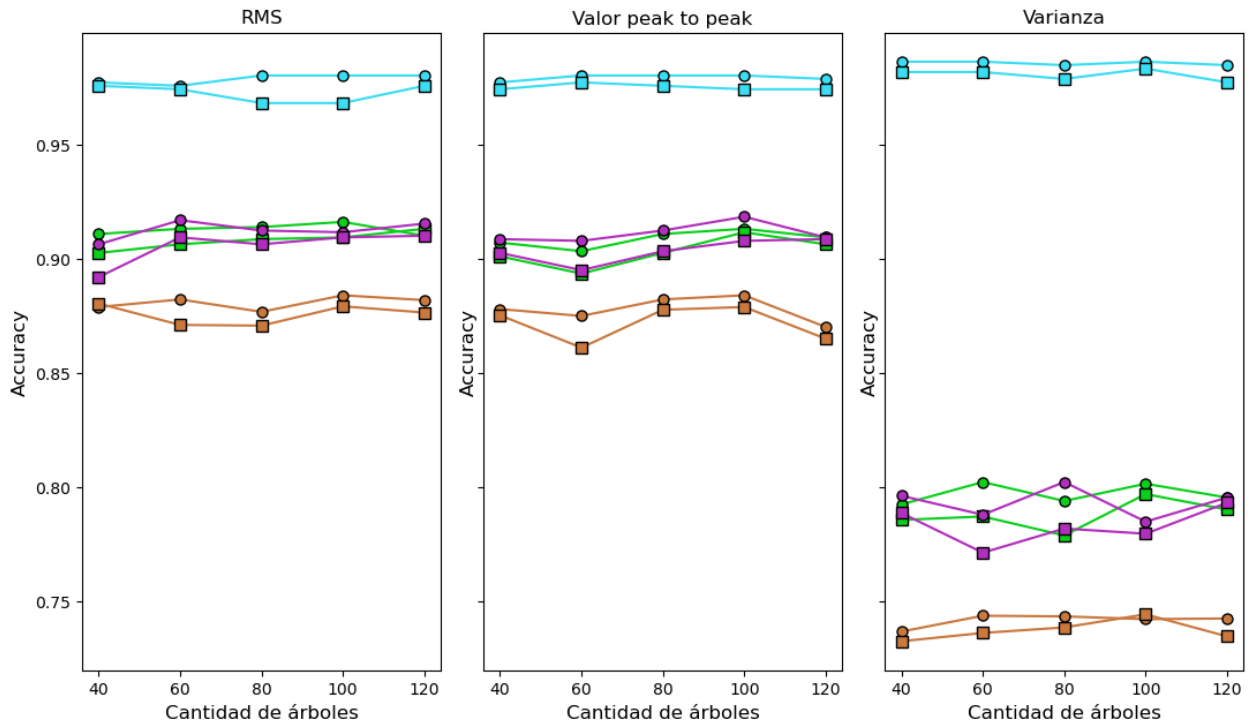


Figura 4.47: Valores de accuracy obtenidos de modelos RF.

Los resultados de la clasificación del estado de la **válvula**, se encuentran en la figura 4.48. De estos gráficos notamos en primer lugar que aparentemente la cantidad de arboles no influye demasiado en el desempeño del modelo, ya que la mayoría de las curvas no presentan un gran cambio en el “eje Y”. También es posible observar que en la mayoría de los casos las curvas de criterio gini (círculos), superan a las curvas de criterio de entropía (cuadrados), sin embargo, el factor que más influye en el desempeño es la cantidad de datos por ciclo. Se destaca que si bien en términos generales la varianza presentó peores resultados en relación al RMS y al valor peak to peak, para las curvas obtenidas usando 1 dato por ciclo la situación fue al revés.

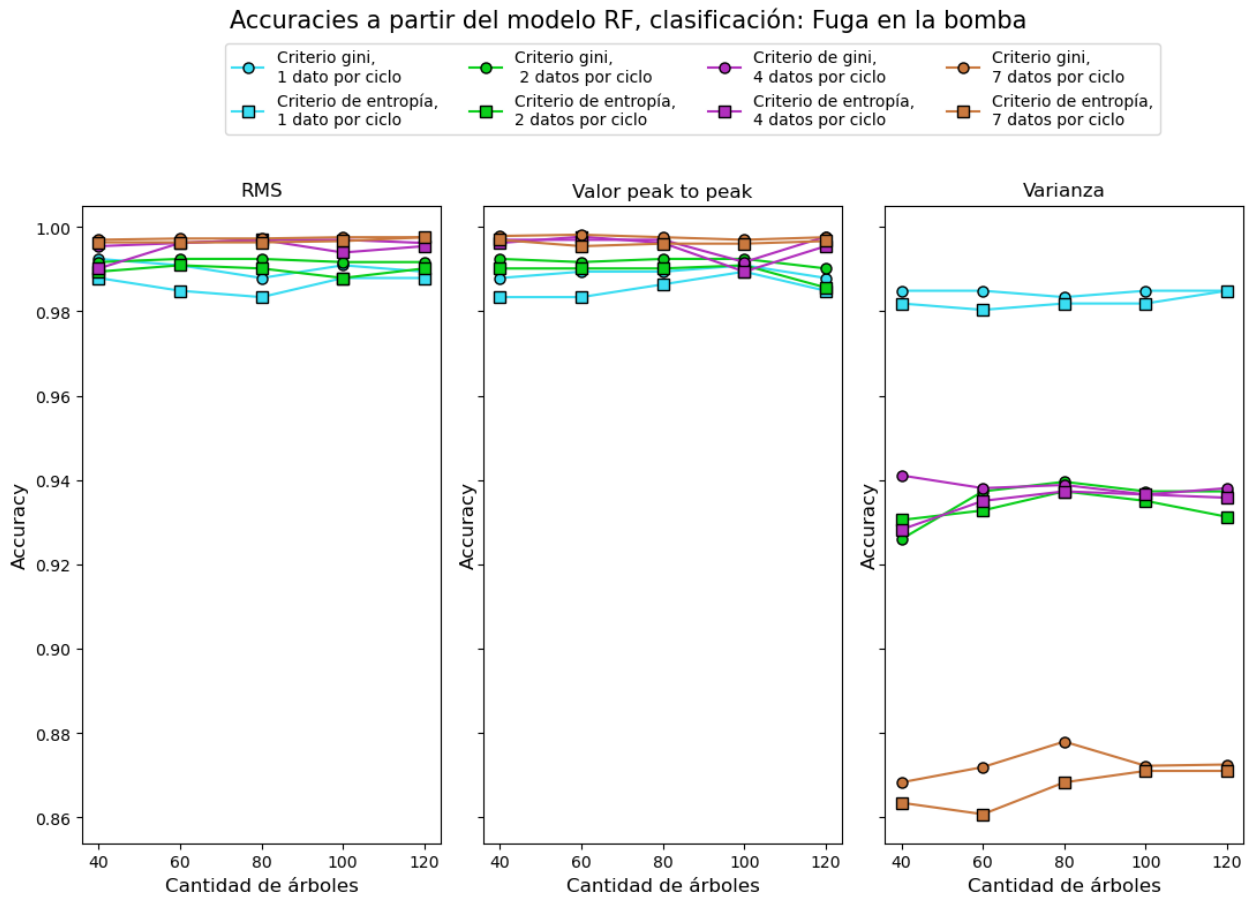


Figura 4.48: Valores de accuracy obtenidos de modelos RF.

Para la clasificación del estado de la **bomba**, cuyos resultados se presentan en las figuras 4.48 y 4.49, se tienen algunas diferencias respecto a las demás clasificaciones. Para la varianza se mantiene el comportamiento general mostrado en la figuras anteriores, sin embargo, al ver la figura 4.49 notamos que para el RMS y el valor peak to peak el valor del accuracy tiende a disminuir al aumentar los datos por ciclo, lo cual es contrario a la mayoría de los resultados vistos.

Accuracias a partir del modelo RF, clasificación: Fuga en la bomba

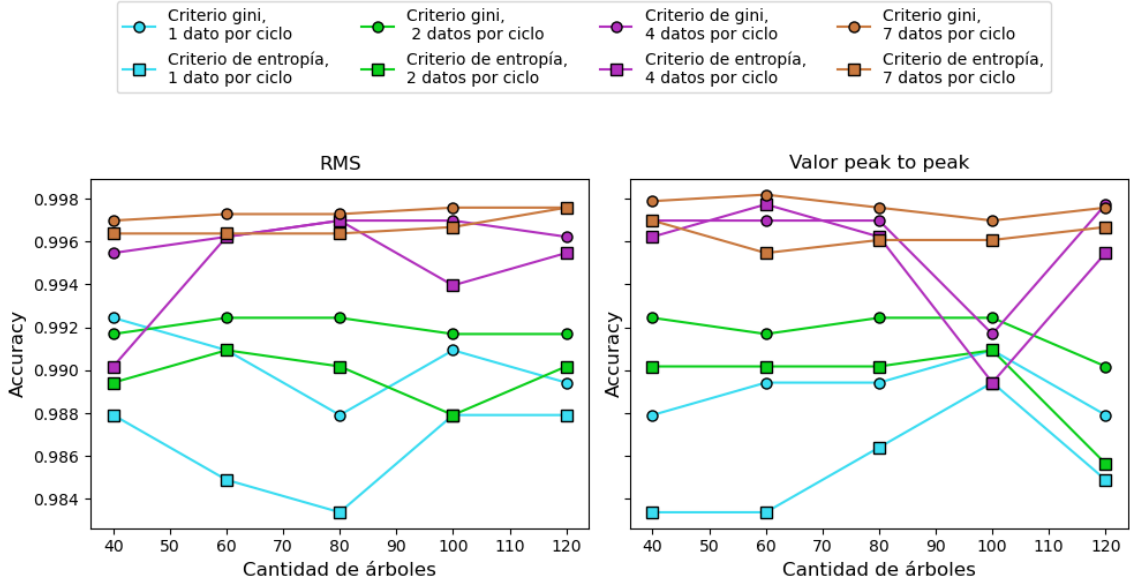


Figura 4.49: Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.48

Accuracias a partir del modelo RF, clasificación: Estado del acumulador

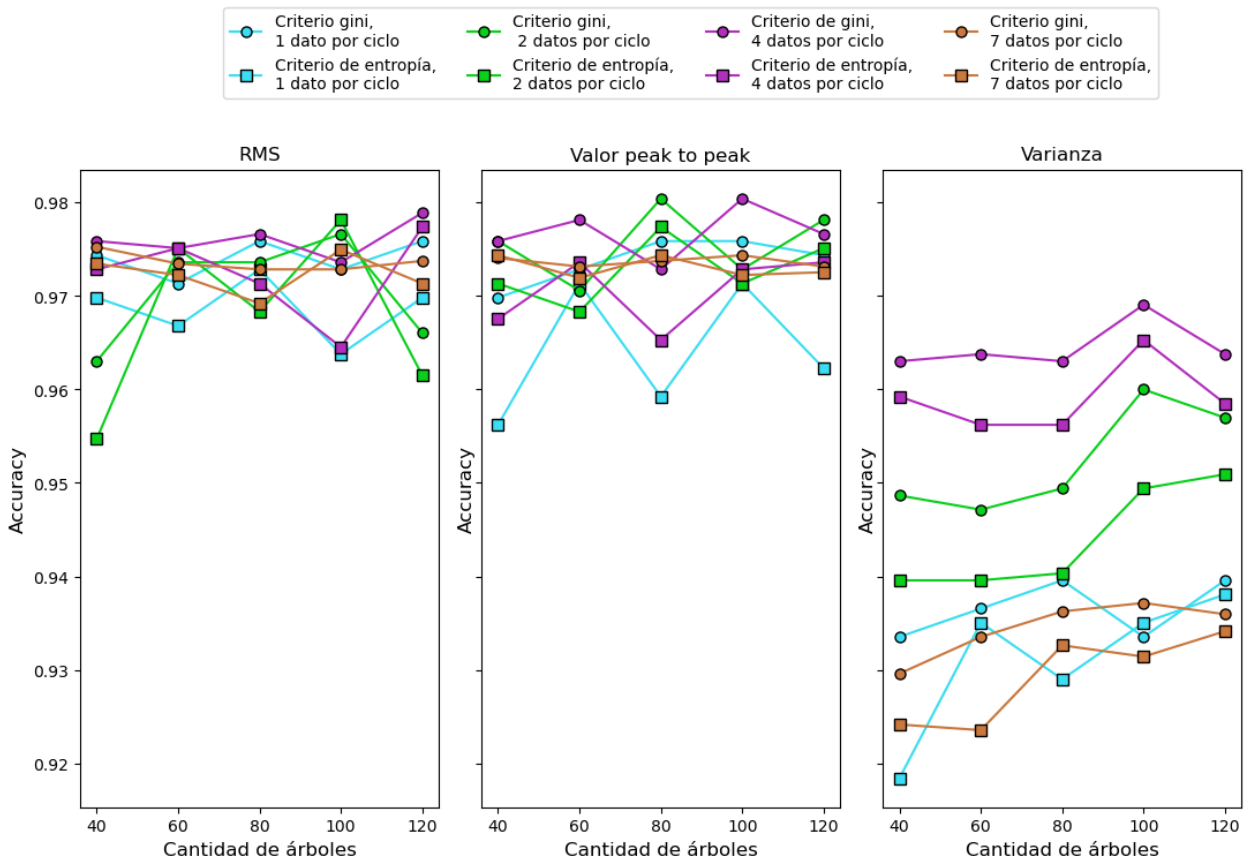


Figura 4.50: Valores de accuracy obtenidos de modelos RF.

Accuracias a partir del modelo RF, clasificación: Estado del acumulador

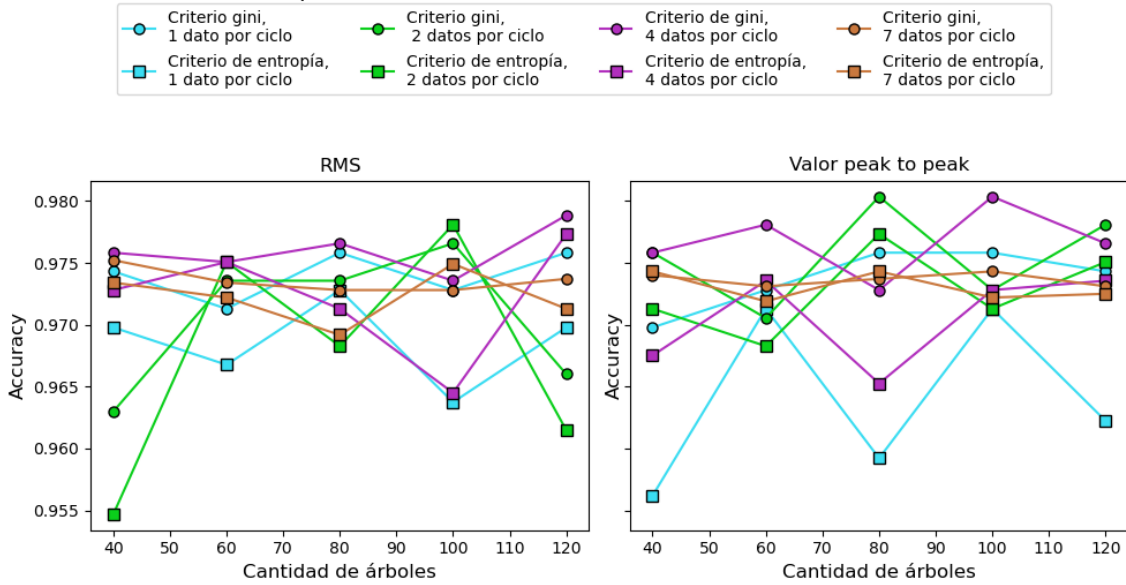


Figura 4.51: Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.50.

Con respecto a los resultados del **acumulador** (figuras 4.50 y 4.51) notamos que la varianza presenta una leve tendencia al alza al aumentar la cantidad de árboles y también que las curvas de 1 dato por ciclo tienen un desempeño mucho más bajo en comparación a los resultados de las clasificaciones anteriores. Para los otros parámetros de tiempo, se ve un comportamiento mucho más irregular donde no es posible notar un patrón de la evolución del accuracy en función de los datos por ciclo, o la cantidad de árboles. Lo único que se podría distinguir como comportamiento general para las curvas de la figura 4.51 es que el criterio gini “le gana” en casi todos los puntos al criterio de entropía de la información.

Accuracies a partir del modelo RF, clasificación: Estabilidad del sistema

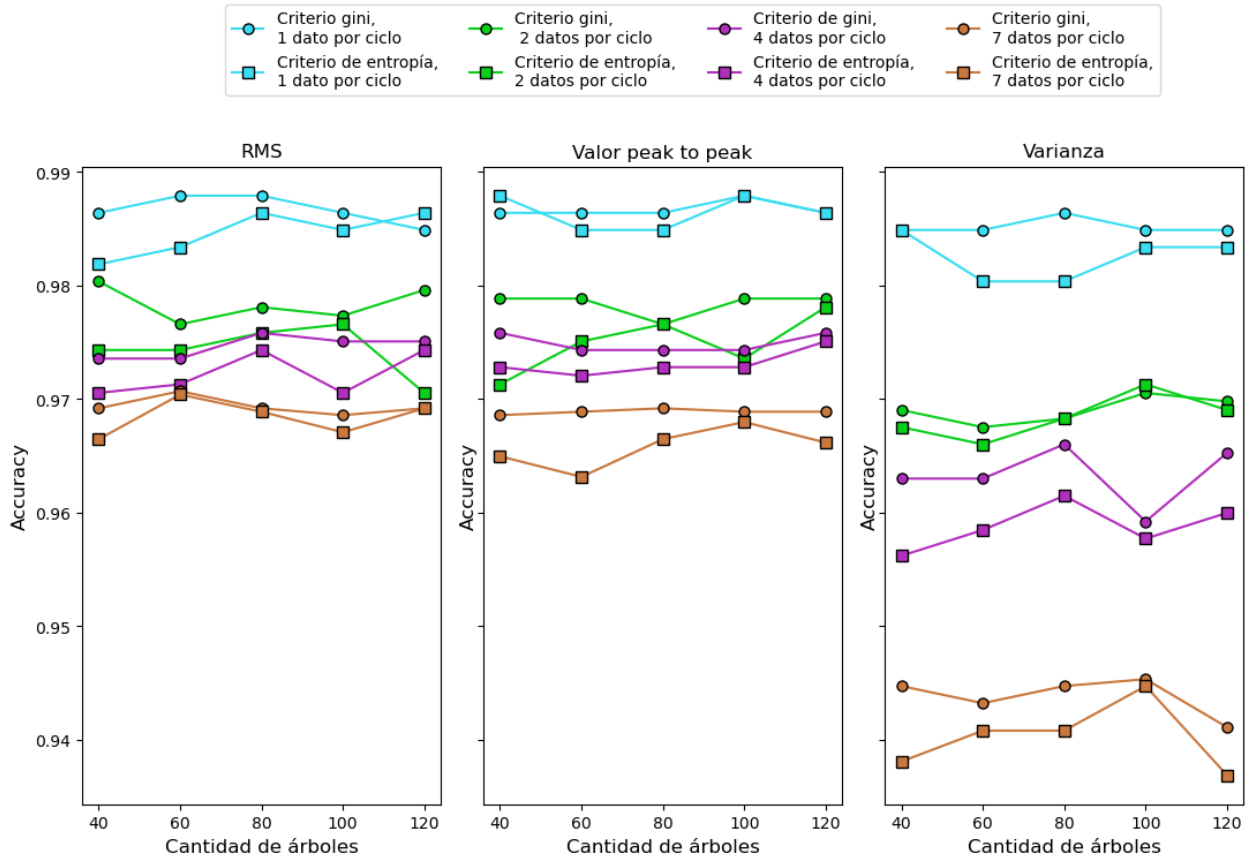


Figura 4.52: Valores de accuracy obtenidos de modelos RF.

En la figura 4.52 correspondiente a la **estabilidad del sistema** se puede ver un comportamiento muy similar a lo mostrado en general por las clasificaciones anteriores, es decir, curvas muy similares entre RMS y valor peak to peak, varianza con desempeño más bajo respecto a los otros parámetros de tiempo, accuracy aproximadamente invariante frente a la cantidad de árboles y un desempeño que disminuye al aumentar los datos por ciclo.

4.5. Resultados a partir de los modelos KNN

Las gráficas con los resultados de los modelos KNN (4.53 a 4.59) son bastante similares a los mostrados en el subcapítulo anterior, ya que muestran la evolución del accuracy al variar la cantidad de vecinos, la cantidad de datos extraídos por ciclo y la diferencia entre usar un peso uniforme y un peso basado en la distancia. Al igual que para los RF, cada color de curva representa un cantidad de datos por ciclo distinta, mientras que para los KNN, las curvas con marcadores circulares representan los modelos con peso uniforme y los curvas con marcadores cuadrados los modelos con peso basado en la distancia.

En la figura 4.53 con los resultados para la clasificación del estado del enfriador se puede ver una vez mas como todos los modelos que usaron los datos provenientes del RMS y del valor peak to peak lograron un accuracy del 100 %. La varianza presenta curvas de resultados que se apegan bastante a lo visto en los RF, salvo por las curvas con 7 datos por ciclo, que presentan un accuracy de un 95 % aproximadamente para 1 y 2 vecinos y luego baja abruptamente a valores entre 90 % y 85 %, lo cual no es un mal resultado, pero como se ve en la figura, en proporción a los demás es notoria su diferencia.

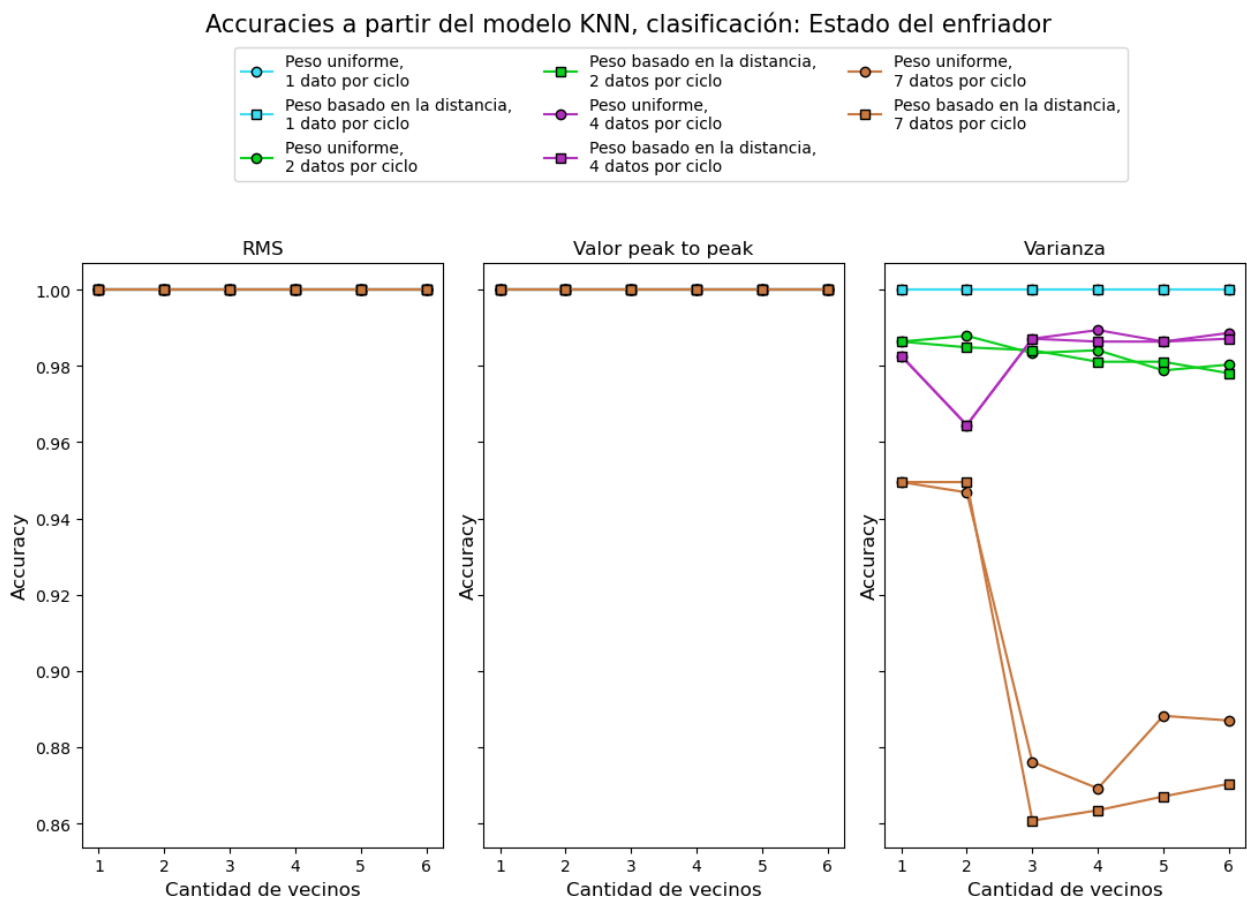


Figura 4.53: Valores de accuracy obtenidos de modelos KNN.

Accuracias a partir del modelo KNN, clasificación: Estado de la válvula

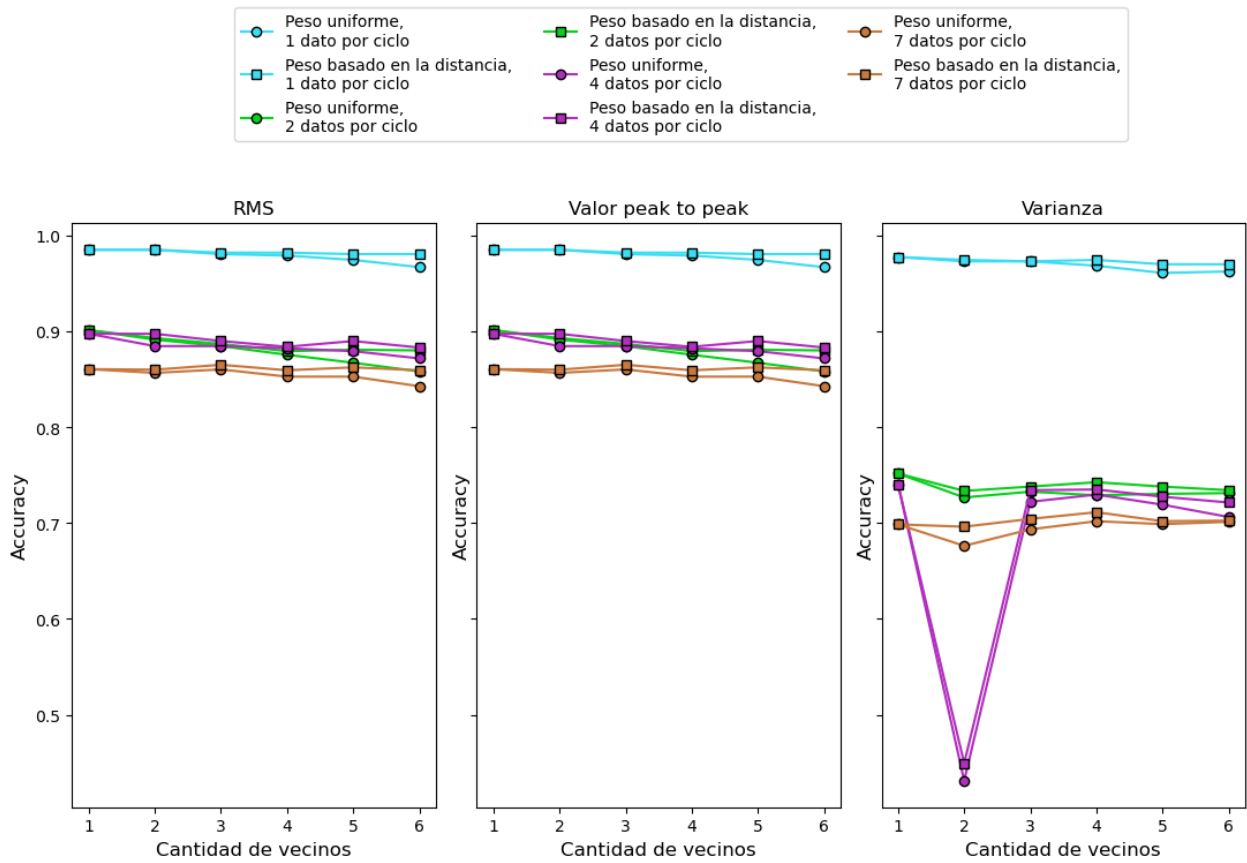


Figura 4.54: Valores de accuracy obtenidos de modelos KNN.

Accuracias a partir del modelo KNN, clasificación: Estado de la válvula

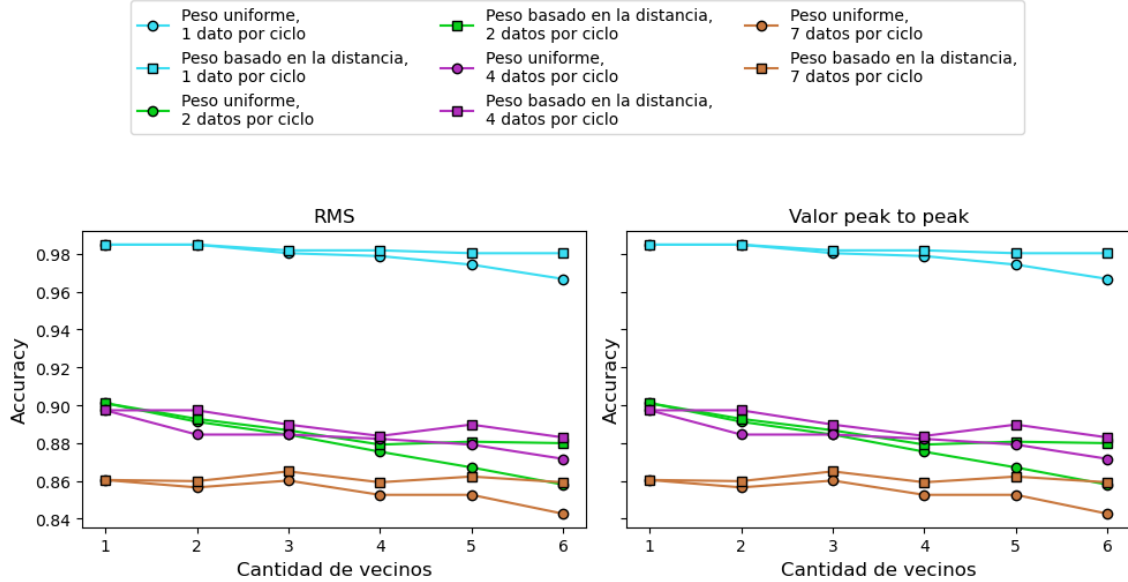


Figura 4.55: Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.54.

Las figuras 4.54 y 4.55 contienen los resultados de la clasificación del estado de la **válvula**. Las curvas del RMS y del valor peak to peak, presentan resultados muy parecidos al mismo caso, pero con los modelos RF, es decir, poco efecto de la cantidad de vecinos y una disminución del accuracy al aumentar la cantidad de datos por ciclo, sin embargo, a diferencia de los modelos RF donde se podía distinguir bien la diferencia entre los criterios, aquí no es fácil notar la diferencia entre peso uniforme y peso basado en la distancia. Por el lado de la varianza se tiene en términos generales lo mismo que para los 2 primeros parámetros de tiempo, excepto por las curvas de 4 datos por ciclo, donde ambas presentan un accuracy muy bajo (cercano al 45 %) para 2 vecinos.

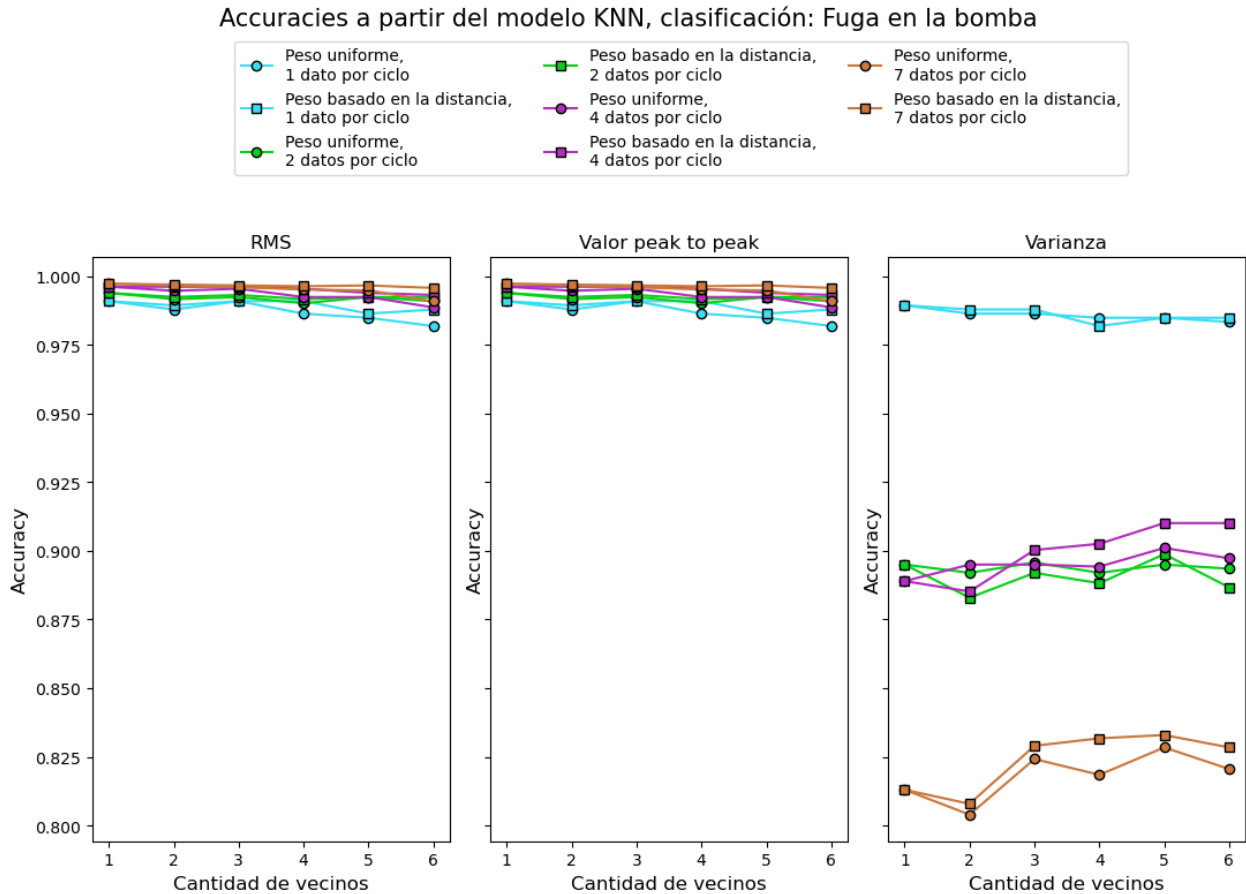


Figura 4.56: Valores de accuracy obtenidos de modelos KNN.

Los resultados de la **bomba** (figuras 4.56 y 4.57) presentan resultados prácticamente iguales a los RF para la misma configuración. Para el RMS y el valor peak to peak no hay cambio notorio con respecto a la cantidad de vecinos, a diferencia de las demás clasificaciones, aquí el accuracy aumenta, al subir la cantidad de datos por ciclo. También se puede ver para los 3 parámetros de tiempo que los modelos que usaron un peso basado en la distancia consiguieron un mejor resultado que los modelos con peso uniforme.

Accuracias a partir del modelo KNN, clasificación: Fuga en la bomba

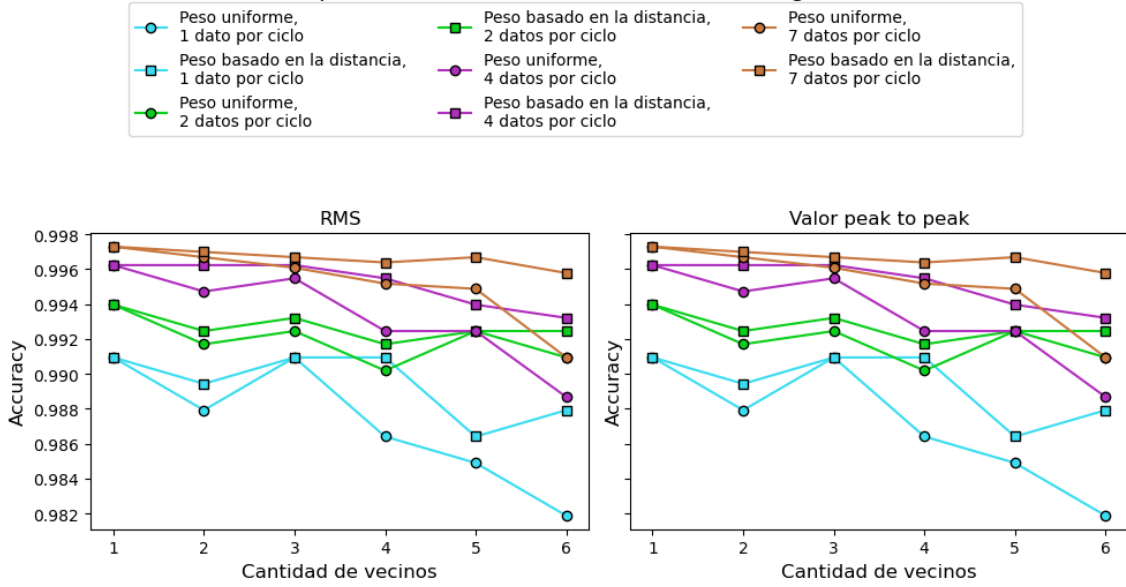


Figura 4.57: Vista ampliada de los parámetros de tiempo RMS y valor peak to peak presentes en la figura 4.57.

Accuracias a partir del modelo KNN, clasificación: Estado del acumulador

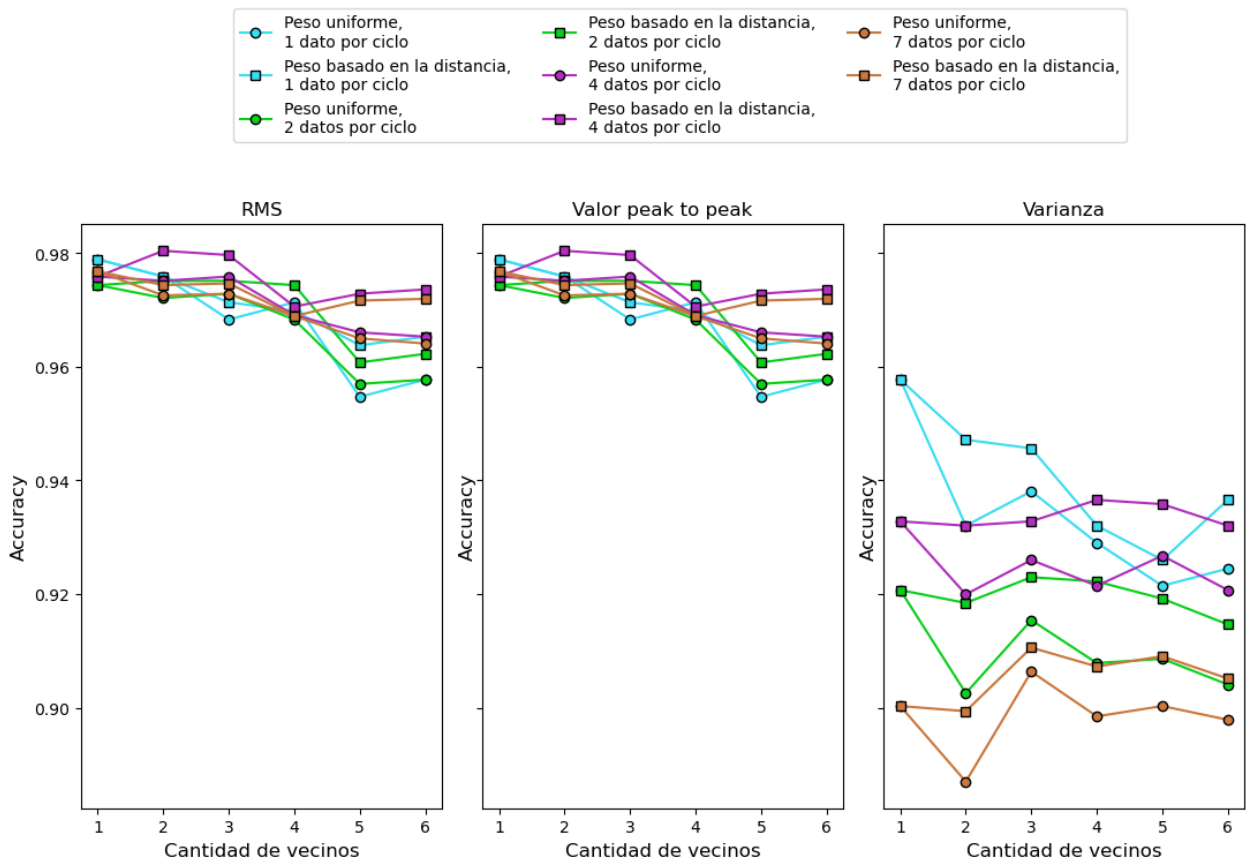


Figura 4.58: Valores de accuracy obtenidos de modelos KNN.

En las figuras 4.58 y 4.59 con los resultados del **acumulador** y de la **estabilidad del sistema** respectivamente, se pueden ver una vez más resultados muy similares a los obtenidos por las mismas clasificaciones para el caso de RF, además de mantener los comportamientos generales presentados por las diferentes clasificaciones.

Accuracias a partir del modelo KNN, clasificación: Estabilidad del sistema

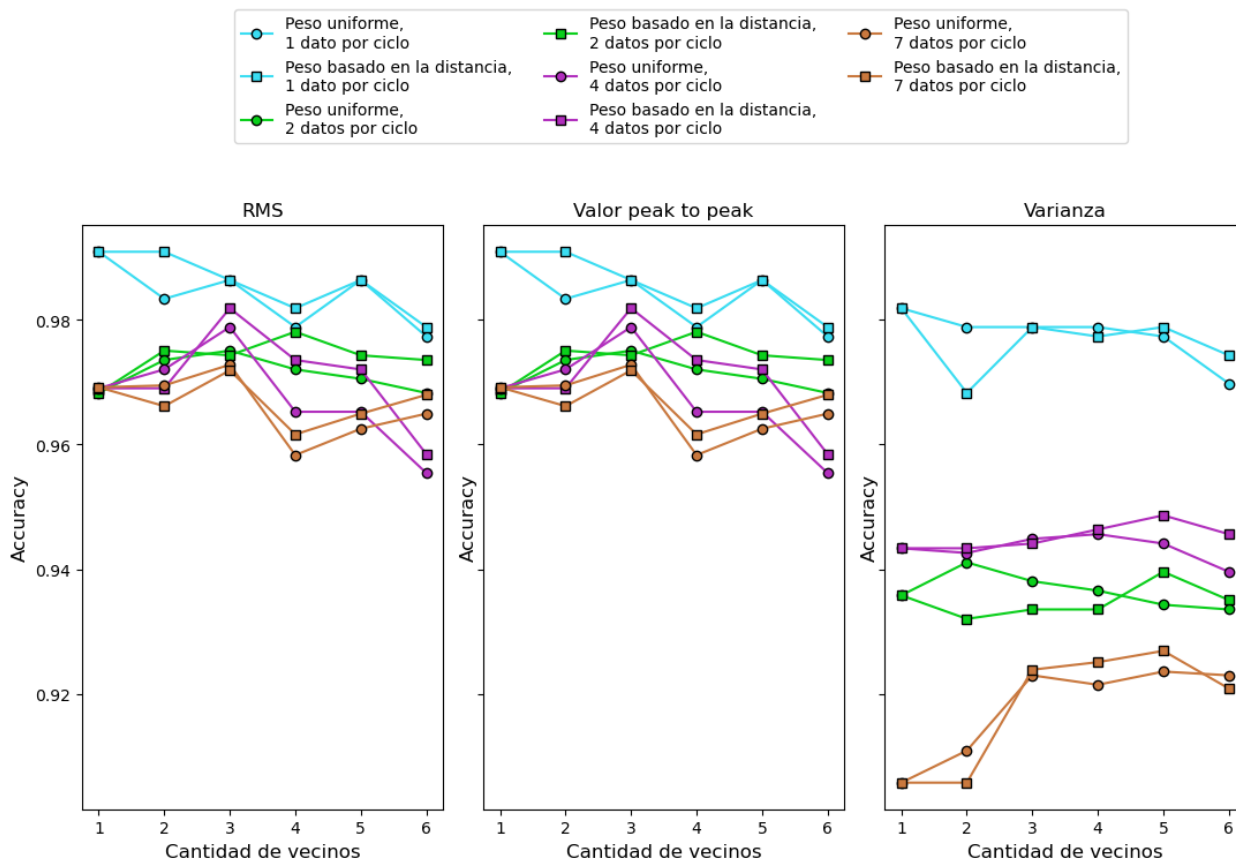


Figura 4.59: Valores de accuracy obtenidos de modelos KNN.

4.6. Resultados a partir de los modelos SVM

En este subcapítulo se muestran los resultados obtenidos a partir de los modelos SVM. En primer lugar se estudia el desempeño de los modelos al variar los datos por ciclo y los kernels, para luego, en base estos primeros resultados fijar las mejores configuraciones de kernel y datos por ciclo para luego estudiar el efecto de los parámetros “C” y “gamma”.

4.6.1. Accuracy en función del largo de ventana para diferentes kernels

Las figuras 4.60 a 4.60 nos muestran el accuracy obtenido por los modelos SVM en función de los datos por ciclo, para los kernels rbf, sigmoid y lineal. Solo se muestran los gráficos de los parámetros de tiempo RMS y varianza, ya que como se ha visto anteriormente el valor peak to peak y la media presentan resultados muy similares al RMS. En cuanto al comportamiento general de las curvas, podemos ver que en todos los casos el kernel rbf es que obtiene los mejores resultados. Por otra parte, para efecto de los datos por ciclos no se puede decir a priori que se muestra una tendencia al alza o a la baja al aumentar los datos por ciclo, pero si se puede ver que para las 5 clasificaciones la mejor accuracy fue alcanzada con 1 dato por ciclo.

En cuanto a comportamientos específicos, tenemos que el **enfriador** (figura 4.60) logra resultados muy buenos tanto para el RMS como para la varianza. Para la **válvula** (figura 4.61), notamos en primer lugar que sus rangos de accuracy son mas bajos que los del enfriador, y que además, en esta combinación, la varianza superó al RMS. Para la **bomba** (figura 4.62) se puede apreciar que el RMS supera a la varianza y además que el gráfico de la varianza es el único donde se nota una tendencia a la baja clara al aumentar los datos por ciclo. El **acumulador** (figura 4.63) se apega mucho al comportamiento general, y lo único destacable es que algunas de sus curvas fueron las que obtuvieron el peor resultado entre todos los modelos SVM. Para la **estabilidad del sistema** se puede destacar que aparentemente, su curva rbf fue la con mejor desempeño después de la curva rbf del enfriador.

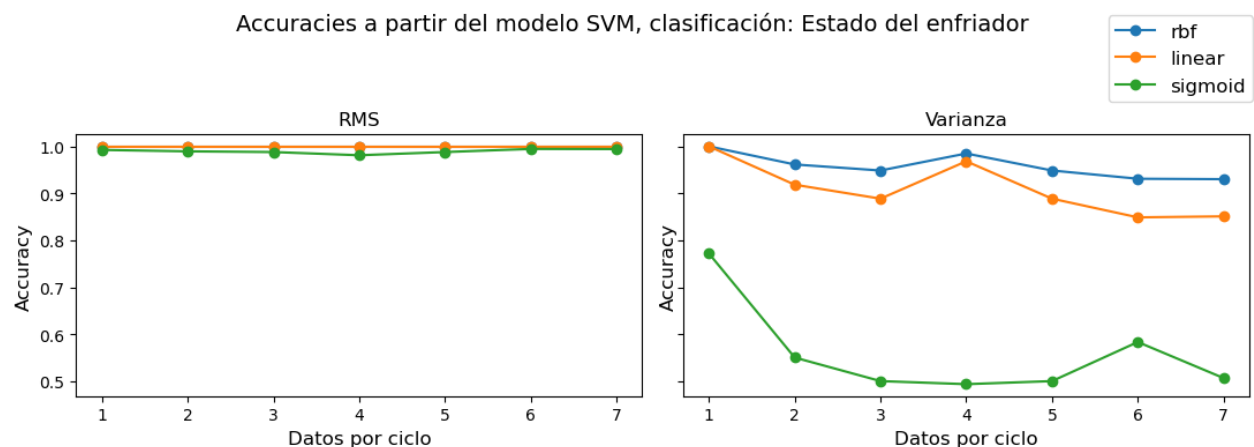


Figura 4.60: Valores de accuracy obtenidos de modelos SVM.

Accuracias a partir del modelo SVM, clasificación: Estado de la válvula

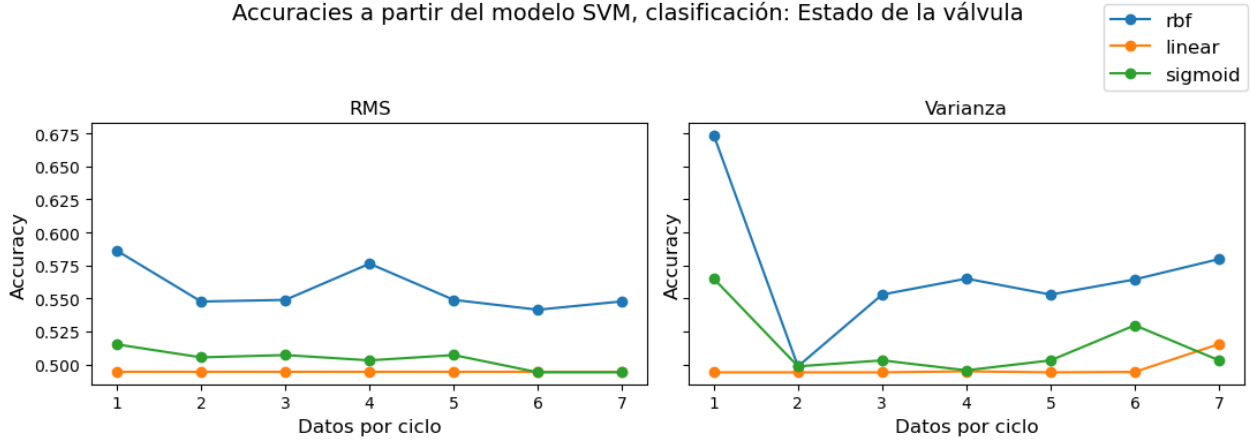


Figura 4.61: Valores de accuracy obtenidos de modelos SVM.

Accuracias a partir del modelo SVM, clasificación: Fuga en la bomba

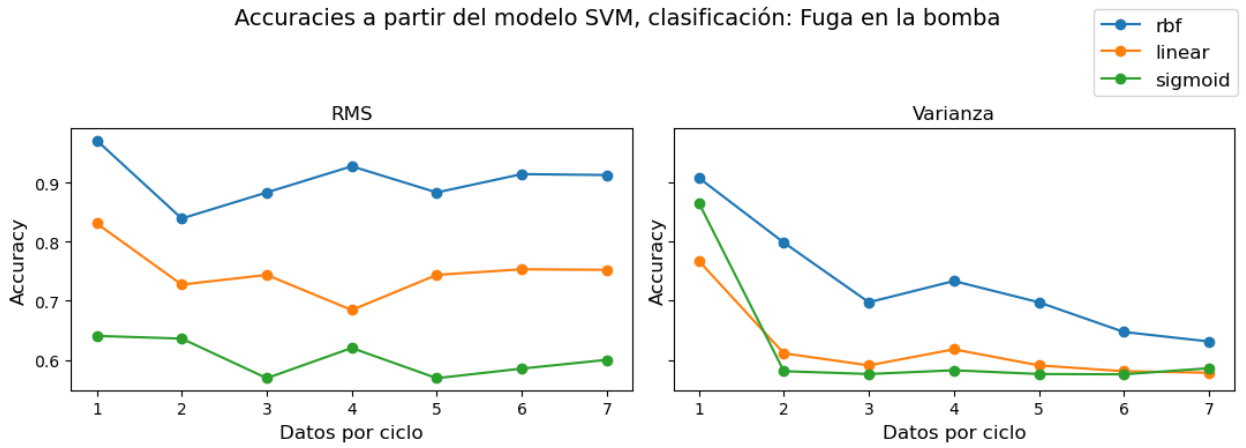


Figura 4.62: Valores de accuracy obtenidos de modelos SVM.

Accuracias a partir del modelo SVM, clasificación: Estado del acumulador

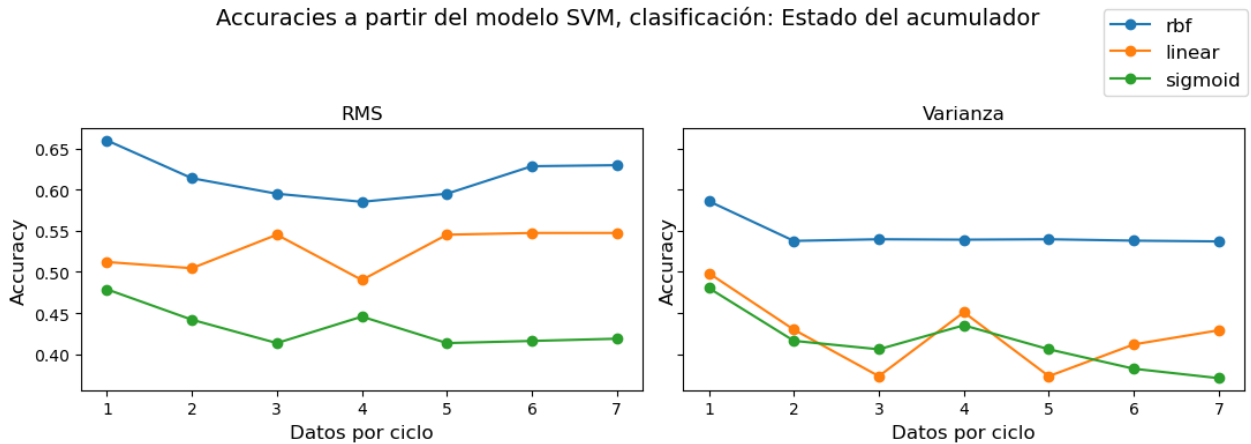


Figura 4.63: Valores de accuracy obtenidos de modelos SVM.

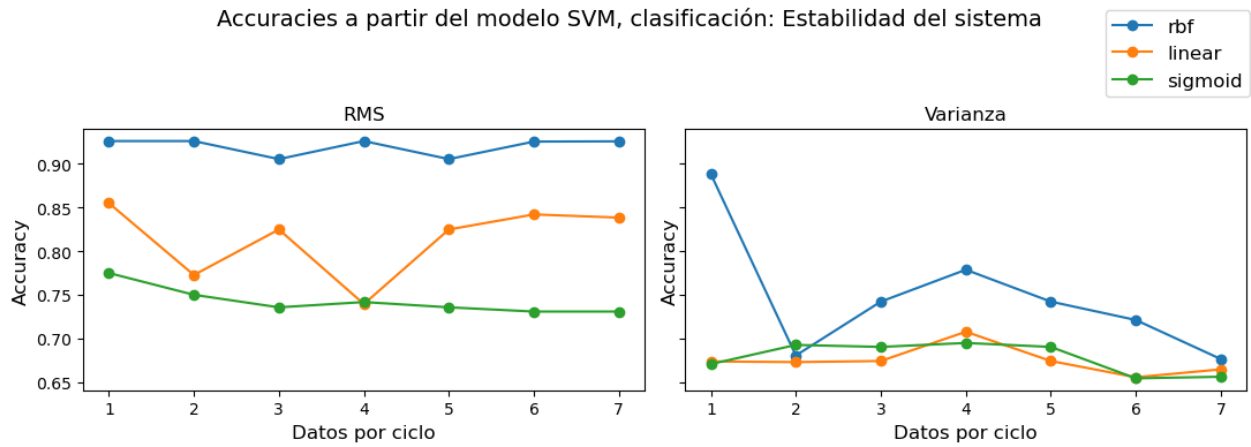


Figura 4.64: Valores de accuracy obtenidos de modelos SVM.

4.6.2. Heatmaps con la variación de los parámetros C y gamma

A diferencia de los modelos vistos anteriormente, que solo eran sensibles a 1 o 2 dos hiperparámetros, el desempeño del SVM también cambia mucho al variar los parámetros “C” y “gamma”. En las figuras 4.65 a 4.69 se muestran los valores de accuracy al variar los parámetros “C” y “gamma”, fijando un kernel rbf y 1 dato por ciclo extraído por los parámetros de tiempo.

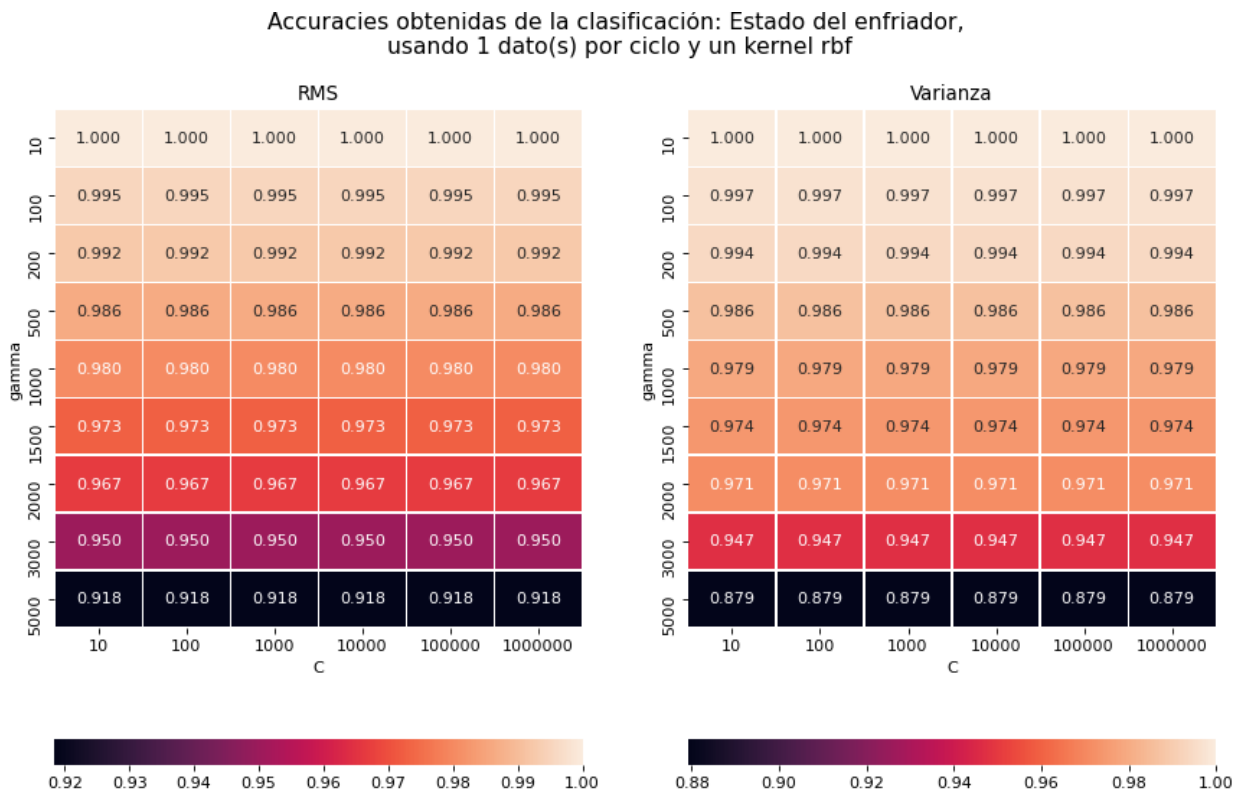


Figura 4.65: Heatmap con la variación del accuracy en función de los parámetros c y gamma.

Para el **enfriador** (figura 4.65) vemos que el accuracy no varia con respecto al parámetro C mientras que disminuye al aumentar gamma. Se puede ver también que tanto el RMS como la varianza lograron resultados muy buenos llegando a un accuracy igual 100 %.

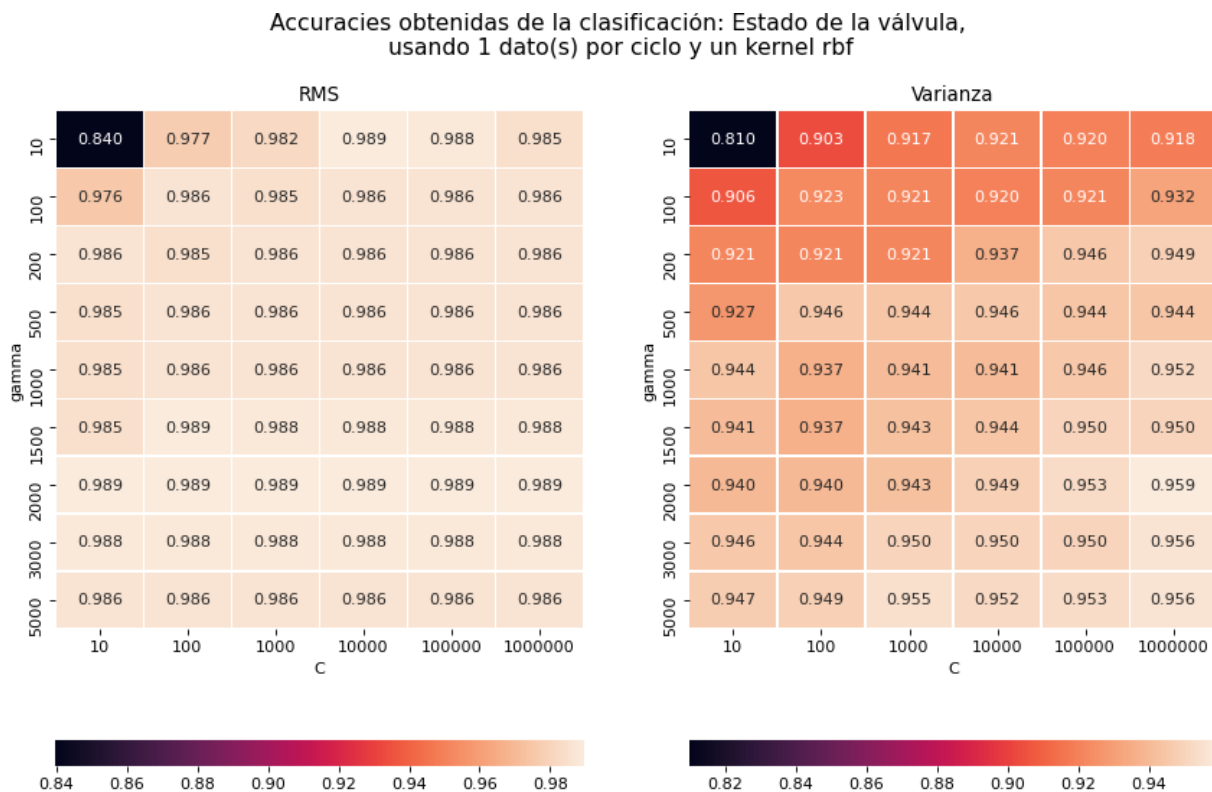


Figura 4.66: Heatmap con la variación del accuracy en función de los parámetros c y gamma.

Para el caso de la **válvula** (figura 4.66) se muestra una tendencia a aumentar el accuracy a medida que se aumenta C y gamma, lo que es mucho más notorio para la varianza que para el RMS. Nuevamente ambos parámetros de tiempo entregaron muy buenos y similares resultados

Accuracies obtenidas de la clasificación: Fuga en la bomba,
usando 1 dato(s) por ciclo y un kernel rbf

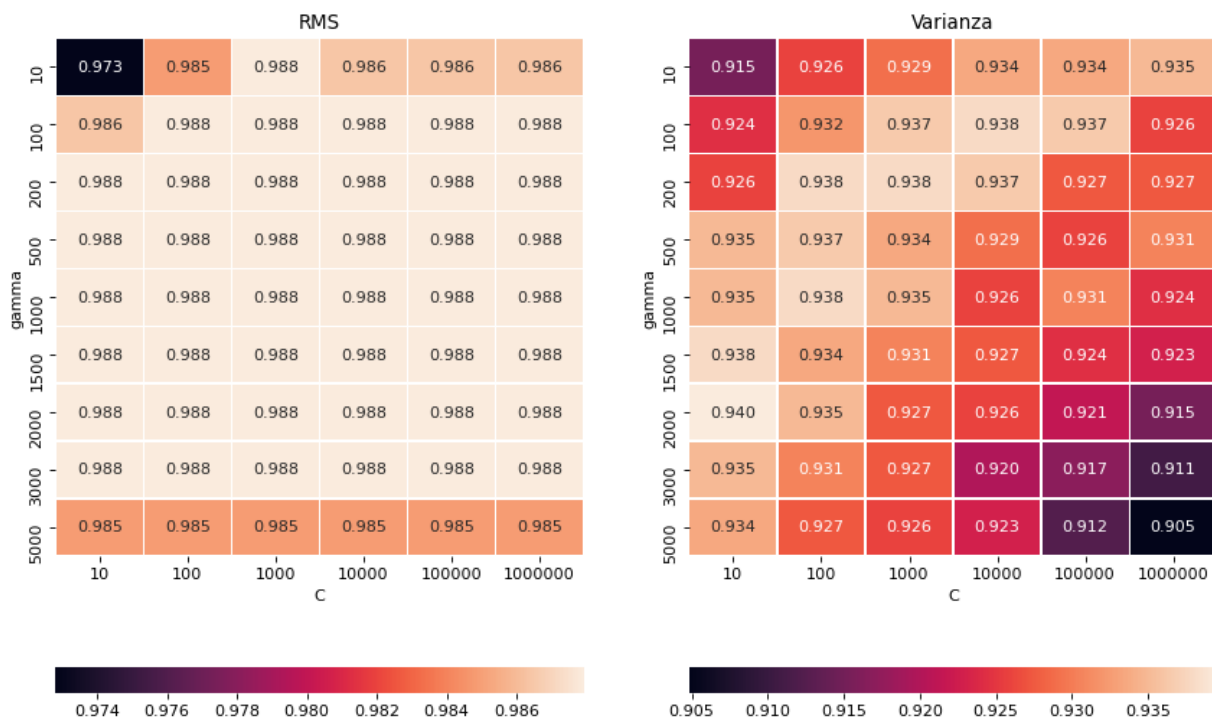


Figura 4.67: Heatmap con la variación del accuracy en función de los parámetros c y γ .

Para la **bomba** (figura 4.67) se ve que el accuracy correspondiente al RMS no varía mucho frente a ninguno de los 2 parámetros. Para los resultados de la varianza, se ve que los mejores valores de accuracy se alcanzan cerca de $C=1000$ y $\gamma=200$, reduciendo su desempeño al aumentar o disminuir los parámetros.

Accuracias obtenidas de la clasificación: Estado del acumulador,
usando 1 dato(s) por ciclo y un kernel rbf

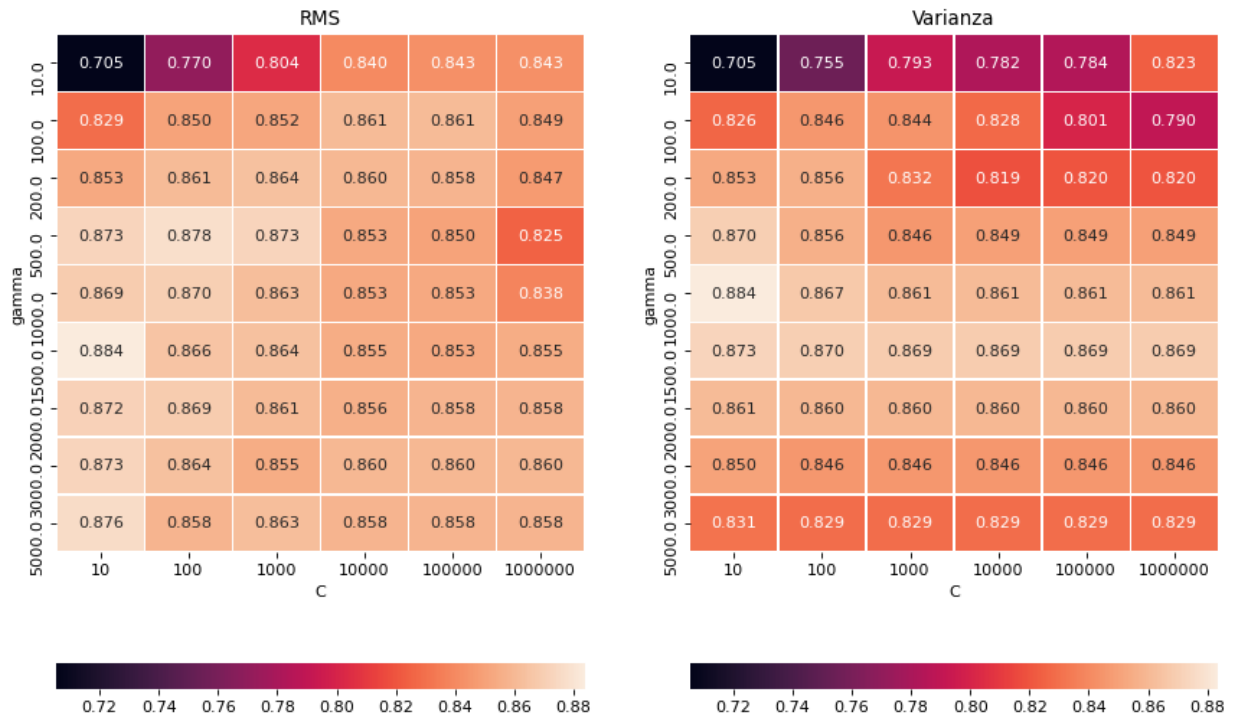


Figura 4.68: Heatmap con la variación del accuracy en función de los parámetros c y gamma.

Para el **acumulador** (figura 4.68) se ve un comportamiento similar al del enfriador, ya que el accuracy no varía mucho con C y gamma es el parámetro que más determina su desempeño. Se destaca que nuevamente de manera similar al enfriador, ambos heatmaps presentan resultados muy similares.

Accuracias obtenidas de la clasificación: Estabilidad del sistema, usando 1 dato(s) por ciclo y un kernel rbf

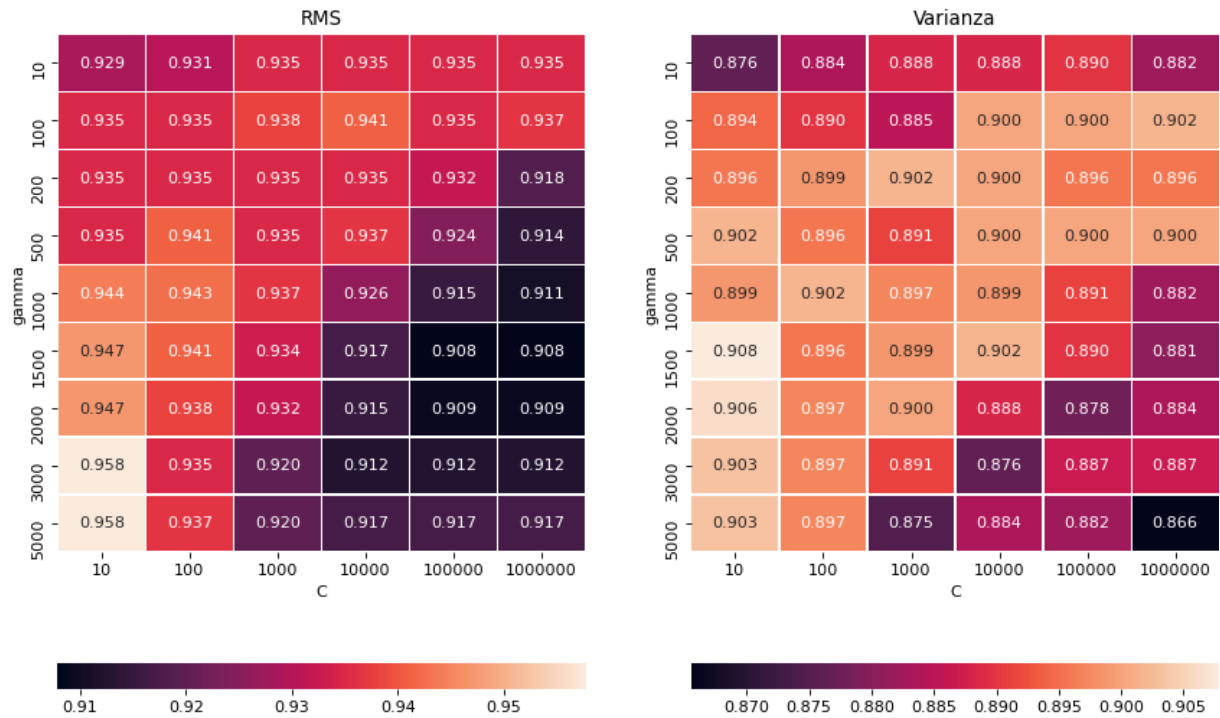


Figura 4.69: Heatmap con la variación del accuracy en función de los parámetros c y γ .

Para la clasificación de la **estabilidad del sistema** se ve que en ambos parámetros de tiempo el desempeño tiende a bajar para valores grandes de C y γ . En este caso ambos heatmaps no parecen mucho, ya que los valores del gráfico de la varianza no muestran un patrón tan claro como el del RMS.

De los heatmaps en general, se destaca que se pudieron conseguir valores muy altos de accuracy para todas las combinaciones especialmente para resultados logrados a partir de datos de varianza, los cuales como vimos anteriormente, en la mayoría de los casos lograron resultados significativamente menores al resto de los parámetros de tiempo.

4.7. Resultados a partir de los modelos NN

4.7.1. Accuracy en función del largo de ventana para diferentes funciones de activación

De forma similar a los modelos SVM, en las figuras 4.70 a 4.74 se muestra cómo cambia el valor del accuracy en función de la cantidad de datos por ciclo para diferentes funciones de activación. Si bien en general se muestran comportamientos bastante distintos, se puede ver que la mayoría de las curvas mejora su desempeño al aumentar la cantidad de datos por ciclo, así como también el hecho de que las curvas obtenidas a partir de modelos que usaron la función de activación sigmoide tuvieron peores desempeños en comparación a las funciones ReLU y TanH.

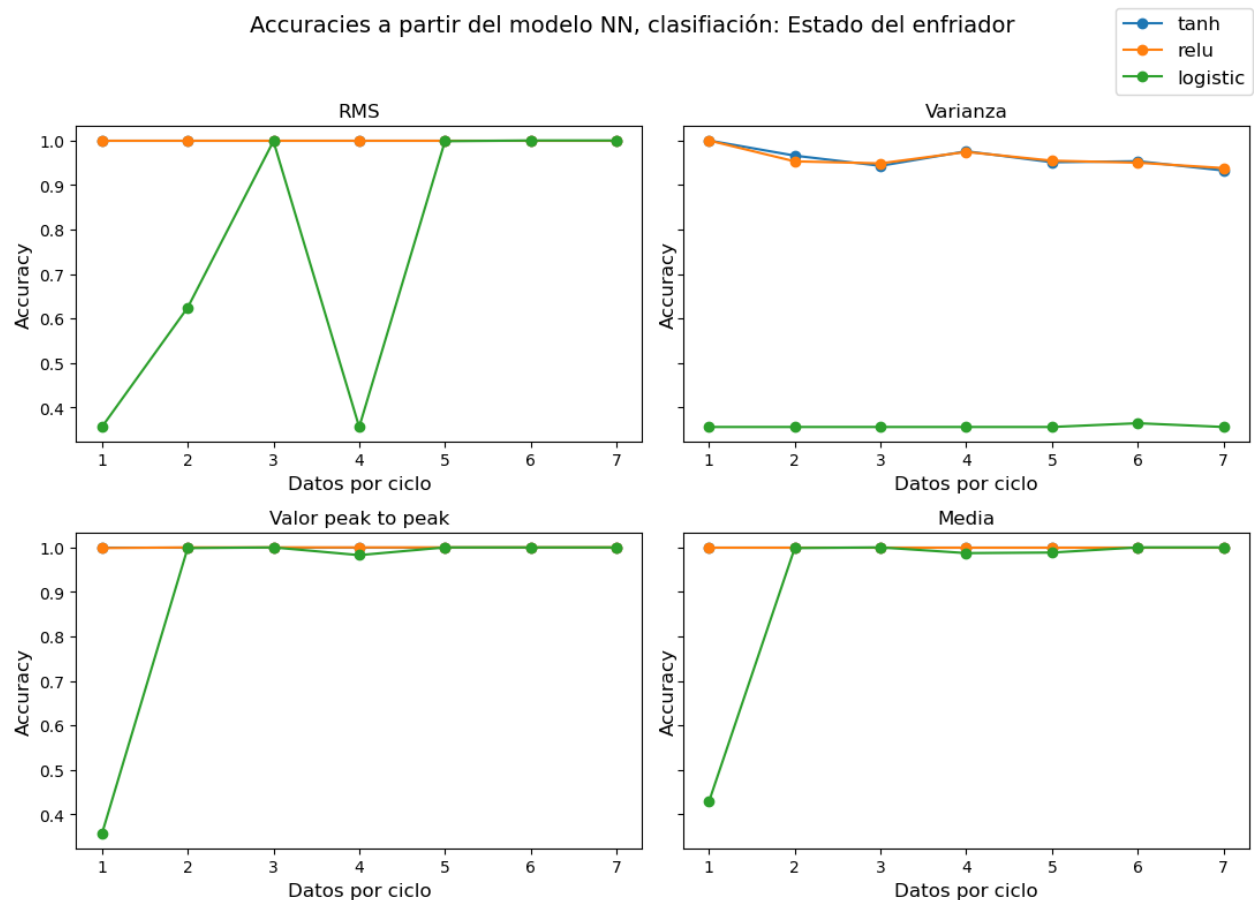


Figura 4.70: Accuracy obtenido a partir de modelos MLP.

Para los gráficos del **enfriador** (figura 4.72) como fue recurrente a lo largo de este capítulo, se puede ver que muchas de las combinaciones alcanzan o se acercan bastante a un accuracy del 100 %.

Accuracias a partir del modelo NN, clasificación: Estado de la válvula

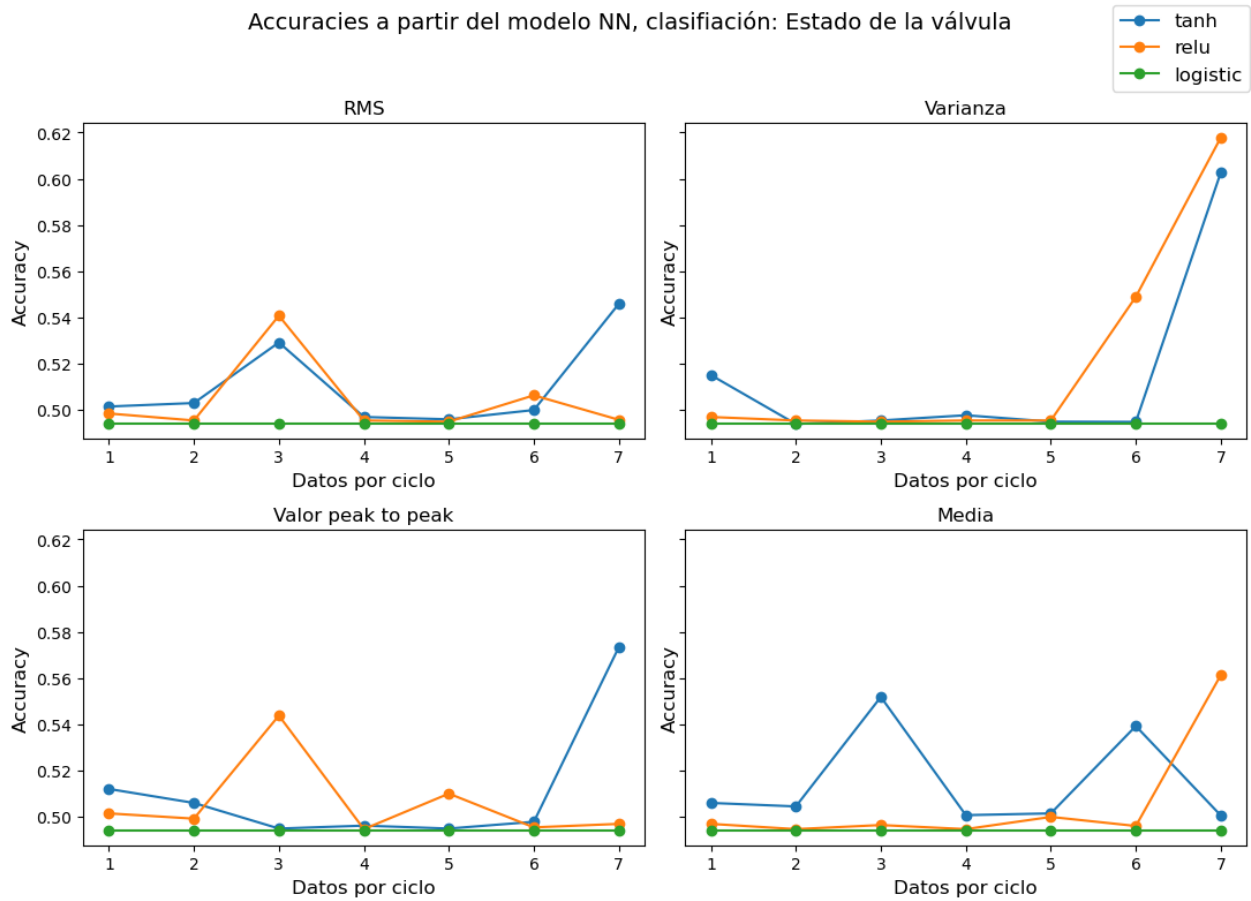


Figura 4.71: Accuracy obtenido a partir de modelos MLP.

Para el caso de la **válvula** (figura 4.71) vemos que en ninguno de los parámetros de tiempo se presenta un patrón claro con respecto a a la evolución del accuracy en función de los datos por ciclo, aunque se destaca que aparentemente la curva ReLU logra mejores desempeños que la curva de la TanH.

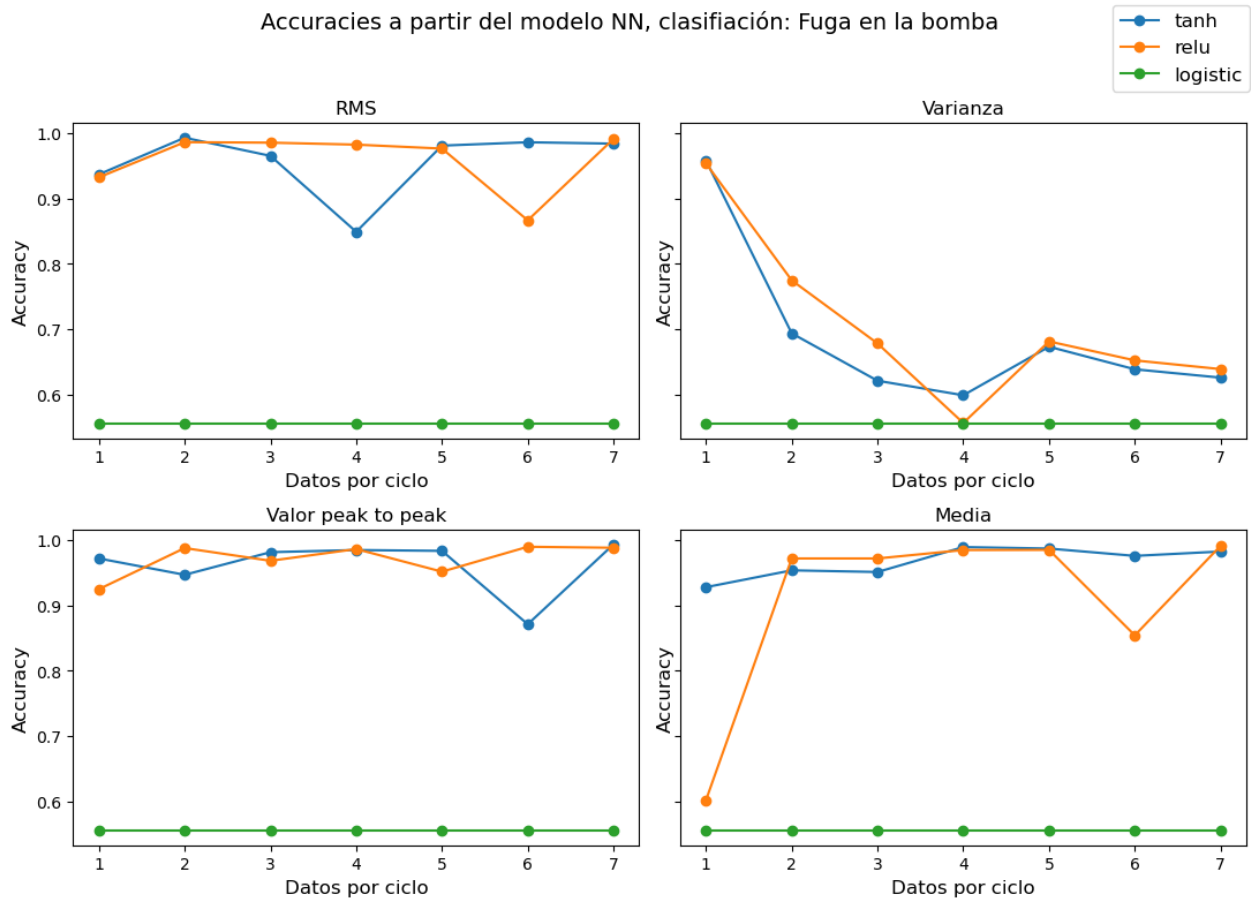


Figura 4.72: Accuracy obtenido a partir de modelos MLP.

Para los gráficos de la figura 4.72, pertenecientes a la **bomba**, tampoco es claro el comportamiento que se tiene al variar los datos por ciclo. Las curvas ReLU y TanH. no presentan algún patrón que se repita en los distintos parámetros de tiempo, sin embargo la curva sigmoide entrega un valor de accuracy constante a lo largo de los datos por ciclo para cada parámetro de tiempo.

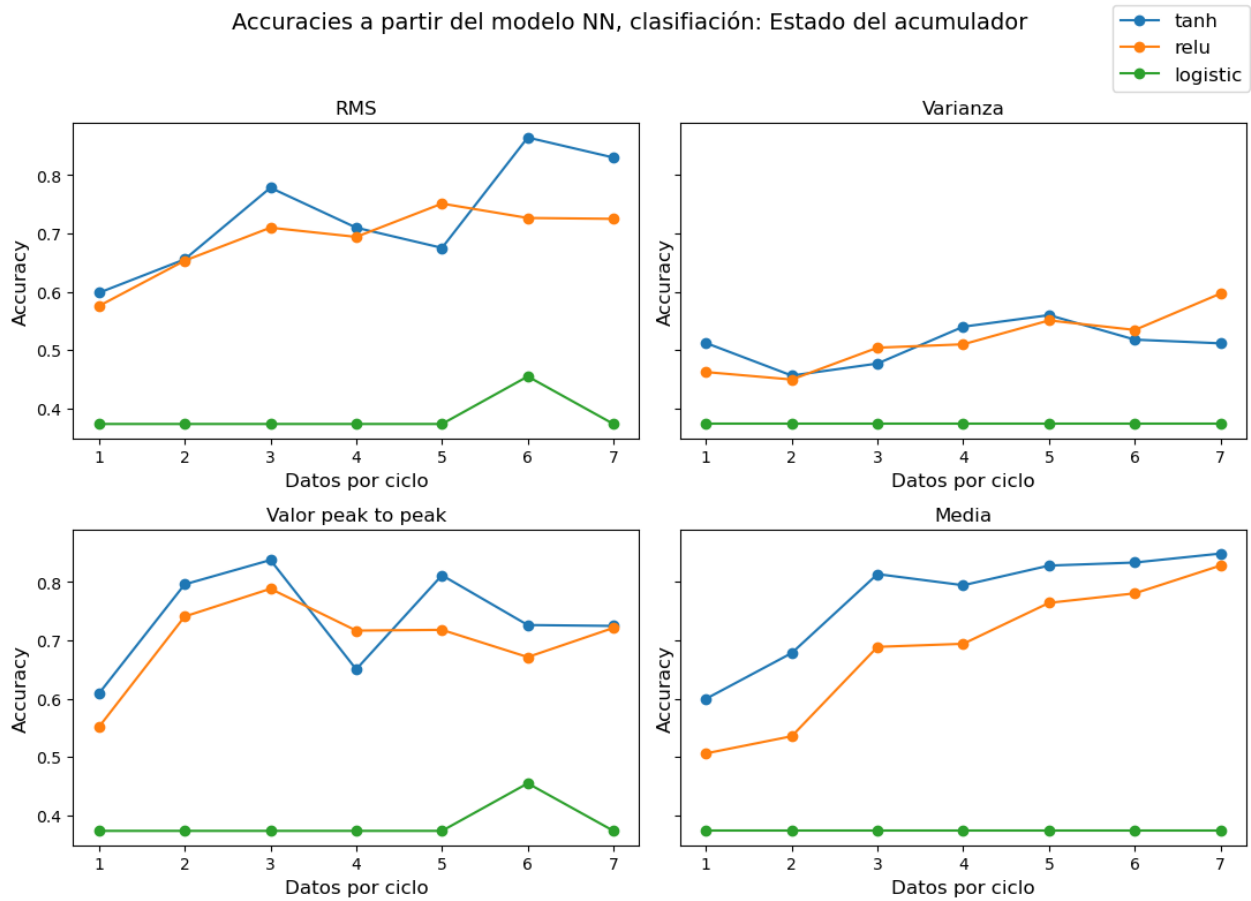


Figura 4.73: Accuracy obtenido a partir de modelos MLP.

En el caso del **acumulador** (figura 4.73) a diferencia de todo el resto de gráficos de los modelos MLP, si es posible distinguir patrones generales en las curvas. En primer lugar se tiene que para todos los parámetros de tiempo, el valor del accuracy tiende al alza al aumentar la cantidad de datos por ciclo, y en segundo lugar, la curva de la TanH. presenta una forma muy similar a la curva ReLU en los 4 gráficos.

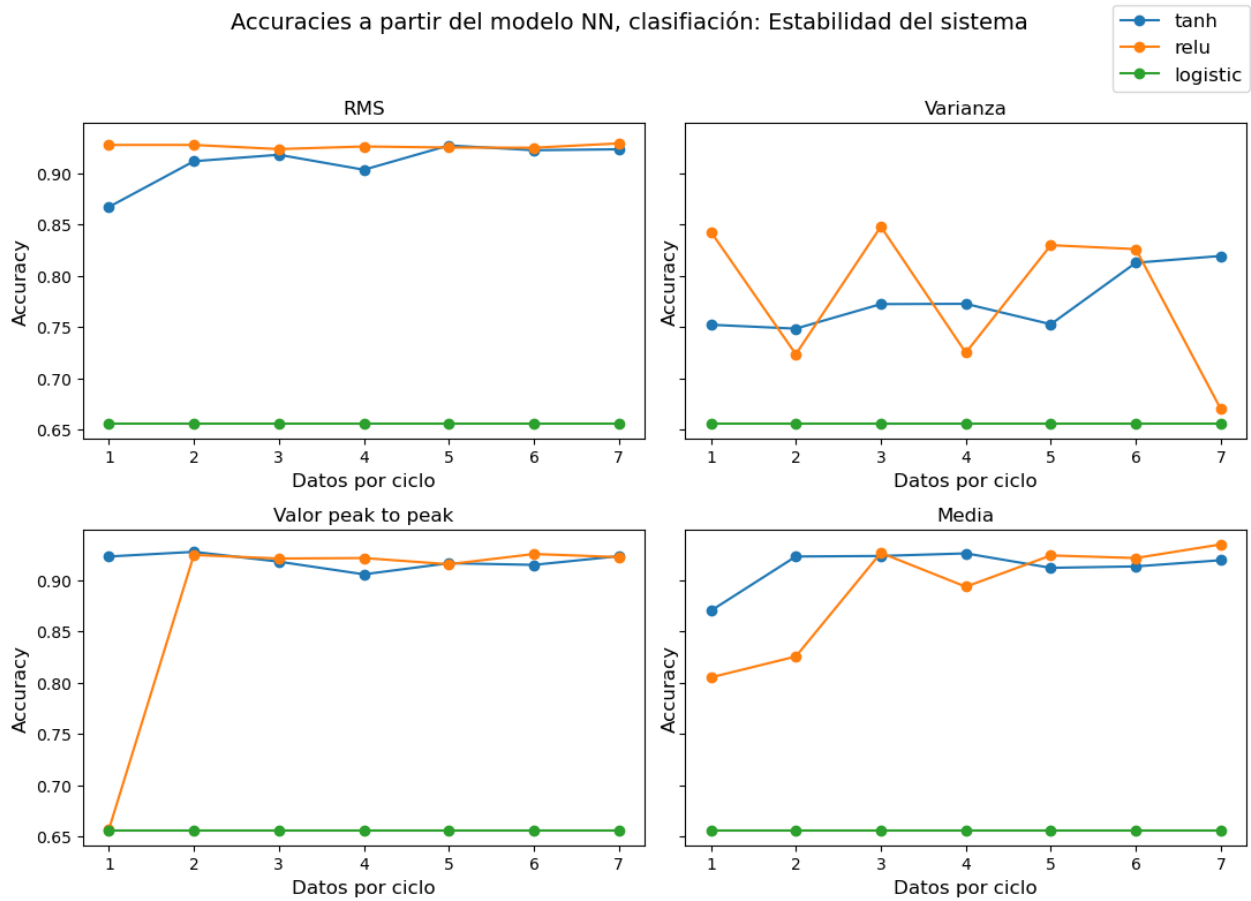


Figura 4.74: Accuracy obtenido a partir de modelos MLP.

Para la **estabilidad del sistema** (figura 4.74) se puede destacar que nuevamente la curva sigmoide presento un comportamiento constante y por otra parte que la curva ReLU muestra un mejor desempeño que la curva de la TanH.

4.7.2. Heatmaps con la variación del learning rate y el parámetro alpha

Debido a que los modelos MLP no presentaron algún patrón regular en cuanto a la evolución del accuracy respecto a la cantidad de vecinos, en cuanto a los parámetros de tiempo, o en cuanto al desempeño de las funciones de activación, para cada clasificación se fijó la combinación de parámetro de tiempo-cantidad de datos que maximizara su accuracy (ver tabla 4.1 y luego se variaron los parámetros alpha y learning rate para las funciones de activación ReLU y TanH.

Tabla 4.1: Combinaciones parámetro de tiempo-cantidad de datos por ciclo para probar los parámetros alpha y learning rate

Clasificación	Parámetro de tiempo	Datos por ciclo
Enfriador	RMS	1
Válvula	Media	7
Bomba	Valor peak to peak	7
Acumulador	RMS	6
Sistema	Media	7

En el heatmap correspondiente al **enfriador** (figura 4.75) se puede ver que no hay un cambio notorio al variar alpha y el learning rate, sin embargo se logran resultados bastante altos de accuracy. Para la **válvula** (figura 4.76), por un lado se puede apreciar que el learning rate no afecta mucho su desempeño, y por otro se puede ver que esta clasificación alcanza sus valores más altos de accuracy en un rango más acotado del parámetro alpha. Pasando a la **bomba** (figura 4.77) notamos nuevamente que el accuracy prácticamente no varia con respecto al learning rate, mientras que los mejores valores se alcanzan para alpha cercano a 0.01. En la figura 4.78 con los gráficos del acumulador, podemos notar un leve aumento en la sensibilidad hacia el learning rate, aunque ahora el mejor valor se alcanza en alpha=0.01 para la TanH. y alpha=0.001 para la ReLU. Finalizando con la **estabilidad del sistema** (figura 4.79) podemos observar que como se fue repitiendo en todas las clasificaciones anteriores, el desempeño de los modelos no se ve muy afectado por el learning rate y los mejores valores de accuracy se consiguen en alpha=0.001. Se destaca que en el caso de la válvula, la bomba y la estabilidad del sistema, la ReLU superó a la TanH., en el enfriador presentaron el mismo desempeño y en el acumulador la función TanH supero a la función ReLU.

Accuracias obtenidas de la clasificación: Estado del enfriador, usando 1 dato(s) por ciclo y RMS

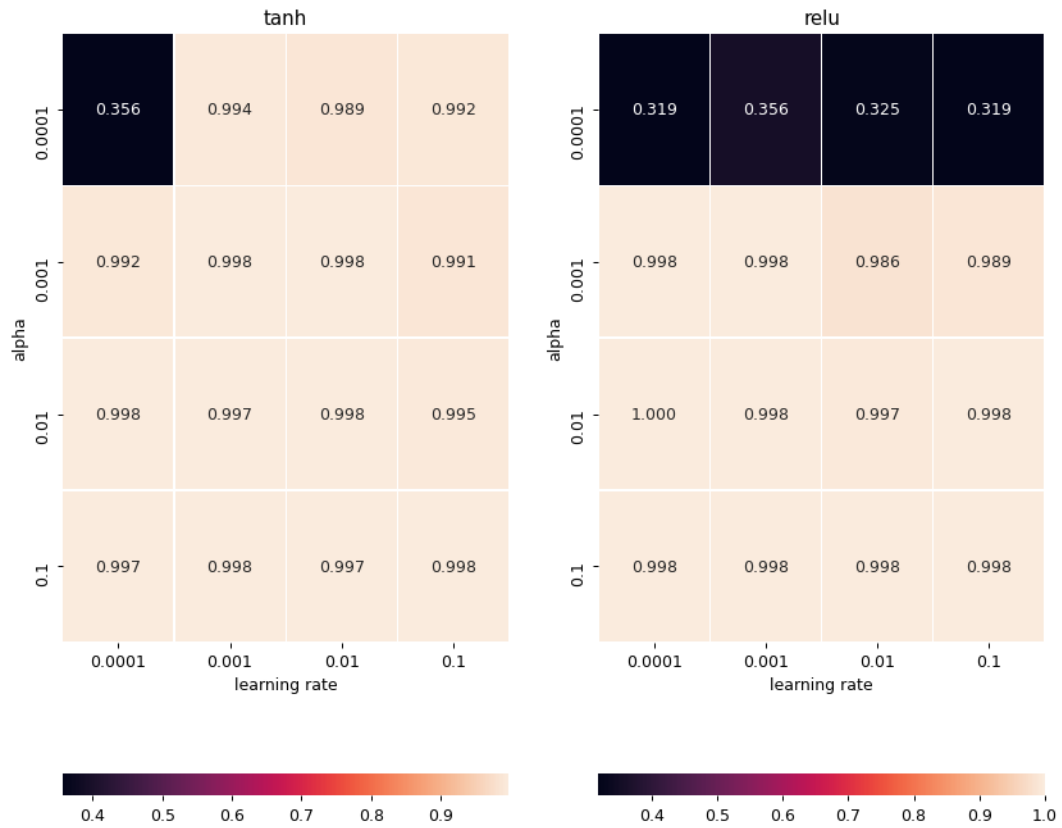


Figura 4.75: Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.

Accuracies obtenidas de la clasificación: Estado de la válvula, usando 7 dato(s) por ciclo y Variance

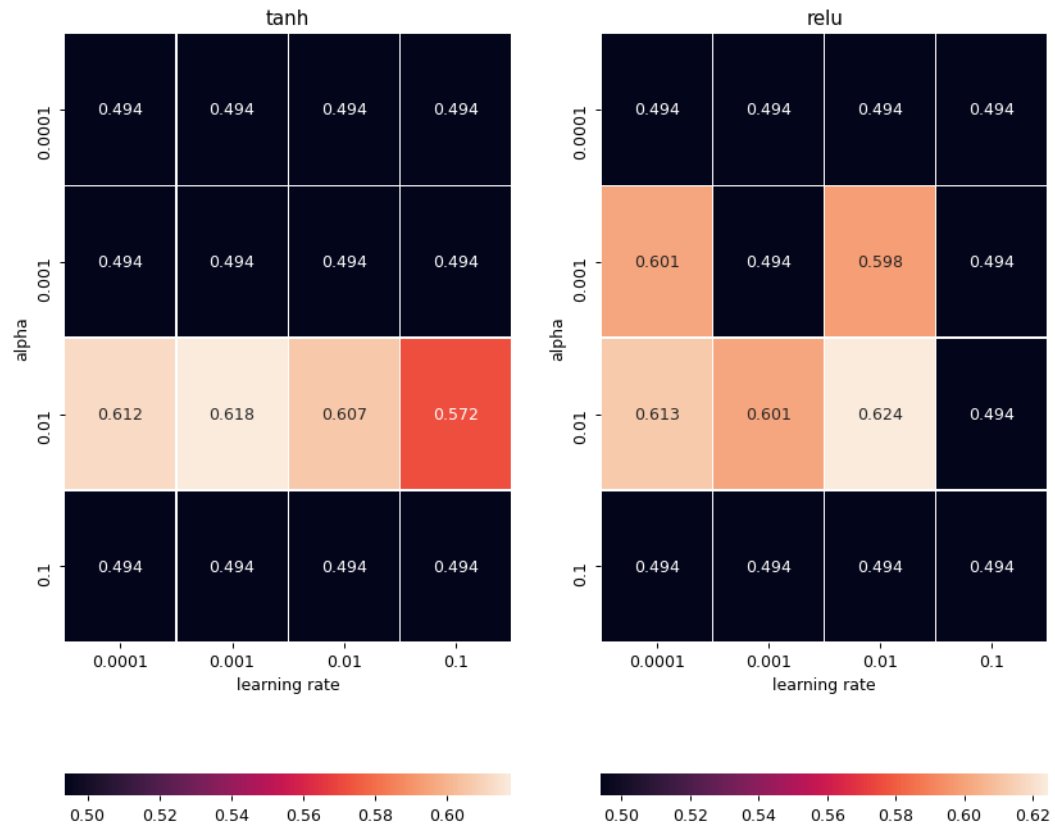


Figura 4.76: Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.

Accuracies obtenidas de la clasificación: Fuga en la bomba,
usando 7 dato(s) por ciclo y P2P

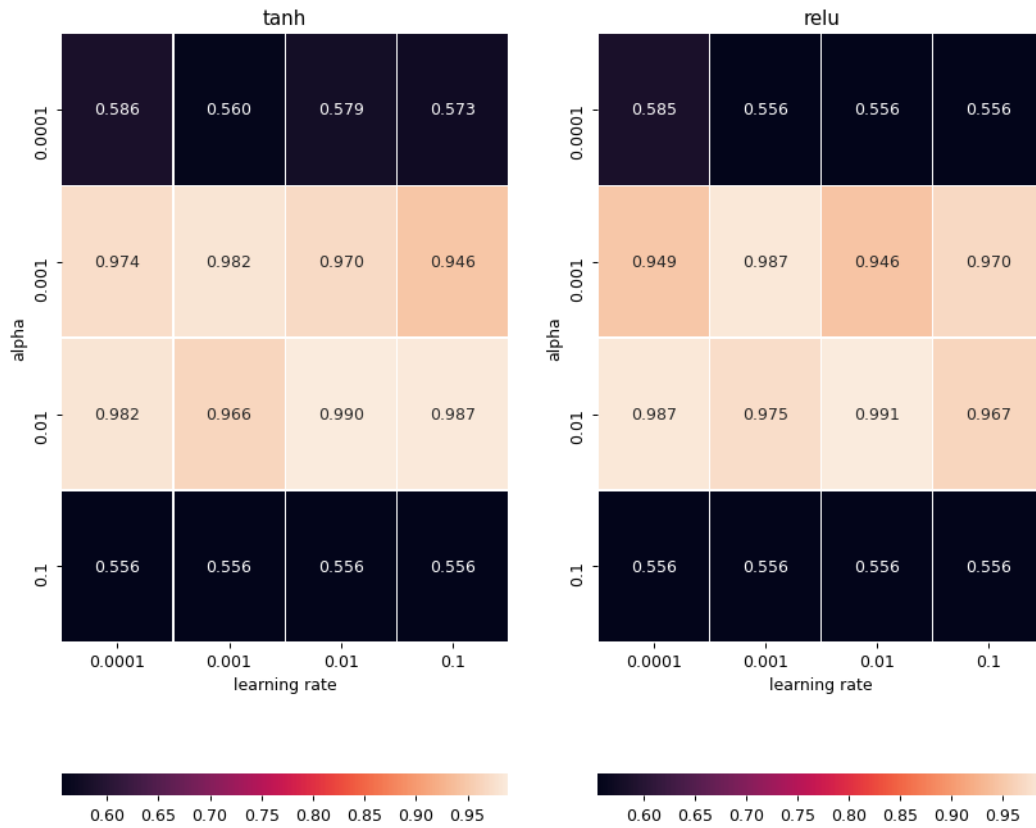


Figura 4.77: Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.

Accuracies obtenidas de la clasificación: Estado del acumulador, usando 6 dato(s) por ciclo y RMS

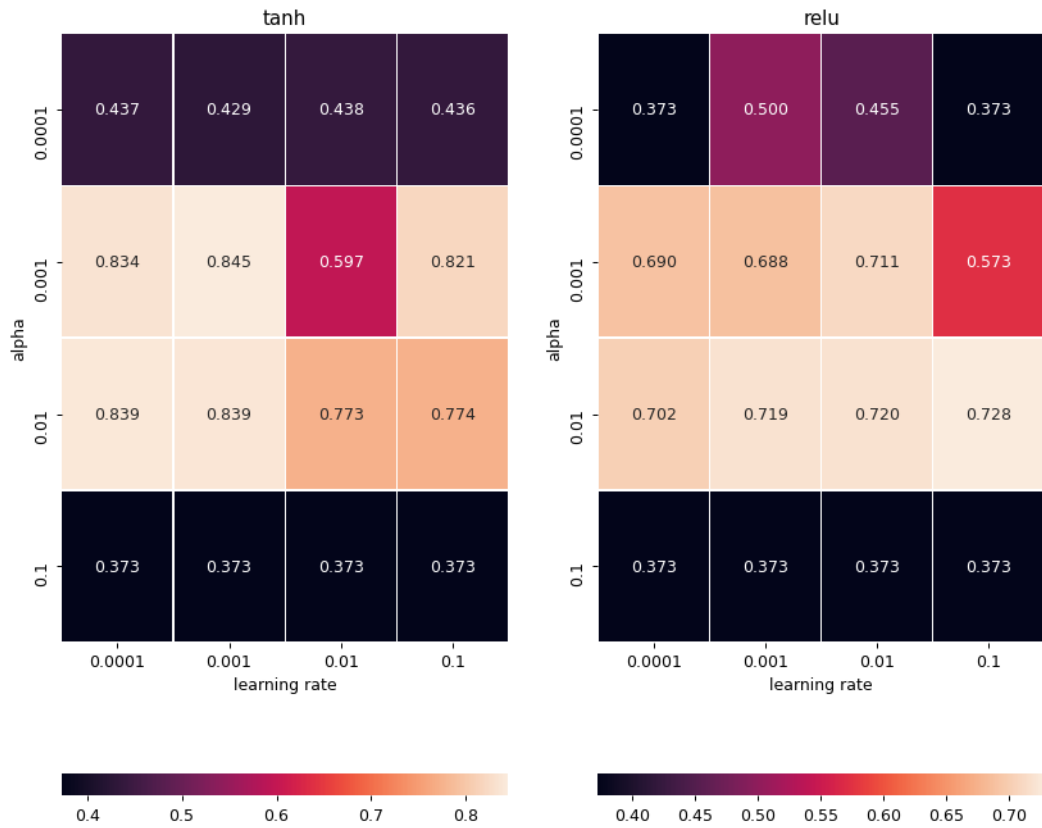


Figura 4.78: Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.

Accuracies obtenidas de la clasificación: Estabilidad del sistema, usando 7 dato(s) por ciclo y Mean

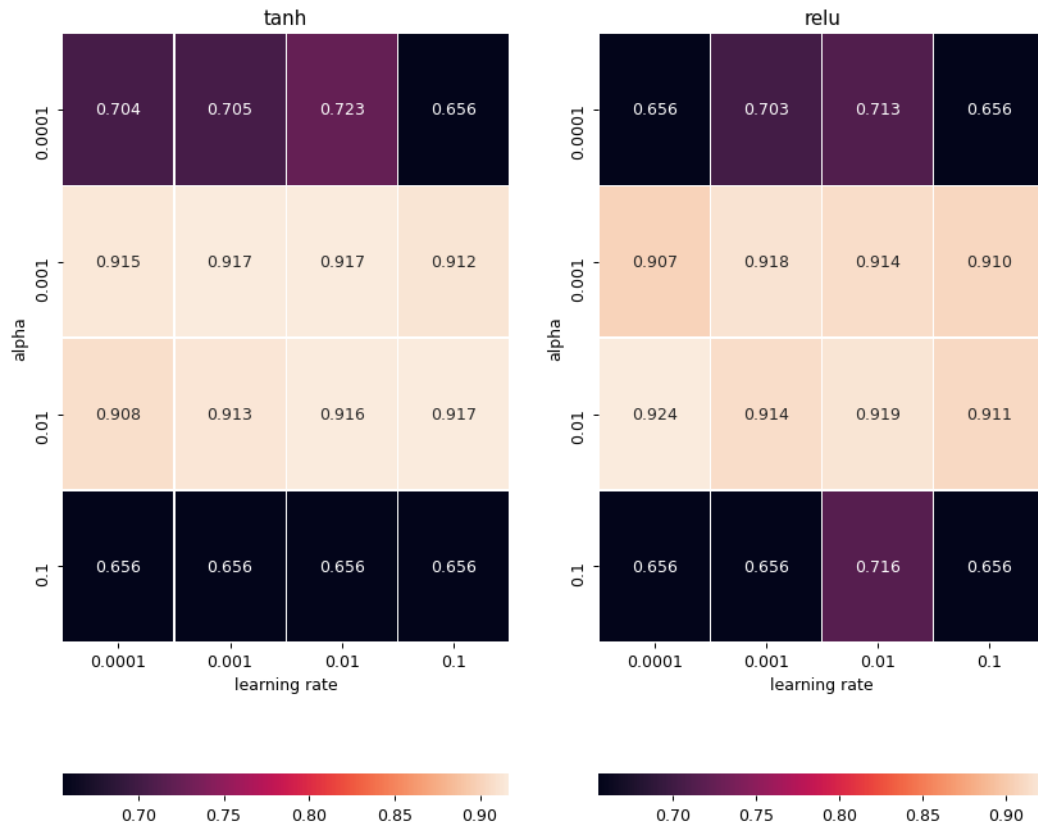


Figura 4.79: Heatmap con la variación del accuracy en función de los parámetros learning rate y alpha.

Capítulo 5

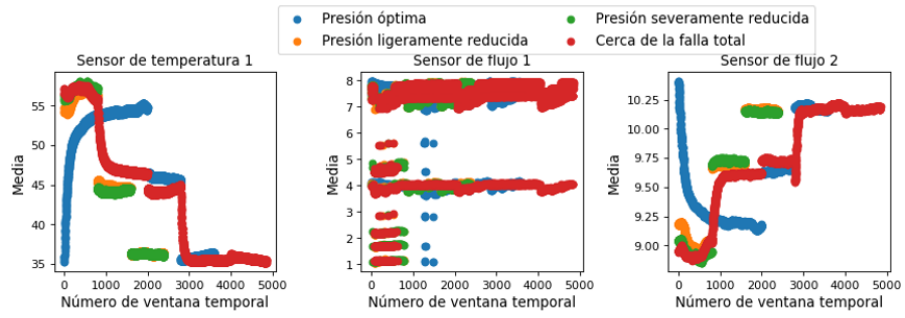
Discusión

En este capítulo se profundizará más en los resultados obtenidos, su sentido con respecto a la física del problema y a los aspectos más matemáticos que se encuentran detrás de los modelos de ML.

5.1. Efecto del largo de las ventanas de tiempo

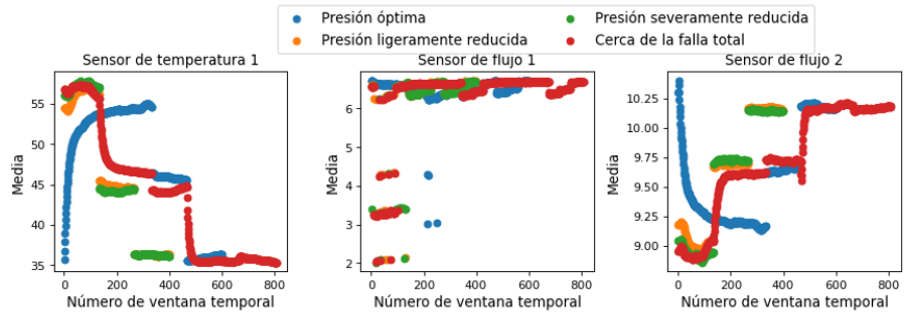
Como pudimos ver a lo largo del capítulo de resultados, el tamaño de las ventanas de tiempo sobre las cuales se aplicaron los parámetros de tiempo o más simplemente los datos extraídos por ciclo, fueron uno de los factores que tuvieron mas incidencia en el desempeño de los modelos. Como se explicó en la metodología, el aplicar parámetros de tiempo sobre los datos es algo usual al trabajar con series temporales, y era aún más importante en nuestro caso porque nos ayudo a sobrellevar el hecho de que los sensores entregaran diferentes volúmenes de datos. En un inicio se pensó que el reducir las ventanas y el agregar overlap a las ventanas ayudaría al desempeño del modelo puesto que siempre se busca trabajar con más datos, sin embargo al ir revisando los gráficos de los parámetros de tiempo usando diferentes combinaciones de largos de ventana y overlap ya se pudo notar una aparente baja en la calidad de los datos.

Media para la clasificación: Estado del acumulador, usando 6 dato(s) por ciclo



(a) 6 datos por ciclo

Media para la clasificación: Estado del acumulador, usando 1 dato(s) por ciclo



(b) 1 dato por ciclo

Figura 5.1: Comparación entre gráficos con igual parámetro de tiempo y clasificación, pero diferentes ventanas de tiempo.

En la figura 5.1 se puede notar cómo sobre todo en el sensor de flujo 1, al aumentar la cantidad de de datos que se extraen por ciclo, las clases comienzan a perder distinción entre ellas afectando el desempeño del modelo.

Accuracias a partir del modelo KNN, clasificación: Estado de la válvula

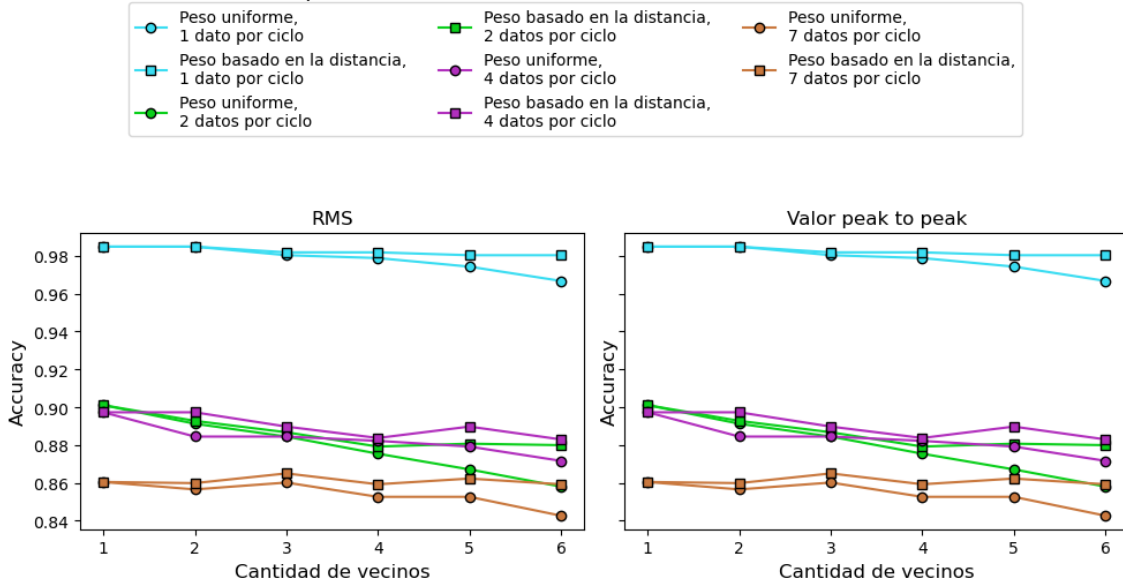


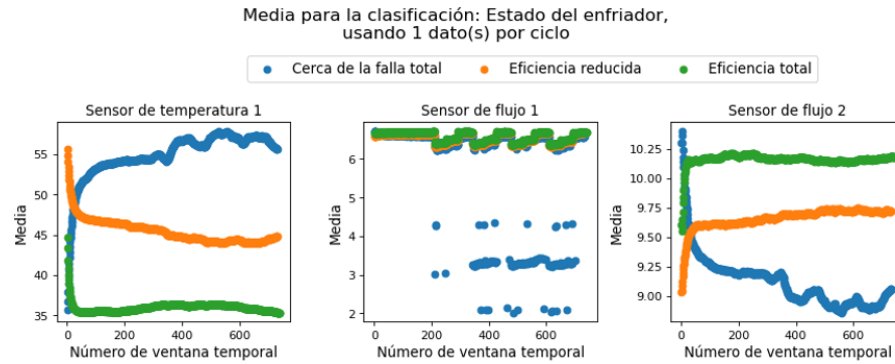
Figura 5.2: Accuracy de modelos KNN obtenido según las clases del estado de la válvula, usando RMS.

Revisando los desempeños de los modelos (ver figura 5.2) las sospechas se hicieron realidad, y efectivamente, contrario a lo que se pensaba al comenzar a trabajar con el dataset se pudo notar que en prácticamente en todos los modelos abordados el valor del accuracy tiende a bajar al aumentar la cantidad de datos extraídos por ciclo o al menos, los mejores resultados se alcanzan con una cantidad baja de datos por ciclo. Lo anterior se puede explicar por el hecho de que al tomar ventanas de tiempo menores a 60 segundos, el respectivo parámetro de tiempo no logra capturar toda la información de un ciclo para un determinado sensor y luego le entrega información “incompleta” a los modelos.

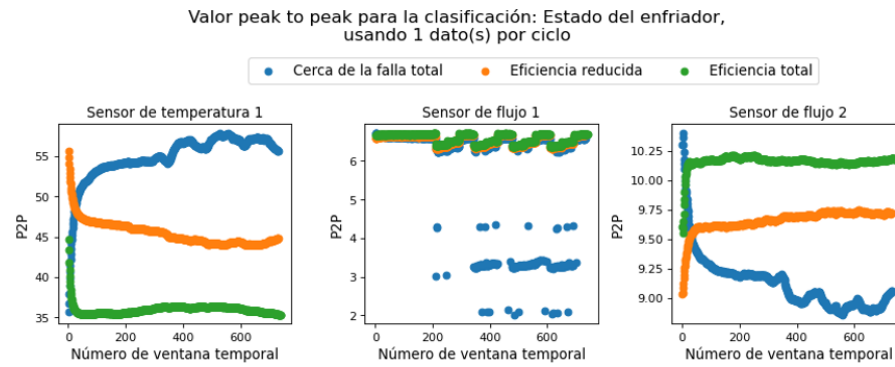
Si bien el aumentar los datos que se entregan al modelo usando overlap o reduciendo la ventana de tiempo que se toma suele mejorar el desempeño, esto es más común en casos como el del conocido *Bearing Fault Dataset*[27], donde se entregan arreglos de datos para diferentes estados de salud que contienen varios ciclos continuos, pero todos pertenecientes a una misma clase. El dataset usado en este trabajo de título en cambio, contiene datos donde el ciclo puede tomar diferentes clases a lo largo de las pruebas.

5.2. Diferencias entre parámetros de tiempo

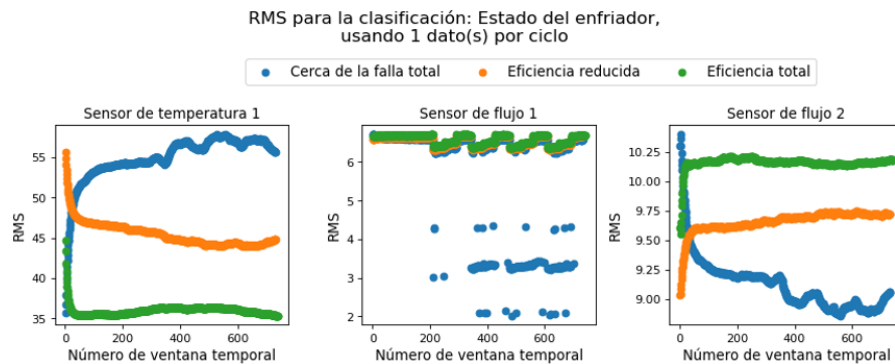
Desde el subcapítulo 4.2 y como se puede ver en la figura 5.3, se notó que los gráficos entregados por los parámetros de tiempo RMS, valor peak to peak y media presentaron resultados bastante similares, con lo cual, a priori se esperaba que obtuvieran también resultados bastante similares. Por otra parte como se ver en la figura 5.4, la varianza fue el único parámetro de tiempo que presentó un comportamiento distinto.



(a) Media



(b) RMS



(c) Varianza

Figura 5.3: Comparación entre los parámetros: Media, RMS y valor peak to peak, usando el mismo largo de ventana de tiempo y los mismos 3 sensores.

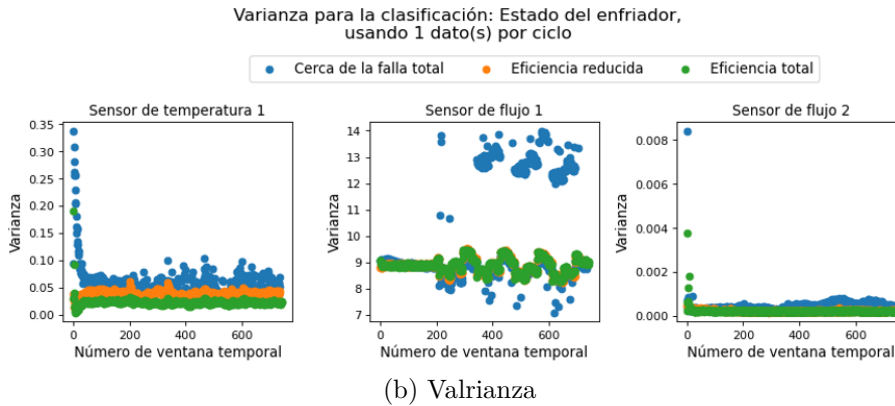
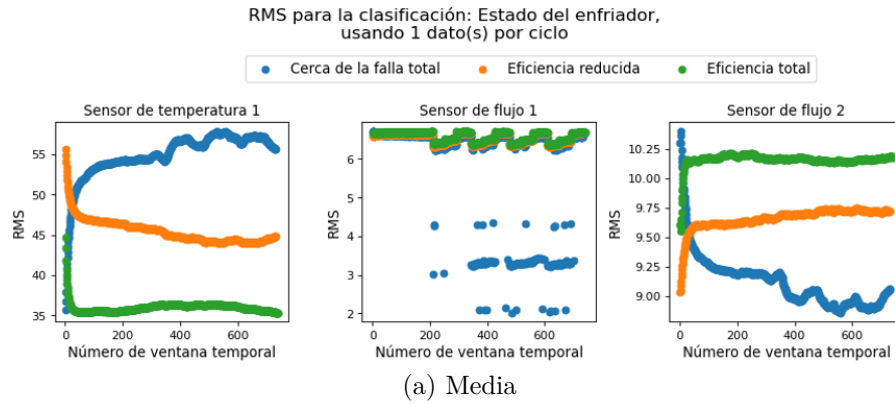


Figura 5.4: Comparación entre gráficos con igual parámetro de tiempo y clasificación, pero diferentes ventanas de tiempo.

El efecto de la similitud entre los parámetros de tiempo RMS, valor peak to peak y media, se notó bastante a la hora de obtener resultados, ya que prácticamente no hubieron diferencias en los desempeños de los modelos al clasificar los estados de salud de los diferentes activos físicos, usando diferentes ventanas de tiempo o al usar modelos distintos. Este efecto se puede explicar por el hecho de que el RMS, la media y el valor peak to peak son métricas que describen directamente la forma de la señal recibida por los sensores, mientras que la varianza es una métrica que describe la distribución de los datos.

También se notó que en la mayoría de los resultados de los diferentes modelos, la varianza presenta un desempeño más bajo que los demás parámetros de tiempo. Lo que se puede explicar gráficamente al mirar figura 5.4, donde se ve claramente como la varianza tiende a solapar mas los datos extraídos. Desde un punto de vista más estadístico, se puede deber a que las fallas inducidas en el sistema hidráulico generan un efecto más fuerte en los valores que toman las distintas variables de operación que en el cambio de las dispersión de los datos.

5.3. Efecto del cambio de parámetros de los modelos

5.3.1. Random Forest

Uno de los aspectos que mas llama la atención , es el hecho de que la cantidad de árboles no incide demasiado en el accuracy obtenido. Esto sucede posiblemente porque los datos fueron preparados muy cuidadosamente a partir del montaje experimental, y también porque los diferentes estados de salud se encuentran distribuidos de manera uniforme en el dataset, lo que permite que al dividir los datos para los diferentes arboles de decisión, incluso si se tienen pocos, estos lograran generalizar el comportamiento del set de entrenamiento. También se ve puede ver que el criterio de gini presenta resultados un poco mejores al criterio de entropía, lo cual es bastante común y se debe principalmente a la diferencia entre ambos algoritmos[28].

5.3.2. K Nearest Neighbors

En el caso del KNN, el hecho de que la cantidad de vecinos no tenga un efecto significativo en los valores de accuracy se pude explicar por el proceso previo de clusterización. Como se explicó en el capítulo de antecedentes un modelo KNN divide en clusters los datos antes de entrenar, por lo cual seguramente este dataset fue muy bien dividido por cada modelo de KNN, permitiendo así que al predecir la clase de un dato la mayoría de sus vecinos fueran de su misma clase. Con respecto al peso de lo vecinos, es bastante intuitivo pensar que los vecinos más cercanos serán los que seguramente pertenezcan a la misma clase,lo que explica que el peso basado en la distancia presente mejor resultados que el peso uniforme.

5.3.3. Linear Discriminant Analysis

Para el LDA no se variaron hiperparámetros por lo que los resultados mostrados en función de los datos por ciclo y los parámetros de tiempo ya fueron explicados. Lo que si es destacable, es que para algunos casos este tipo de modelos entregó resultados muy buenos (cercaos o iguales a 100 %) mientras que en otros el accuracy fue menor a 50 %. Esto puede deberse a que como se explicó en la metodología, las diferentes clasificaciones usaron diferentes conjuntos de sensores.

5.3.4. Support Vector Machines

En cuanto a los kernels de SVM, podemos decir que se dieron resultados de acuerdo a lo esperado. Si vemos la figura 2.14 notamos que el desempeño mostrado por los kernels sigue el mismo orden que los resultados obtenidos (rbf > lineal > sigmoide). Lo anterior se puede ver claramente en la figura ya mencionada, donde la trasformación que aplica rbf es mucho mas robusta que las otras, y también se puede ver como la forma en que separa los datos la transformación sigmoide puede ser peor que la lineal. Con respecto al parámetro C, en los heatmaps se pudo ver que este no incide mucho en los resultados, lo cual se condice con los resultados de los otros modelos, ya que como hemos visto, lo resultados apuntan a que el dataset es “fácil” de dividir en sus diferentes clases, y este parámetro indica una margen de

tolerancia para datos queden fuera del espacio de su clase.

5.3.5. Multi layer perceptron (MLP)

Como se habló en el capítulo de resultados, la función de activación sigmoide (logistic) fue la que tuvo el peor desempeño de las 3, seguida por la ReLU y finalmente la TanH. Por lo general el desempeño de la función de activación está relacionado al rango en el que se mueven los datos, en nuestro, es entre 0 y 1 y si consideramos este como el recorrido de las funciones tenemos que el dominio de la función ReLU y de la función TanH. también es (0,1), mientras que para la sigmoide, su dominio sería (0.5,1). El problema anterior probablemente este generando una pérdida de información y explica el por qué del desempeño más bajo por parte de la función sigmoide. Para el learning rate, vimos que en términos generales no afecta mucho el accuracy que se obtiene, lo que nos indica que el modelo no tiene problemas para converger. Para el parámetro alpha podemos decir que influye bastante en el desempeño y que además, los mejores valores se encuentran en rangos bastante acotados.

5.4. Diferencias entre sensores

Como se abordó en detalle en el subcapítulo 4.1, el grado de separación de las diferentes clases correspondientes a los diferentes componentes variaba mucho en función de los sensores. Para el caso de los sensores de flujo por ejemplo, se vieron buenos resultados en general, sobre todo con el sensor FS1. Esto tiene sentido si pensamos que aunque, los componentes se vayan degradando con el tiempo, el flujo dentro del sistema no debiese cambiar.

En cuanto a los sensores de temperatura, se pudo ver que en general es muy fácil distinguir entre las diferentes clases, lo que nos dice que los valores que toma la temperatura están altamente coleccionados con los estados de salud de los diferentes componentes del sistema.

Para los gráficos de presión, se tiene resultados mas variados respecto a su capacidad para mostrar clases separadas. Si bien se esperaba que la presión al igual que la temperatura fuera una variable muy correlacionada con los estados de salud, vemos que no es así en todos los casos, lo que podría deberse a las posiciones en las que se encuentran los sensores respecto a otros componentes.

Con respecto a las demás variables, solo los sensores de la potencia del enfriador y de la eficiencia del enfriador muestran curvas distinguibles, esto nos lleva a pensar que particularmente el estado de salud del cooler domina sobre el comportamiento de otros componentes.

5.5. Resultados Generales

En la siguiente tabla se pueden apreciar lo mejores resultados (en porcentaje de accuracy) logrados para cada clasificación usando diferentes modelos. Estos resultados son el mejor entre todas las combinaciones de datos por ciclo y parámetros de tiempo.

Tabla 5.1: Resultados globales

		Enfriador	Válvula	Bomba	Acumulador	Sistema
RF	TT	100 %	98.6 %	99.8 %	97.6 %	98.8 %
KNN	TT	100 %	98.5 %	99.7 %	97.9 %	99.1 %
SVM rbf	TT	100 %	98.9 %	98.8 %	88.4 %	95.8 %
	EP	100 %	95.7 %	64.2 %	65.7 %	-
SVM lineal	TT	100 %	67.4 %	83.1 %	51.2 %	92.6 %
	EP	100 %	100 %	72.4 %	51.6 %	-
LDA	TT	100 %	62.6 %	99.8 %	73.1 %	91.8 %
	EP	100 %	100 %	72.4 %	51.6 %	-
MLP	TT	100 %	62.4 %	99.1 %	83.4 %	91.8 %
	EP	100 %	100 %	80 %	50.4 %	-

- **TT:** Resultado obtenido en el presente trabajo de título.
- **EP:** Resultado obtenido en el estudio previo.

De forma global vemos que se obtuvieron muy buenos resultados. Claramente los mejores puntajes los consigue el enfriador, luego, en segundo lugar podríamos tener a la estabilidad del sistema o el estado de la bomba, seguidos por el estado del acumulador y finalmente el estado de la válvula. En cuanto a modelos, todos consiguen buenos puntajes, sin embargo KNN y RF demuestran ser versátiles y consiguen valores de accuracy muy altos para todas las clasificaciones. Finalmente, con respecto a las diferencias con el estudio previo, notamos que se consigue superar o igualar los puntajes en todas las clasificaciones menos en el estado de la válvula.

Capítulo 6

Conclusiones

Se implementaron varios modelos de RF, KNN, SVM, LDA y MLP, variando además sus parámetros y los datos que se le entregaron para procesar. A raíz de lo anterior se pudo lograr un gran entendimiento del problema abordado, así como también de los modelos que se utilizaron. Se destaca la robustez de los modelos Random Forest y K Nearest Neighbors, ya que incluso siendo modelos que se usan poco actualmente al no estar en el estado del arte, generaron resultados muy buenos.

Uno de los aspectos que más se destacan del análisis de los resultados, fue el desempeño inusual que obtuvo el modelo al variar las ventanas de tiempo, y al no tomar todos los datos de un ciclo de operación, lo que pareció muy raro en un principio, pero finalmente se comprendió que la forma en la cual se generaron los datos fue distinta a los datasets usuales que se usan para estudiar el monitoreo de condición o la predicción de fallas.

Se logró con muy buenos resultados la predicción de los diferentes estados de salud del sistema y sus componentes, superando la mayoría de los benchmarks obtenidos por el estudio previo.

Finalmente se destaca la importancia del desarrollo de técnicas de monitoreo de condición como una gran herramienta que permitirá la reducción de los impactos que pueden causar las fallas en activos físicos como costos monetarios, tiempos de paradas o incluso fatalidades.

Bibliografía

- [1] Helwig, N., Pignanelli, E., & Schütze, A. (2015). Condition monitoring of a complex hydraulic system using multivariate statistics. 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings. <<https://ieeexplore.ieee.org/abstract/document/7151267>>
- [2] Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F., & Cambria, E. (2019). Supervised Learning for Fake News Detection. IEEE Intelligent Systems, 34(2), 76–81. <<https://doi.org/10.1109/mis.2019.2899143>>
- [3] Buchan, Susannah J.; Mahú, Rodrigo; Wuth, Jorge; Balcazar-Cabrera, Naysa; Gutierrez, Laura; Neira, Sergio; Yoma, Néstor Becerra; (2019). An unsupervised Hidden Markov Model-based system for the detection and classification of blue whale vocalizations off Chile. <<http://repositorio.uchile.cl/handle/2250/171568>>
- [4] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.
- [5] train_test_split function documentation. SCKITIT LEARN DOCUMENTATION. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- [6] MinMaxScaler function documentation. SCKITIT LEARN DOCUMENTATION. <<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>>
- [7] Random Forest Classifier model documentation. SCKITIT LEARN DOCUMENTATION <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [8] KNeighbors Classifier model documentation. SCKITIT LEARN DOCUMENTATION <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>
- [9] Support Vector Classifier model documentation. SCKITIT LEARN DOCUMENTATION <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- [10] Linear discriminant analysis model documentation. SCKITIT LEARN DOCUMENTATION https://scikit-learn.org/stable/modules/lda_qda.html
- [11] Multi Layer Perceptron model documentation. SCKITIT LEARN DOCUMENTATION https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
- [12] Escalar Frente A Normalizar datos. ICHI.PRO. (2021, January 11). <https://ichi.pro/es/escalar-frente-a-normalizar-datos-101383047510660>.

- [13] Linear Discriminant Analysis. (2014, August 3). Dr. Sebastian Raschka. https://sebastianraschka.com/Articles/2014_python_lda.html
- [14] S. (2019, June 4). Forward Selection: Definition. Statistics How To. <https://www.statisticshowto.com/forward-selection/>
- [15] Heras, J. M. (2020, September 18). Random Forest (Bosque Aleatorio): combinando árboles. IArtificial.net. <https://www.iartificial.net/random-forest-bosque-aleatorio/>
- [16] Johanna Orellana Alvear - johanna.orellana@ucuenca.edu.ec. Arboles de decision y Random Forest. <https://bookdown.org/content/2031/arboles-de-decision-parte-i.html>
- [17] K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint. Www.Javatpoint.Com. <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
- [18] neighbors module documentation. SCKITIT LEARN DOCUMENTATION <https://scikit-learn.org/stable/modules/neighbors.html>
- [19] Gandhi, R. (2018, July 5). Support Vector Machine — Introduction to Machine Learning Algorithms. Medium. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- [20] Rodrigo, J. A. (2017). Máquinas de Vector Soporte (Support Vector Machines, SVMs). CIENCIA DE LOS DATOS. https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines
- [21] SVM module documentation. SCKITIT LEARN DOCUMENTATION <https://scikit-learn.org/stable/modules/svm.html>
- [22] The difference of kernels in SVM? (2014, March 20). Cross Validated. <https://stats.stackexchange.com/questions/90736/the-difference-of-kernels-in-svm>
- [23] MLP Classifier model documentation. SCKITIT LEARN DOCUMENTATION https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier
- [24] Neuronal networks module documentation. SCKITIT LEARN DOCUMENTATION https://scikit-learn.org/stable/modules/neural_networks_supervised.html
- [25] ¿Qué es una Red Neuronal? (2018, September 3). Codificando Bits. <https://www.codificandobits.com/blog/que-es-una-red-neuronal/>
- [26] Sharma, S. (2021, July 4). Activation Functions in Neural Networks - Towards Data Science. Medium. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- [27] Fault Data Sets. (2020, February 27). Society For Machinery Failure Prevention Technology. <https://www.mfpt.org/fault-data-sets/>
- [28] GeeksforGeeks. (2021, June 29). Gini Impurity and Entropy in Decision Tree - ML. <https://www.geeksforgeeks.org/gini-impurity-and-entropy-in-decision-tree-ml/>
- [29] Salcedo, L. (2020, December 22). Introducción al Machine Learning #3 - Tipos de algoritmos de aprendizaje. Mi Diario Python. <https://pythondiario.com/2017/12/introduccion-al-machine-learning-3.html>

- [30] N. (2021, March 22). Qué es overfitting y underfitting y cómo solucionarlo. Aprende Machine Learning. <https://www.aprendemachinlearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/>
- [31] Mohajon, J. (2021, July 24). Confusion Matrix for Your Multi-Class Machine Learning Model. Medium. <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>
- [32] Ved, M. (2018b, July 20). Feature Selection and Feature Extraction in Machine Learning: An Overview. Medium. <https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96>
<https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96>
- [33] Ved, M. (2018, July 20). Feature Selection and Feature Extraction in Machine Learning: An Overview. Medium. <https://saurabhmirgane007.medium.com/most-popular-decision-tree-algorithms-77bee49994dd>
- [34] What are the advantages of logistic regression over decision trees? Are there any cases where it's better to use logistic regression instead of decision trees? (2017). Quora. <https://www.quora.com/What-are-the-advantages-of-logistic-regression-over-decision-trees-Are-there-any-cases-where-its-better-to-use-logistic-regression-instead-of-decision-trees>
- [35] Rong, Y. (2018). Figure 4 from Post hoc support vector machine learning for impedimetric biosensors based on weak protein-ligand interactions. | Semantic Scholar. Semantic Scholar. <https://www.semanticscholar.org/paper/Post-hoc-support-vector-machine-learning-for-based-Rong-Padron/0981bdd2a18de7f4e13f6713c7f153f6c8c1cbb5/figure/3>
- [36] Kurt, H. I., & Oduncuoglu, M. (2015). Application of a Neural Network Model for Prediction of Wear Properties of Ultrahigh Molecular Weight Polyethylene Composites. International Journal of Polymer Science, 2015, 1–11. https://www.researchgate.net/figure/The-basic-structure-of-an-artificial-neuron-43_fig3_282849515
- [37] Sharma, S. (2019, December 17). Explained: Deep Learning in Tensorflow — Chapter 1 - Towards Data Science. Medium. <https://towardsdatascience.com/explained-deep-learning-in-tensorflow-chapter-1-9ab389fe90a1>. <https://towardsdatascience.com/explained-deep-learning-in-tensorflow-chapter-1-9ab389fe90a1>

Anexo A

Resultados ampliados

A.1. Gráficos de los datos sin procesar

A.1.1. Gráficos de temperatura

A.1.1.1. Gráficos del enfriador

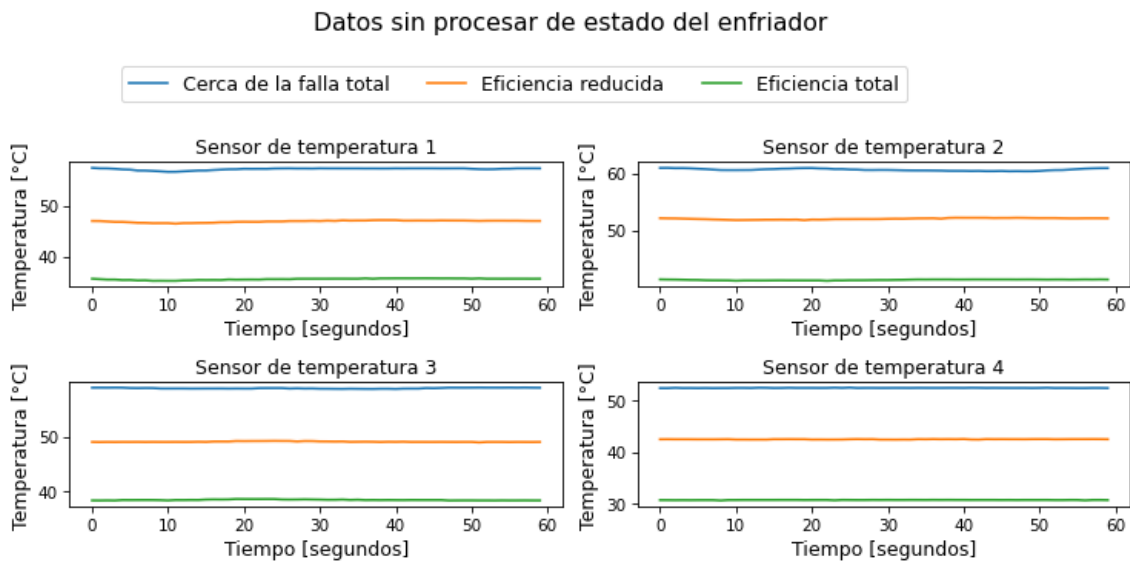


Figura A.1: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado enfriador, iteración 1.

Datos sin procesar de estado del enfriador

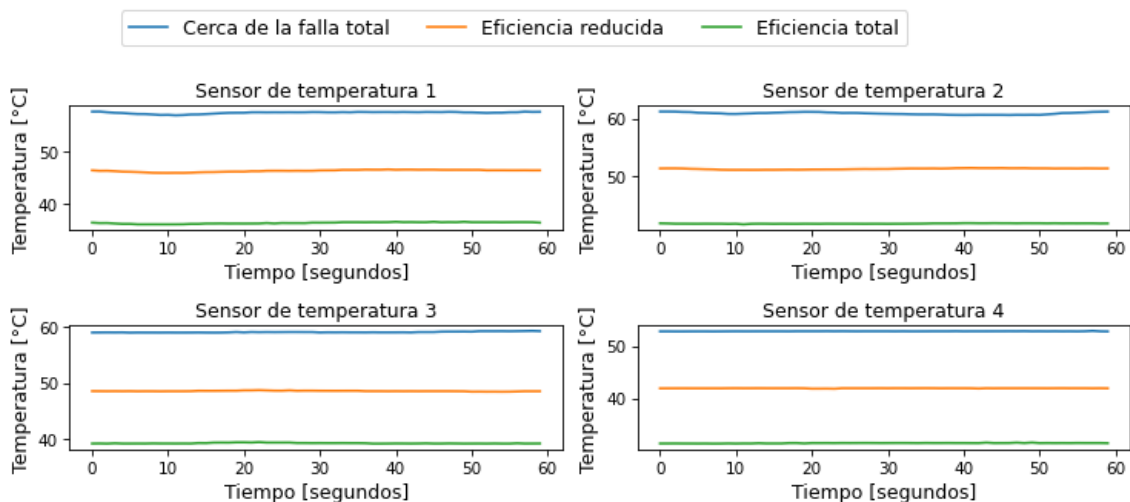


Figura A.2: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 2.

Datos sin procesar de estado del enfriador

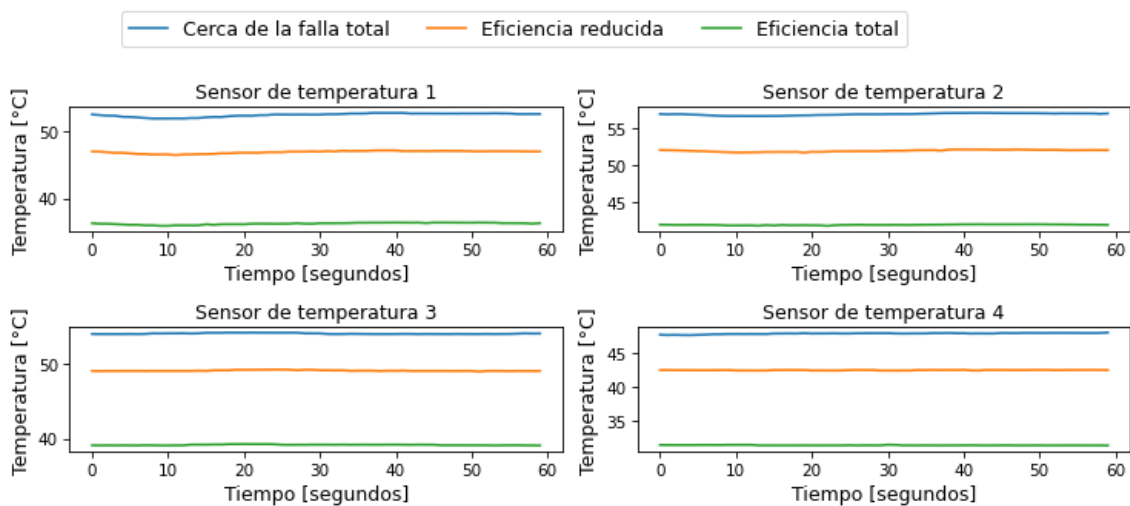


Figura A.3: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 3.

Datos sin procesar de estado del enfriador

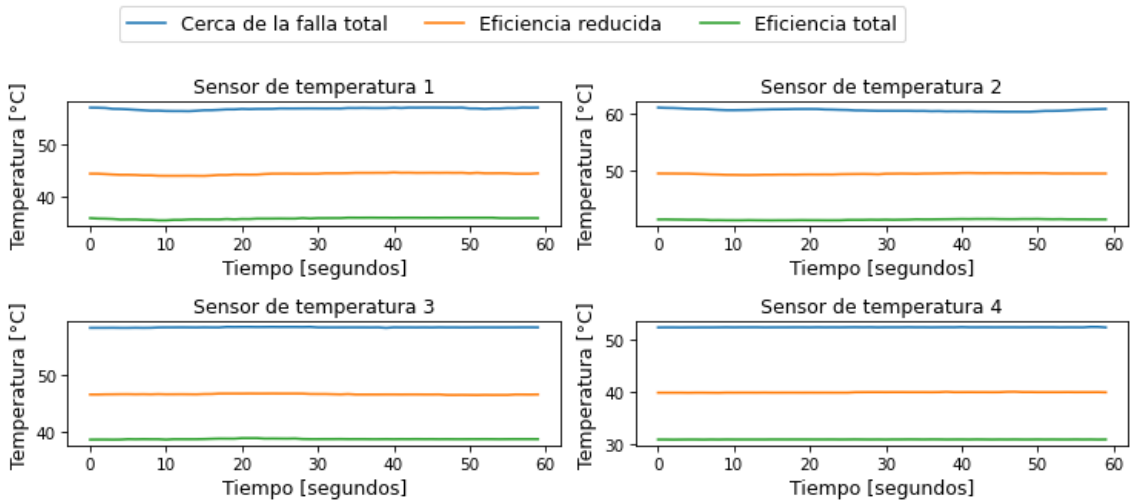


Figura A.4: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 4.

Datos sin procesar de estado del enfriador

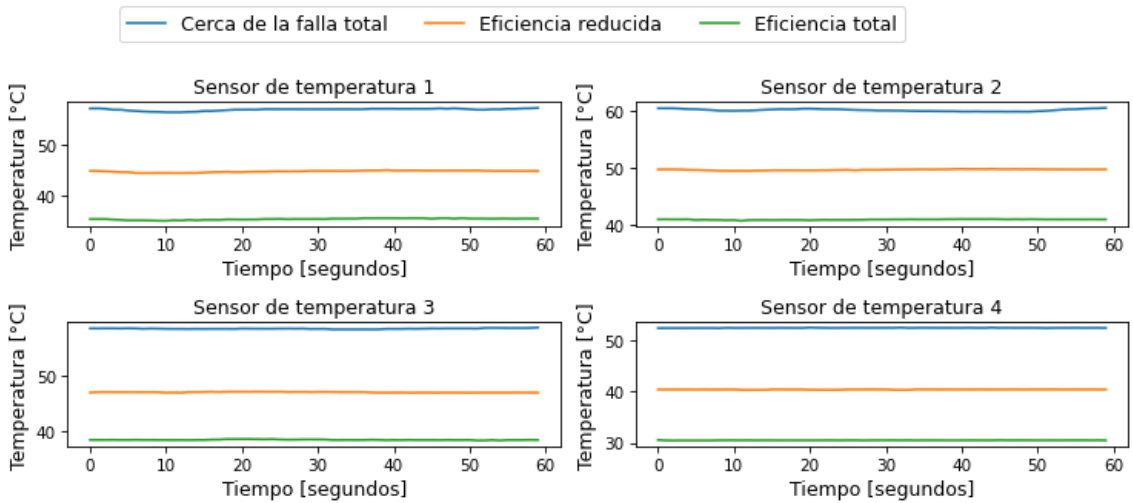


Figura A.5: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del enfriador, iteración 5.

A.1.1.2. Gráficos de la válvula

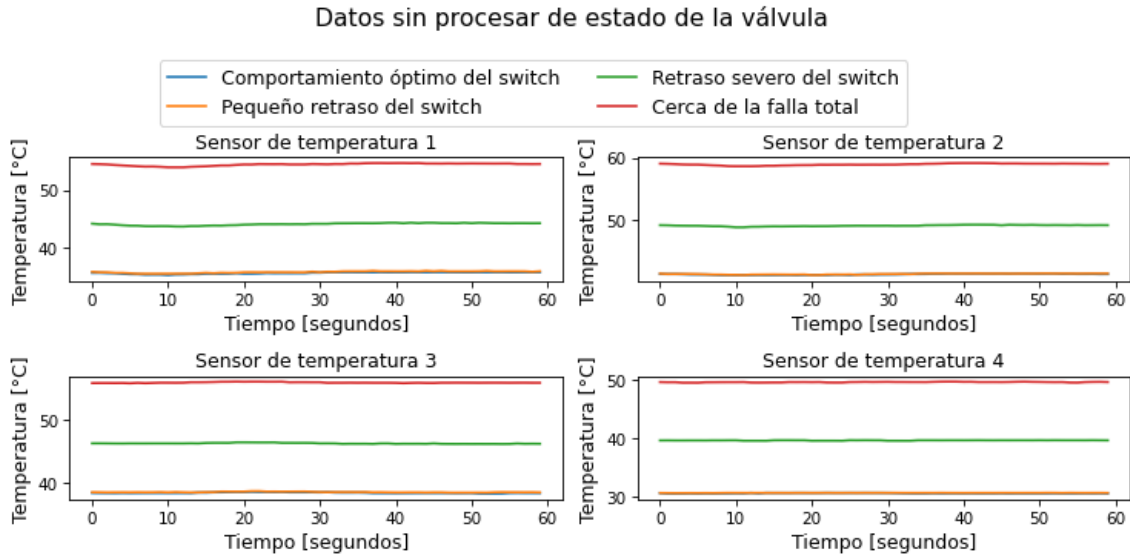


Figura A.6: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 1.

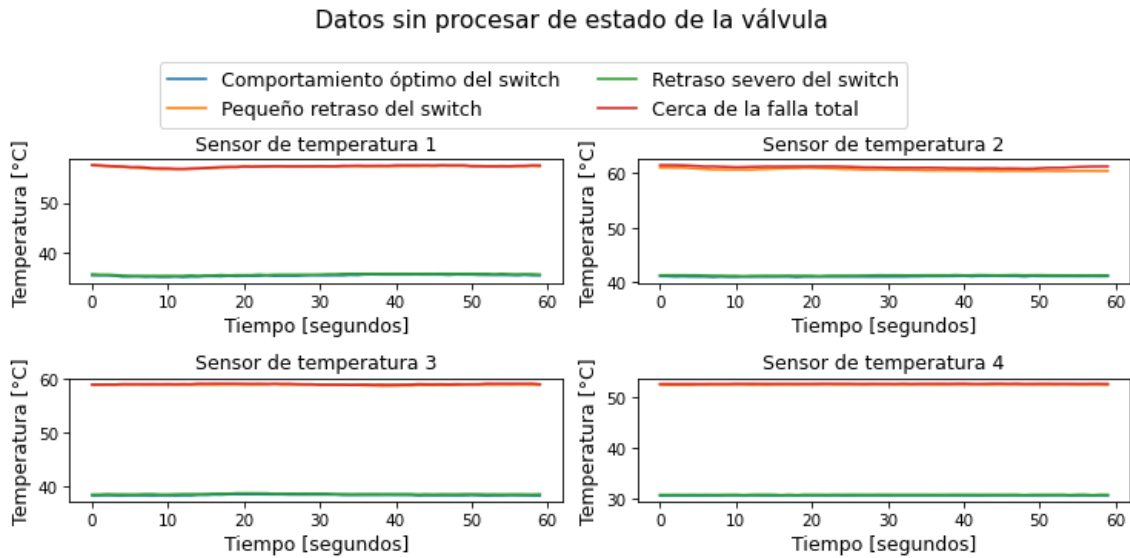


Figura A.7: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 2.

Datos sin procesar de estado de la válvula

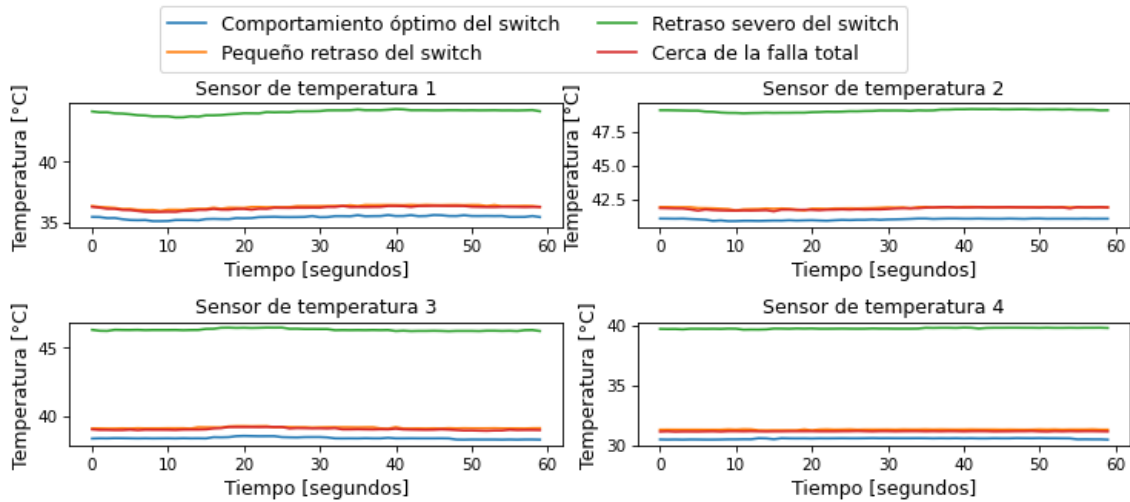


Figura A.8: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 3.

Datos sin procesar de estado de la válvula

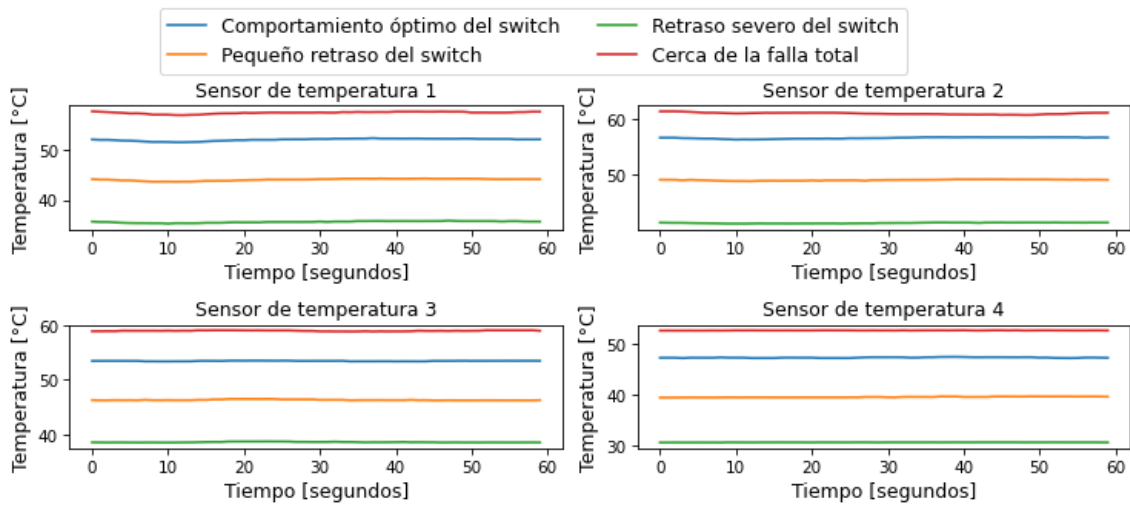


Figura A.9: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 4.

Datos sin procesar de estado de la válvula

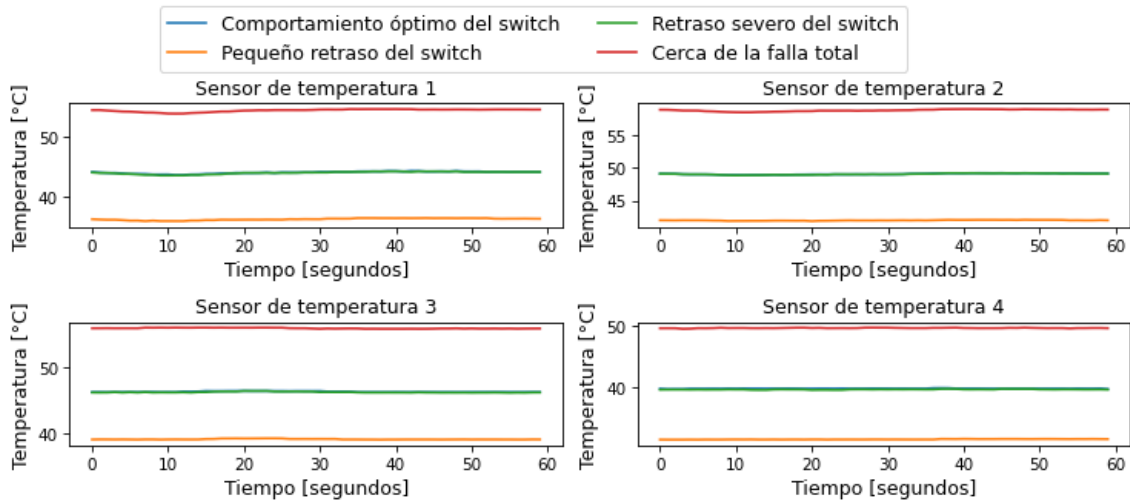


Figura A.10: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado de la válvula, iteración 5.

A.1.1.3. Gráficos de la bomba

Datos sin procesar de fuga en la bomba

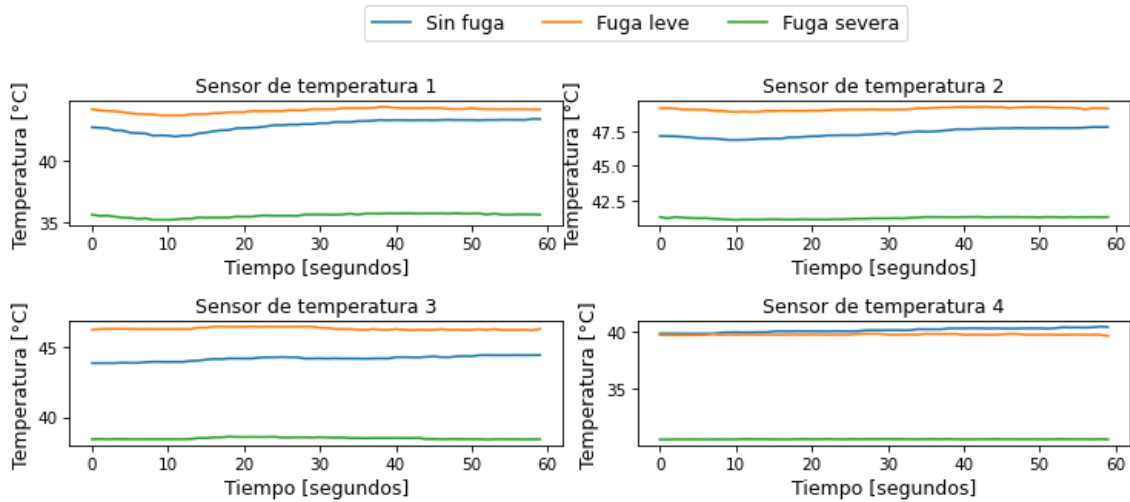


Figura A.11: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 1.

Datos sin procesar de fuga en la bomba

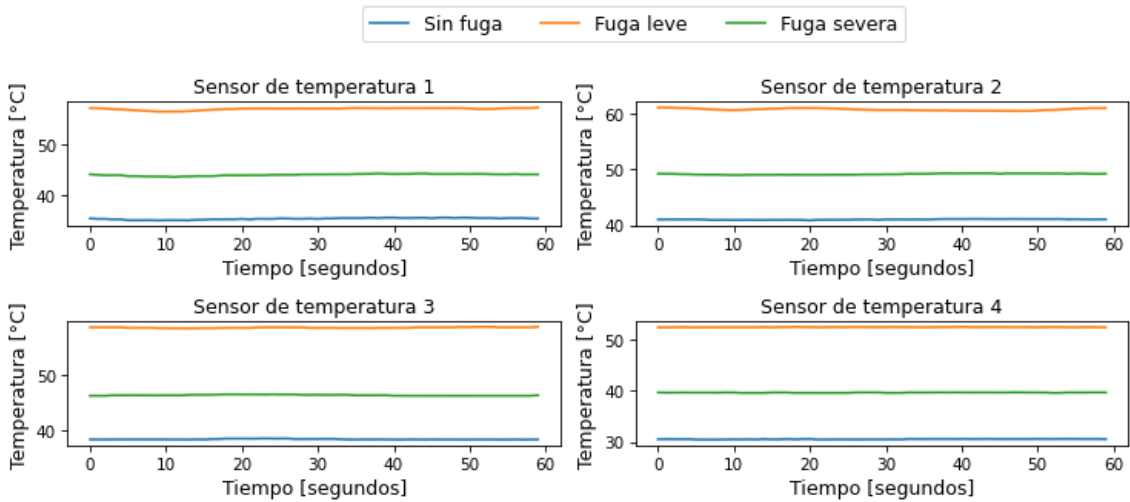


Figura A.12: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 2.

Datos sin procesar de fuga en la bomba

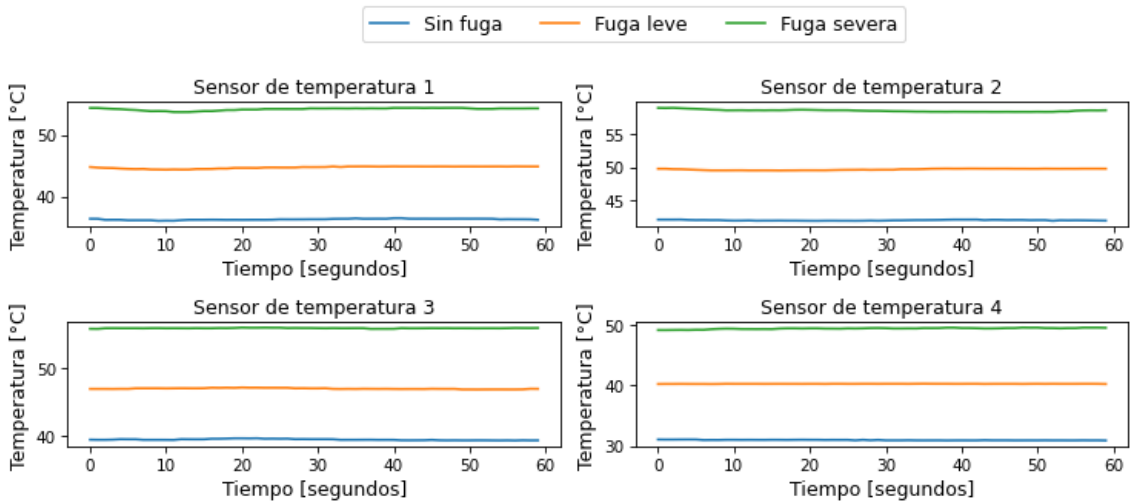


Figura A.13: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 3.

Datos sin procesar de fuga en la bomba

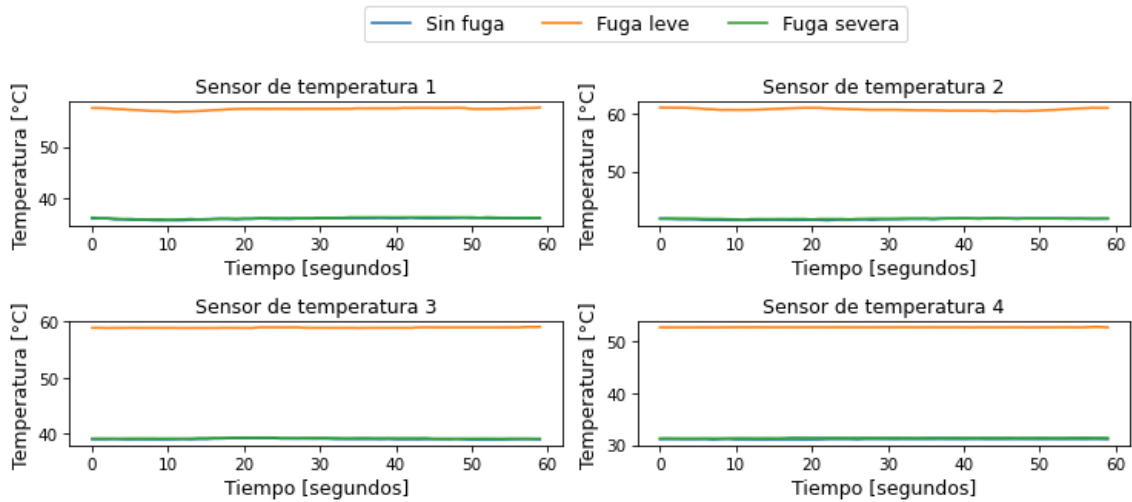


Figura A.14: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 4.

Datos sin procesar de fuga en la bomba

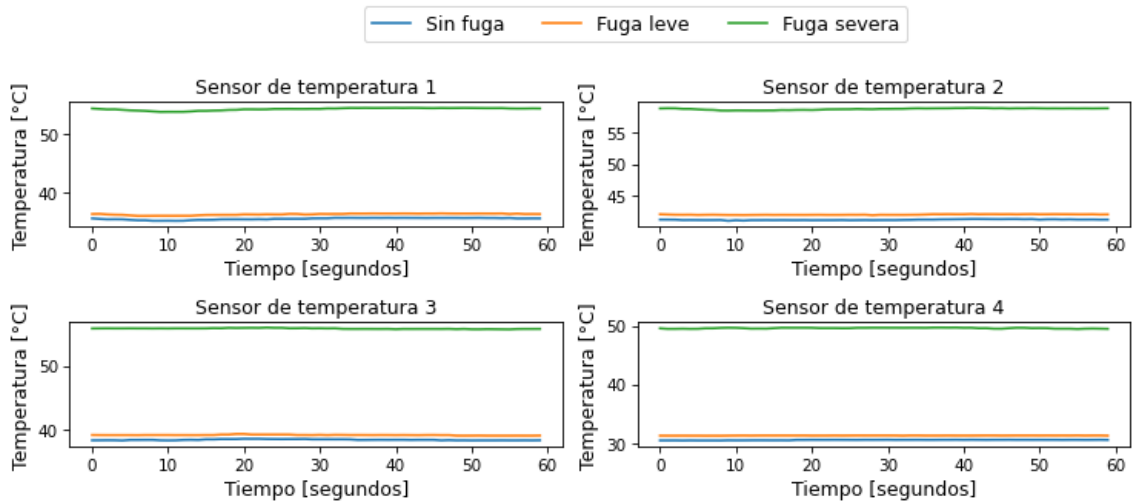


Figura A.15: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la fuga en la bomba, iteración 5.

A.1.1.4. Gráficos del acumulador

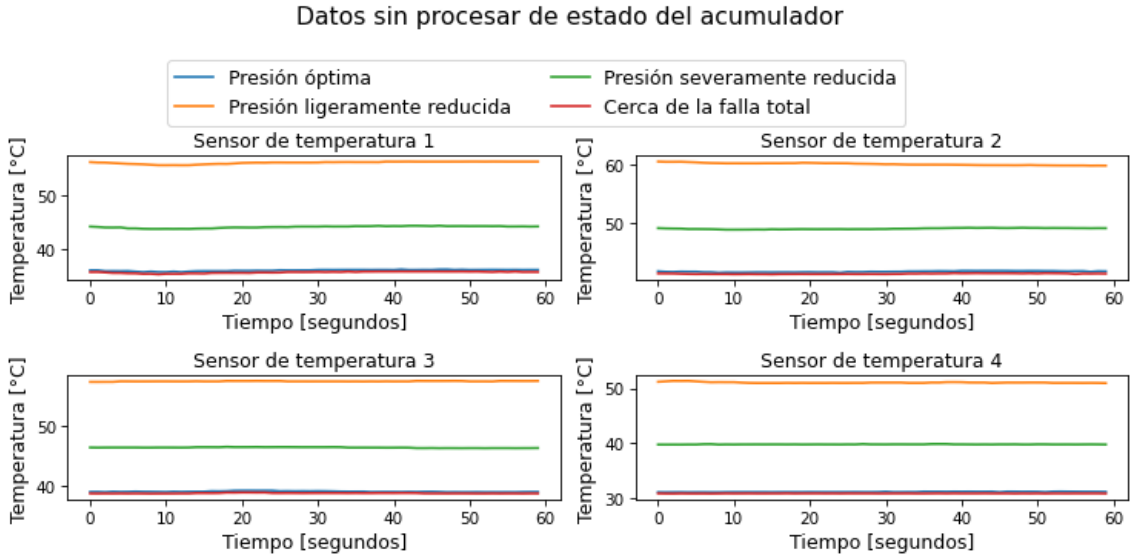


Figura A.16: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 1.

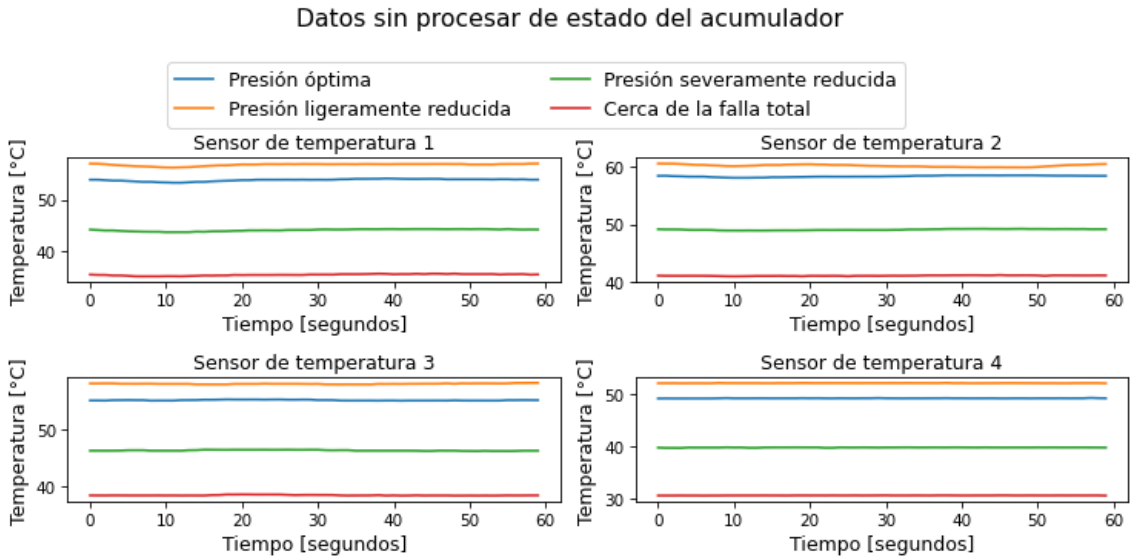


Figura A.17: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 2.

Datos sin procesar de estado del acumulador

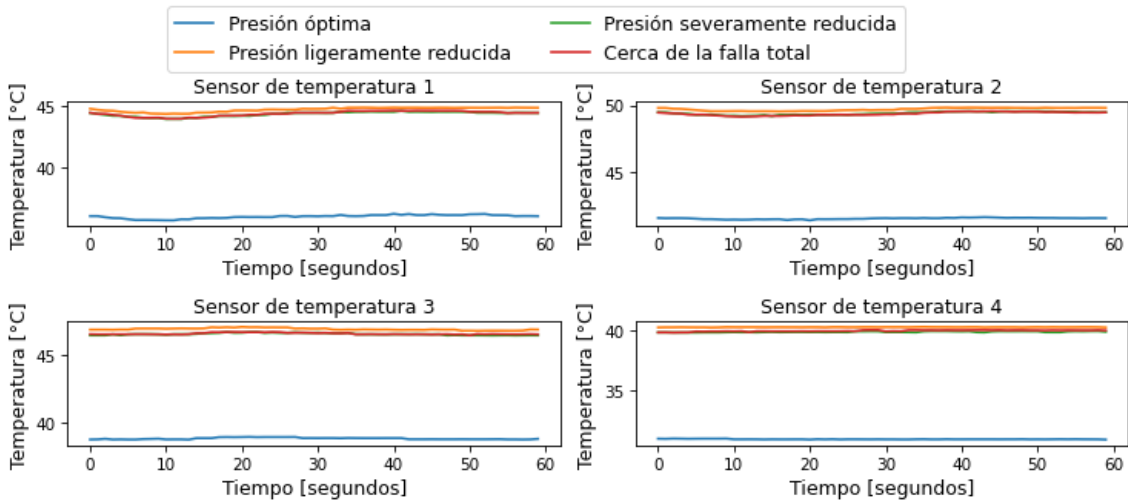


Figura A.18: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 3.

Datos sin procesar de estado del acumulador

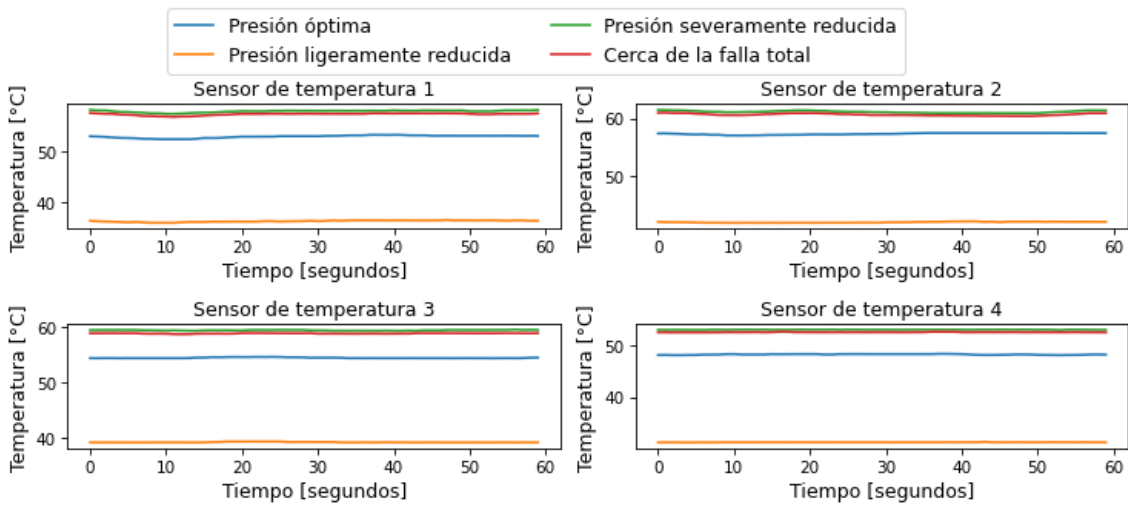


Figura A.19: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 4.

Datos sin procesar de estado del acumulador

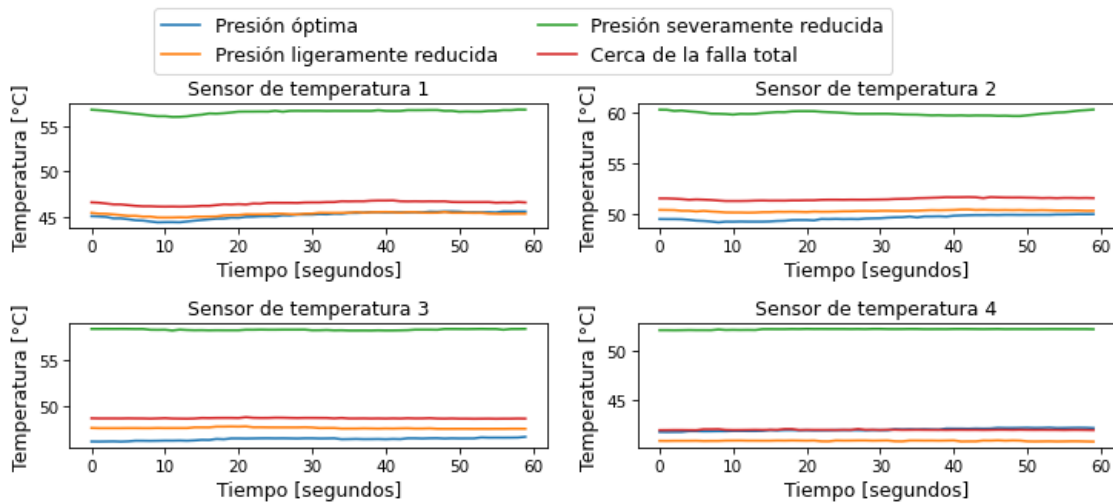


Figura A.20: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases del estado del acumulador, iteración 5.

A.1.1.5. Gráficos de la estabilidad del sistema

Datos sin procesar de estabilidad del sistema

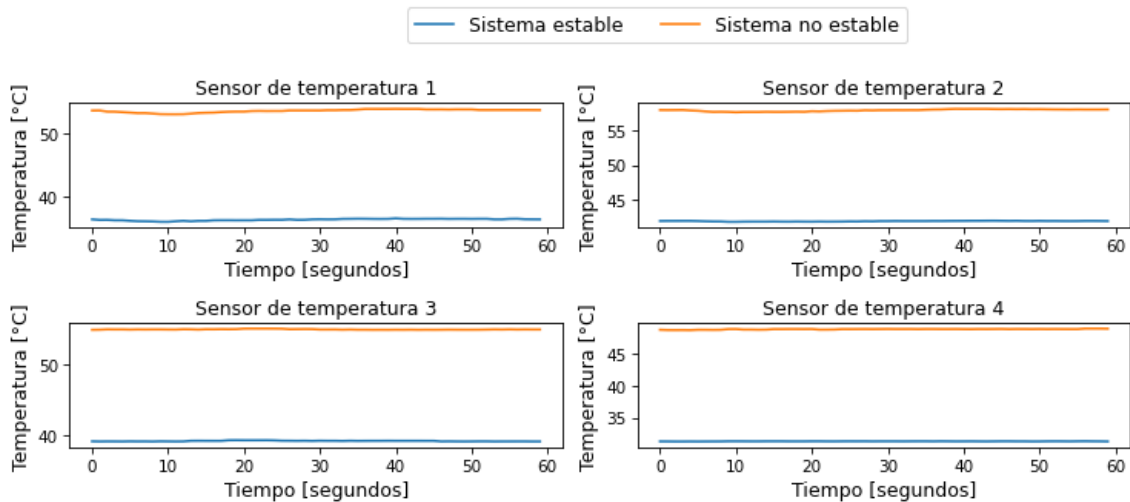


Figura A.21: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 1.

Datos sin procesar de estabilidad del sistema

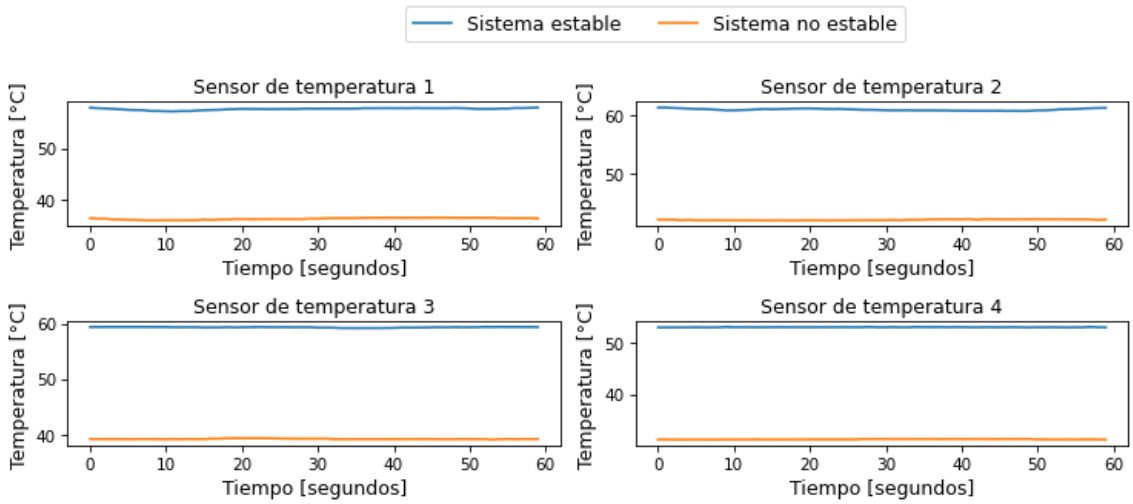


Figura A.22: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 2.

Datos sin procesar de estabilidad del sistema

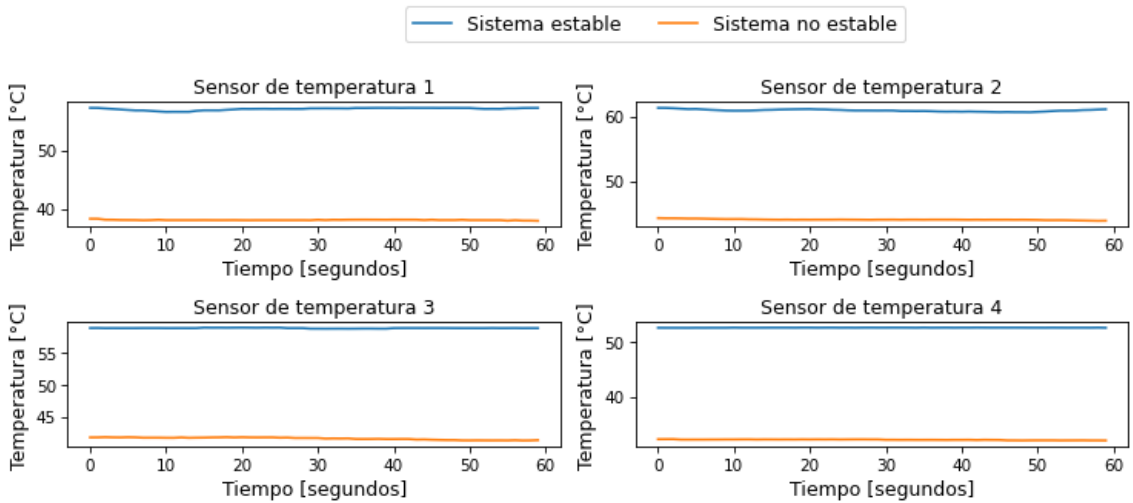


Figura A.23: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 3.

Datos sin procesar de estabilidad del sistema

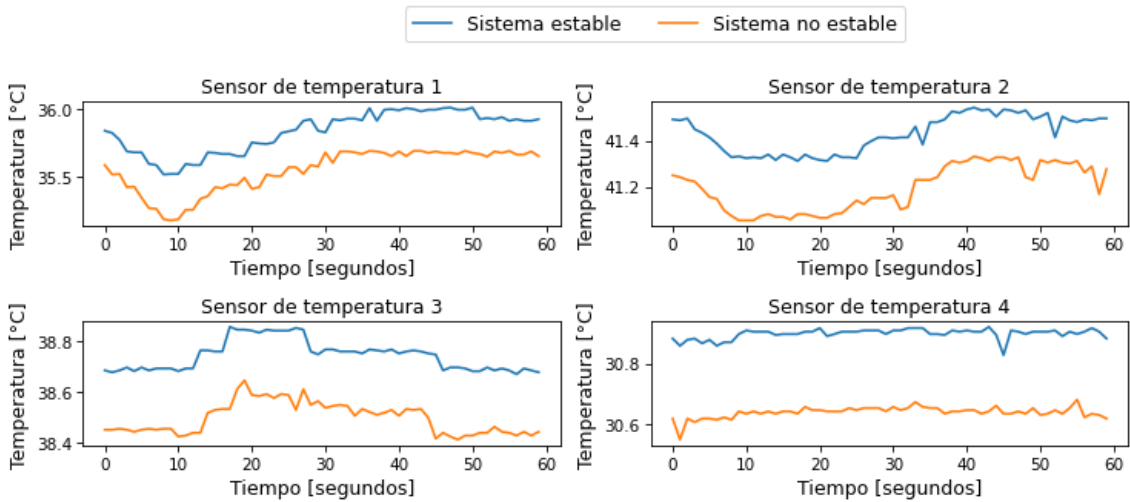


Figura A.24: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 4.

Datos sin procesar de estabilidad del sistema

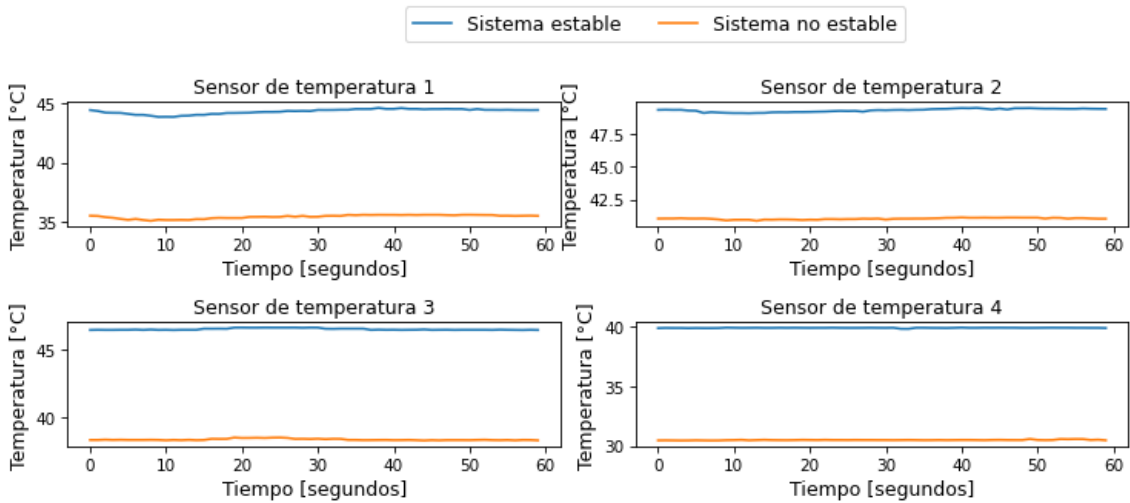


Figura A.25: Gráficos de los datos sin procesar obtenidos por los sensores de temperatura, separando según las clases de la estabilidad del sistema, iteración 5.

A.1.2. Gráficos de flujo

A.1.2.1. Gráficos del enfriador

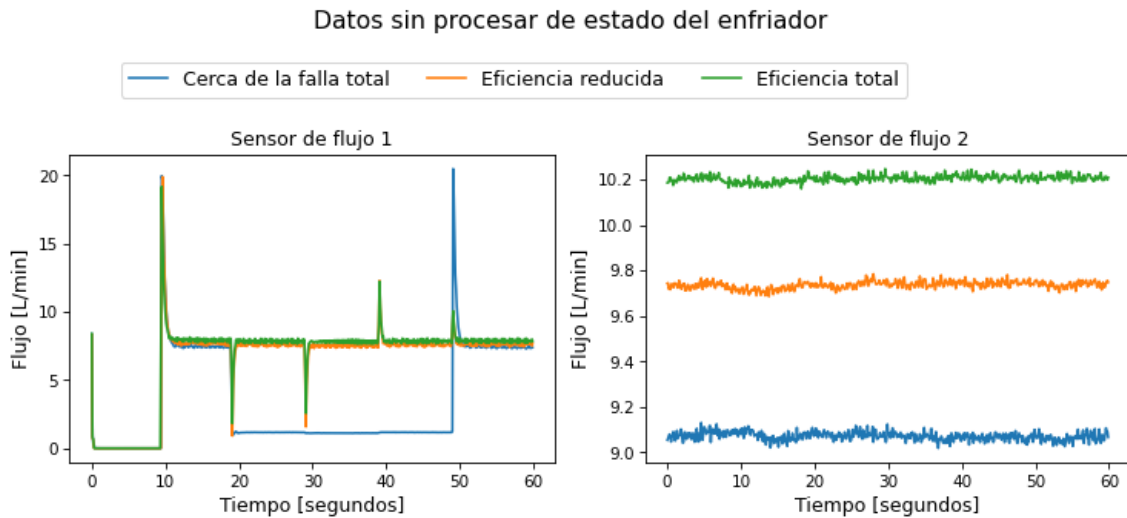


Figura A.26: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 1.

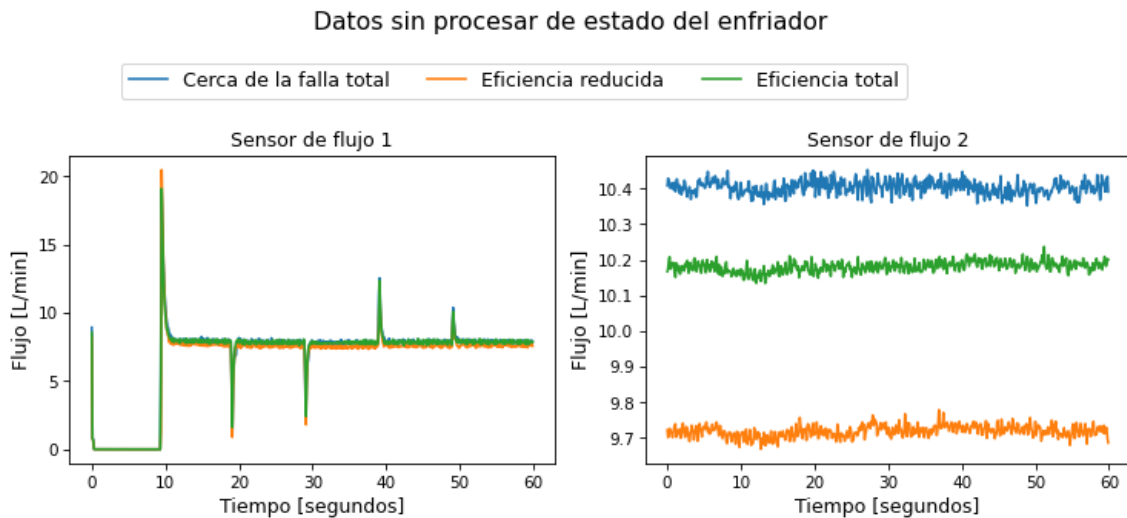


Figura A.27: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 2.

Datos sin procesar de estado del enfriador

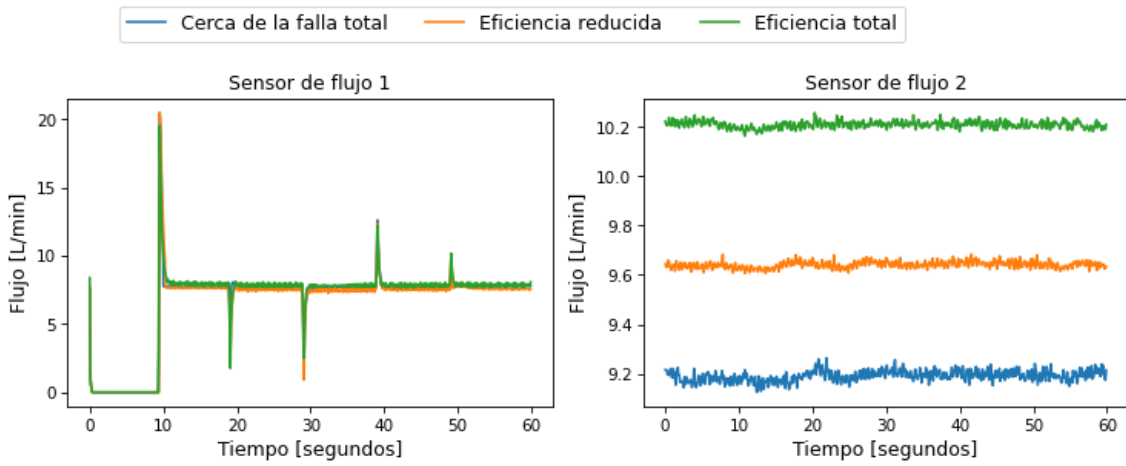


Figura A.28: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 3.

Datos sin procesar de estado del enfriador

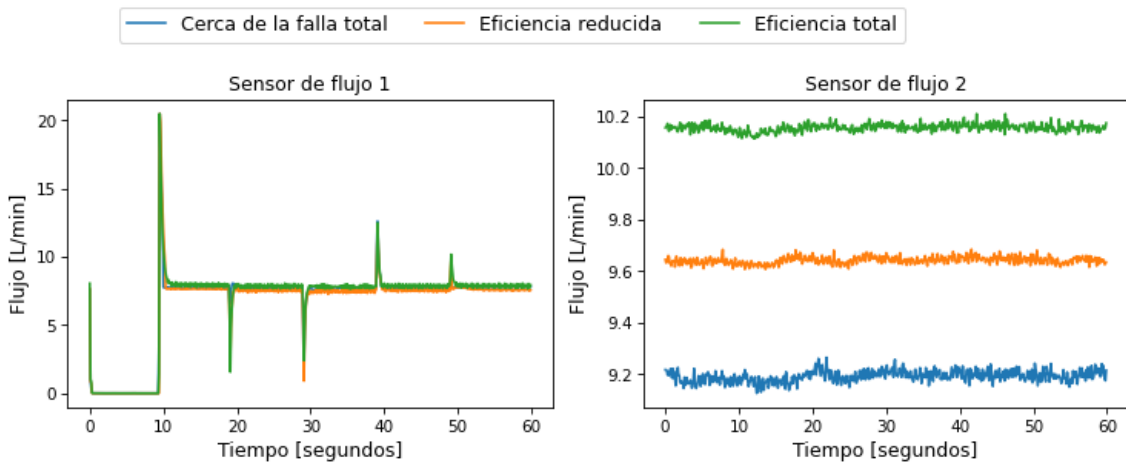


Figura A.29: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 4.

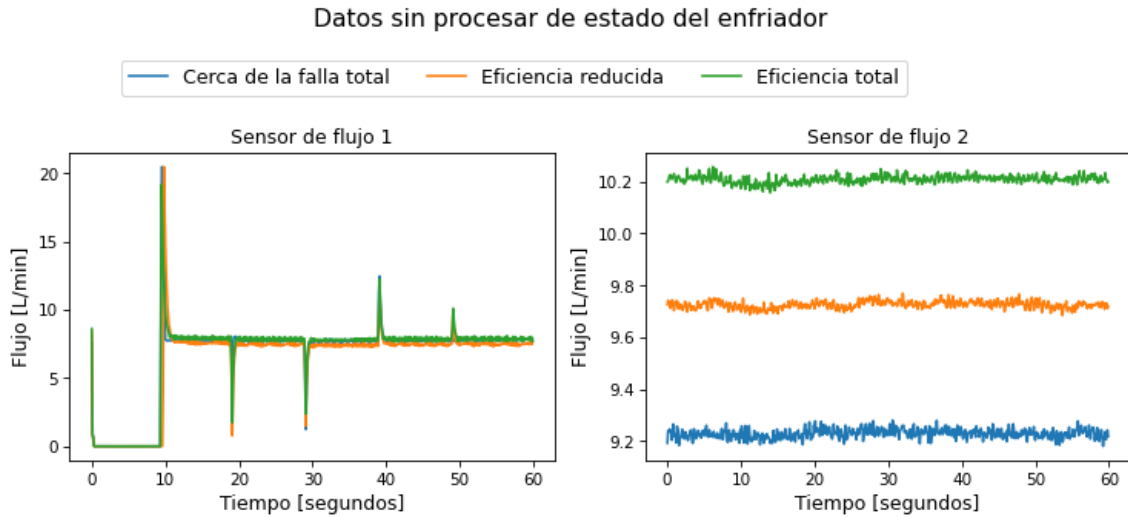


Figura A.30: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del enfriador, iteración 5.

A.1.2.2. Gráficos de la válvula

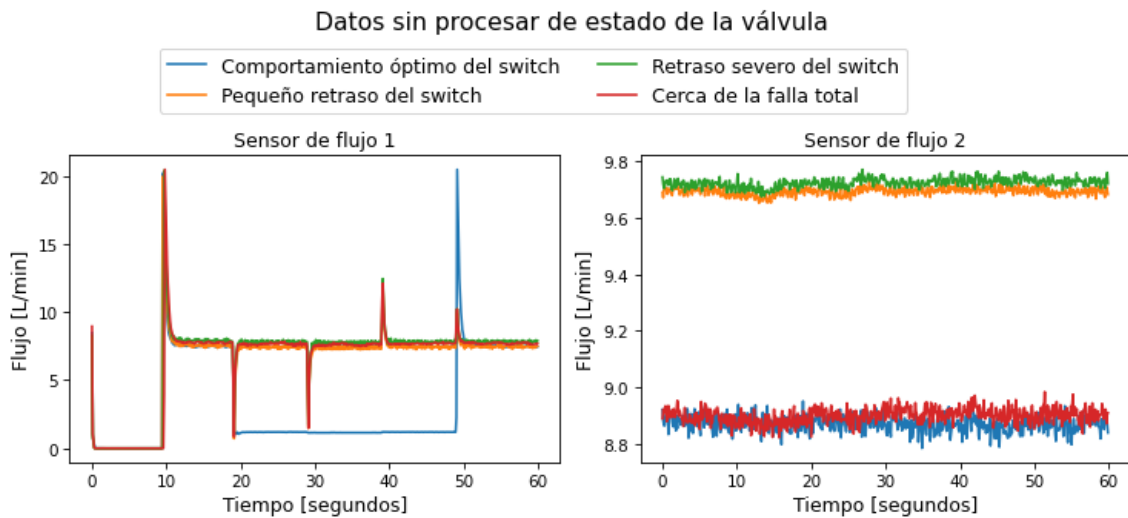


Figura A.31: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 1.

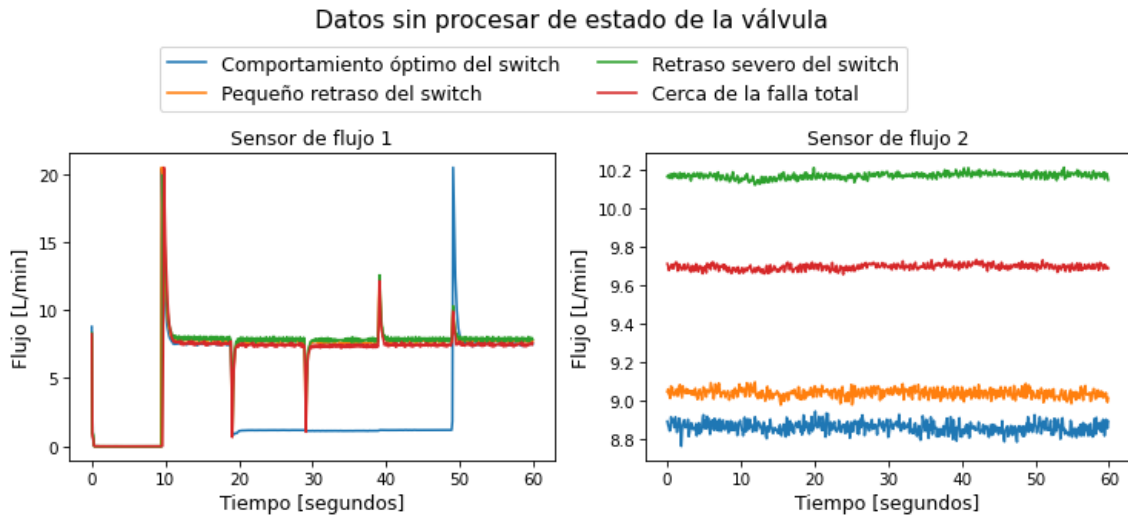


Figura A.32: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 2.

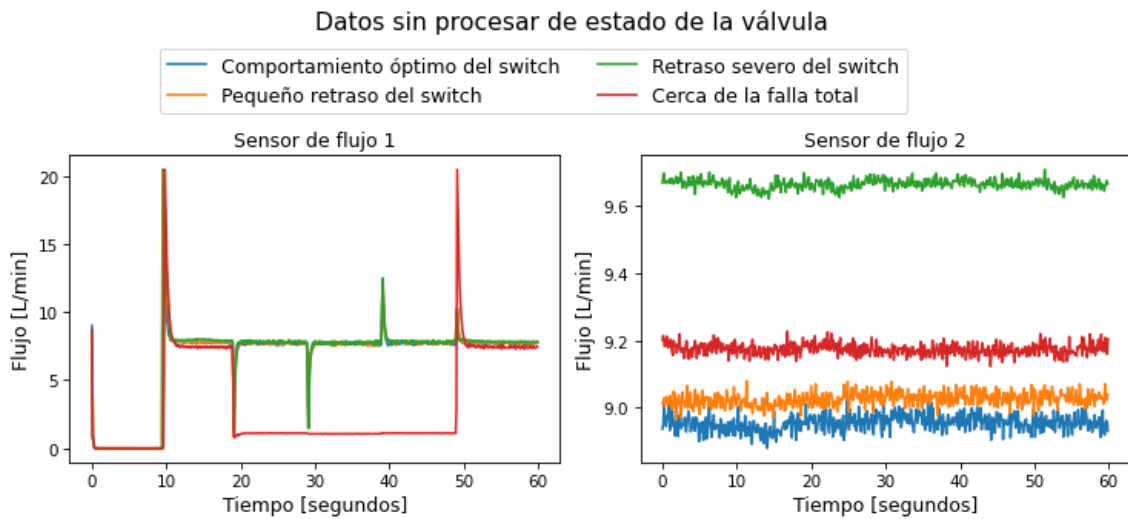


Figura A.33: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 3.

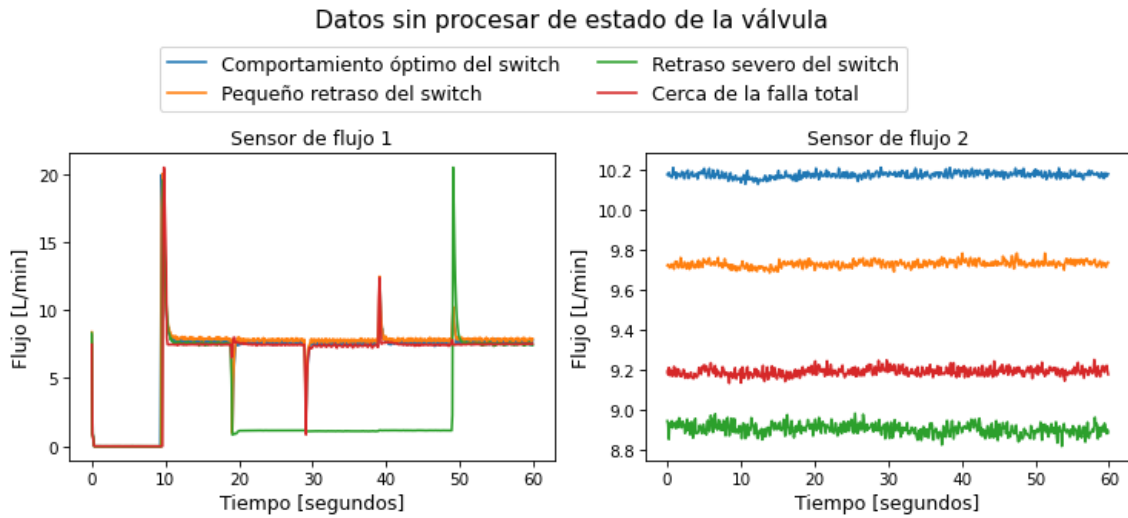


Figura A.34: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 4.

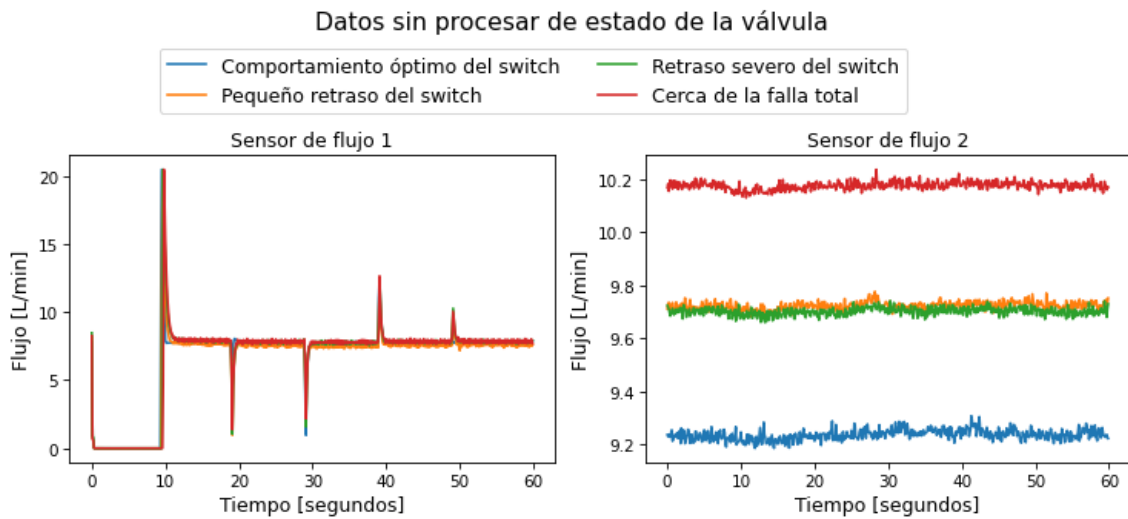


Figura A.35: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado de la válvula, iteración 5.

A.1.2.3. Gráficos de la bomba

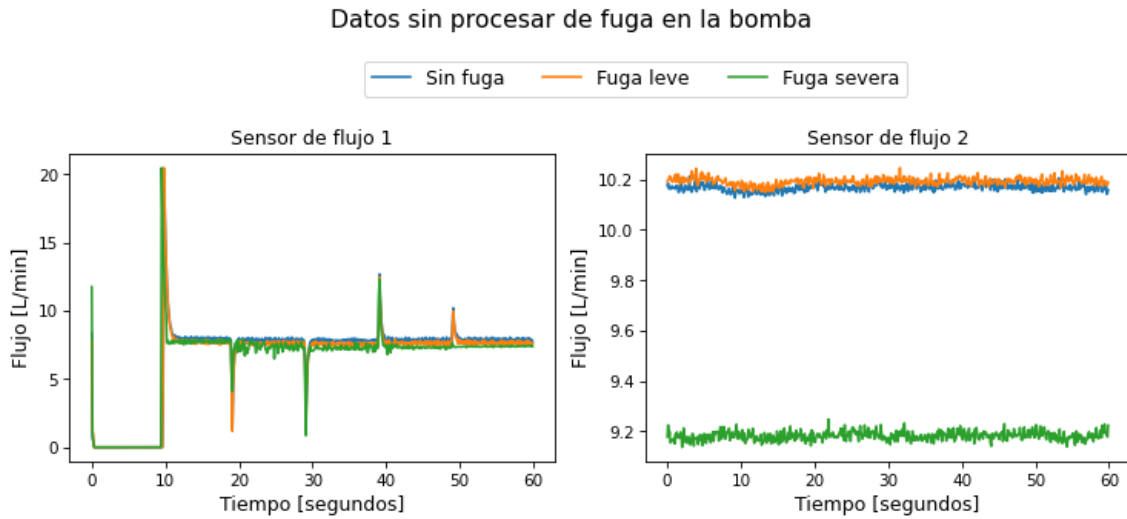


Figura A.36: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 1.

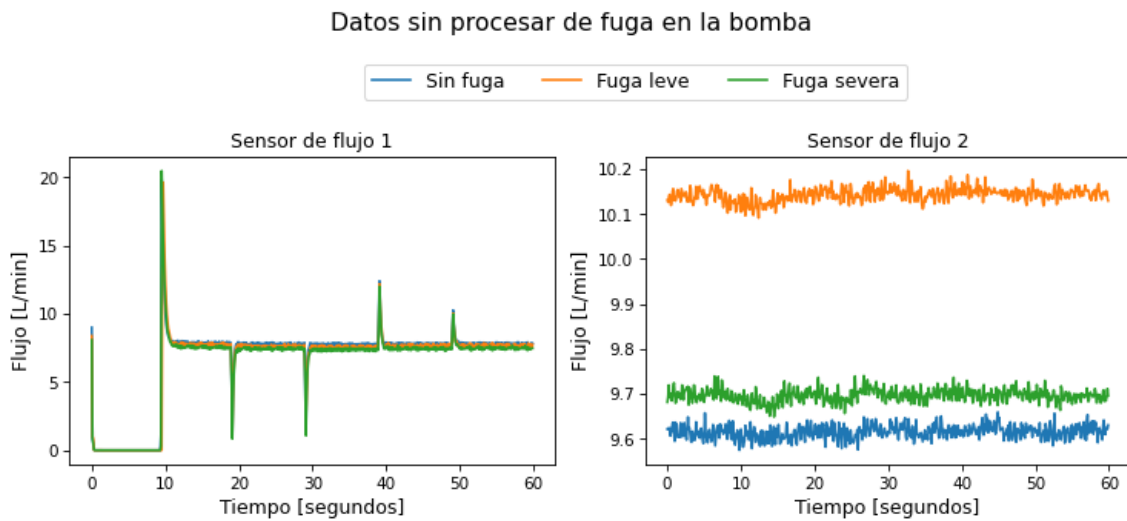


Figura A.37: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 2.

Datos sin procesar de fuga en la bomba

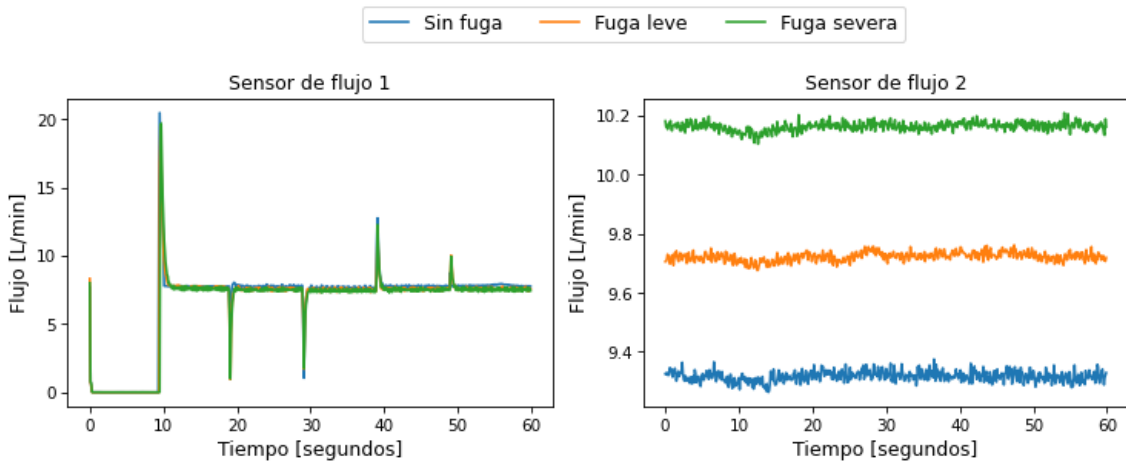


Figura A.38: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 3.

Datos sin procesar de fuga en la bomba

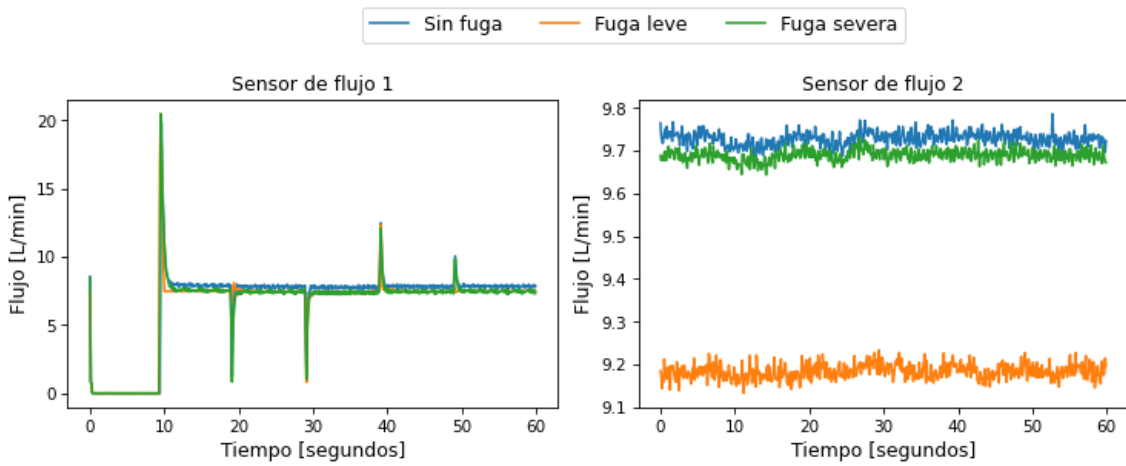


Figura A.39: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 4.

Datos sin procesar de fuga en la bomba

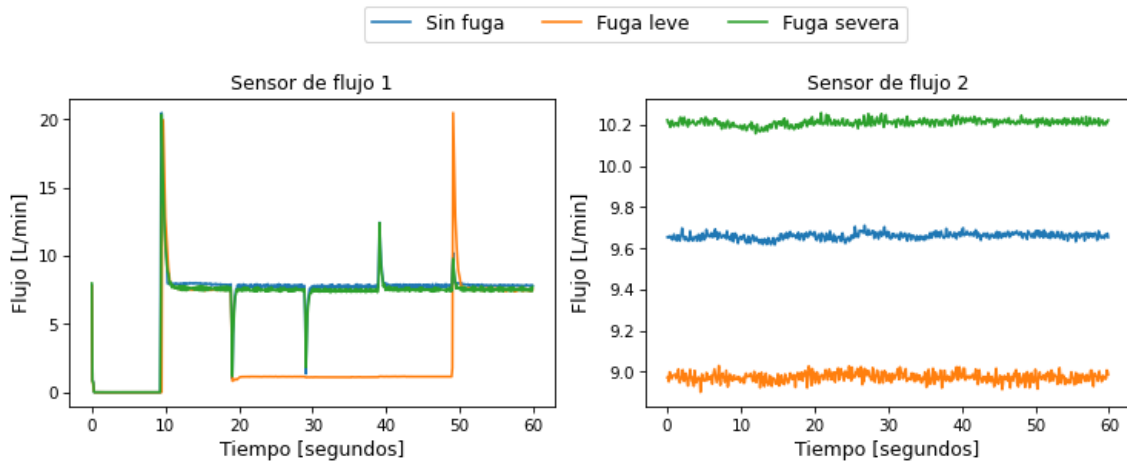


Figura A.40: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la fuga en la bomba, iteración 5.

A.1.2.4. Gráficos del acumulador

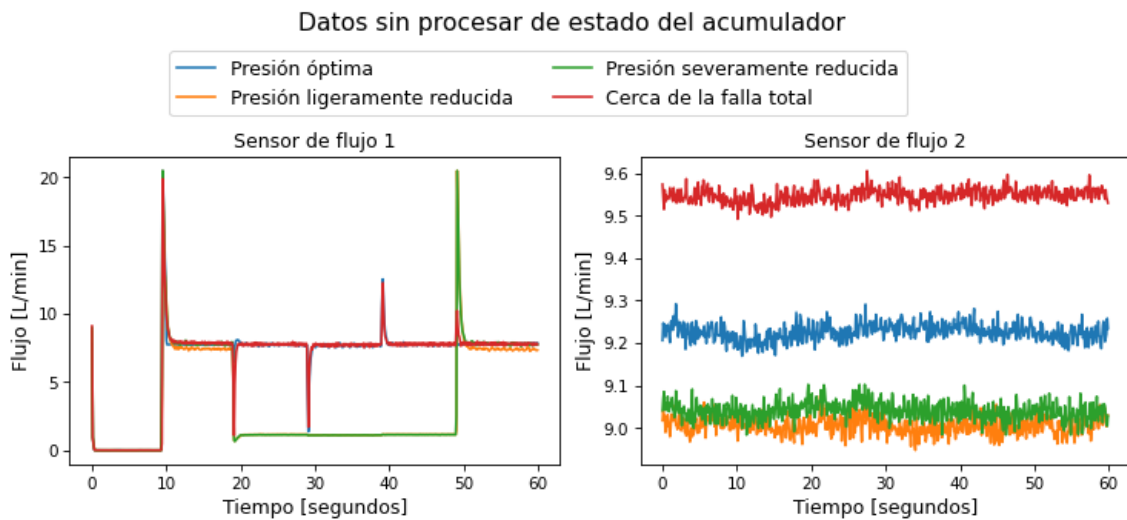


Figura A.41: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 1.

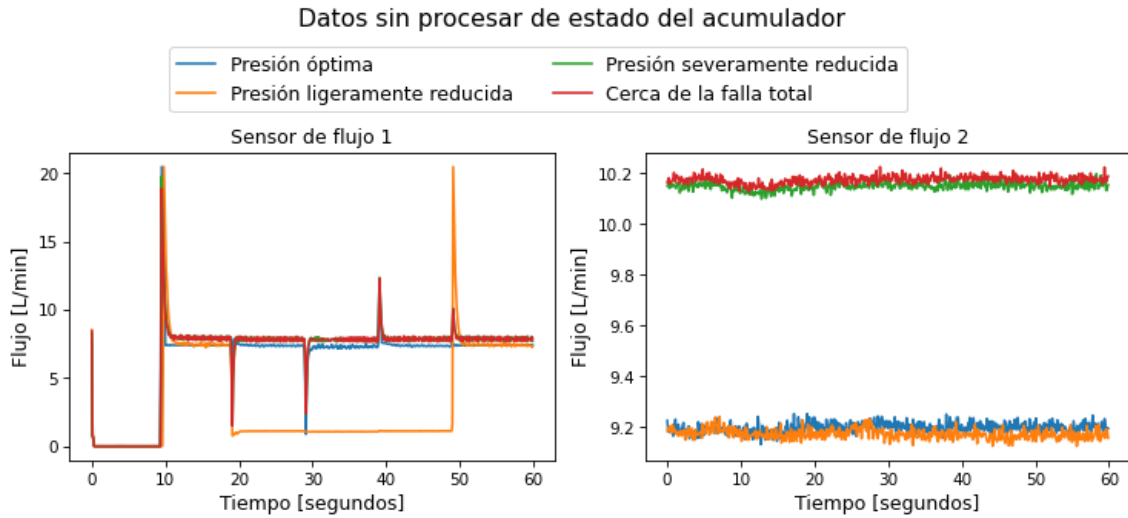


Figura A.42: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 2.

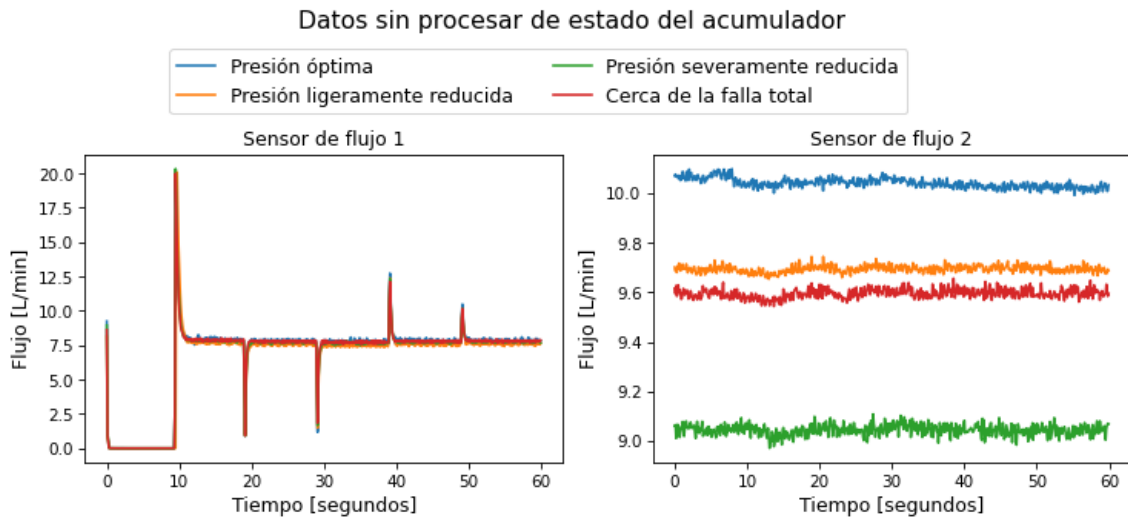


Figura A.43: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 3.

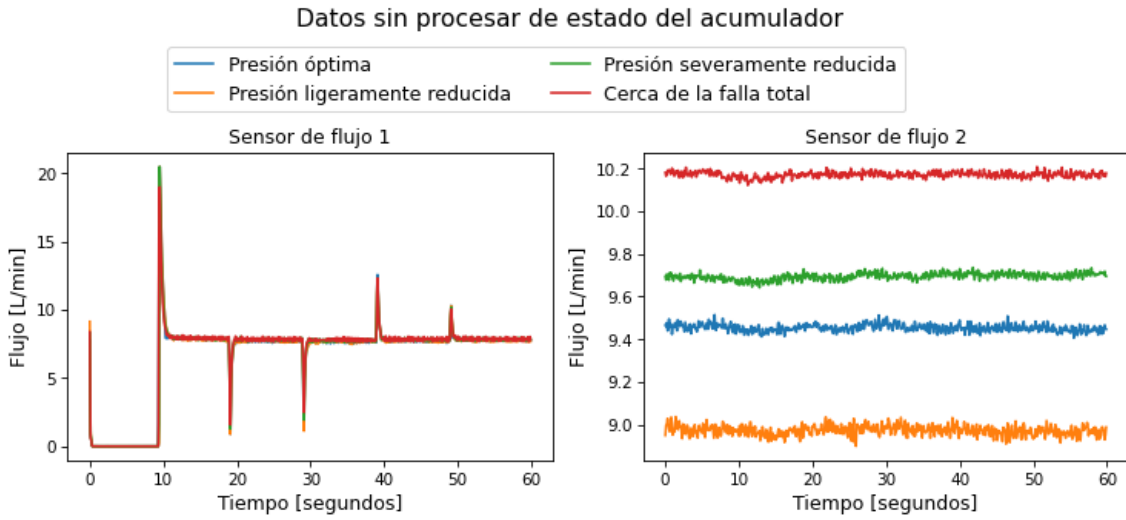


Figura A.44: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 4.

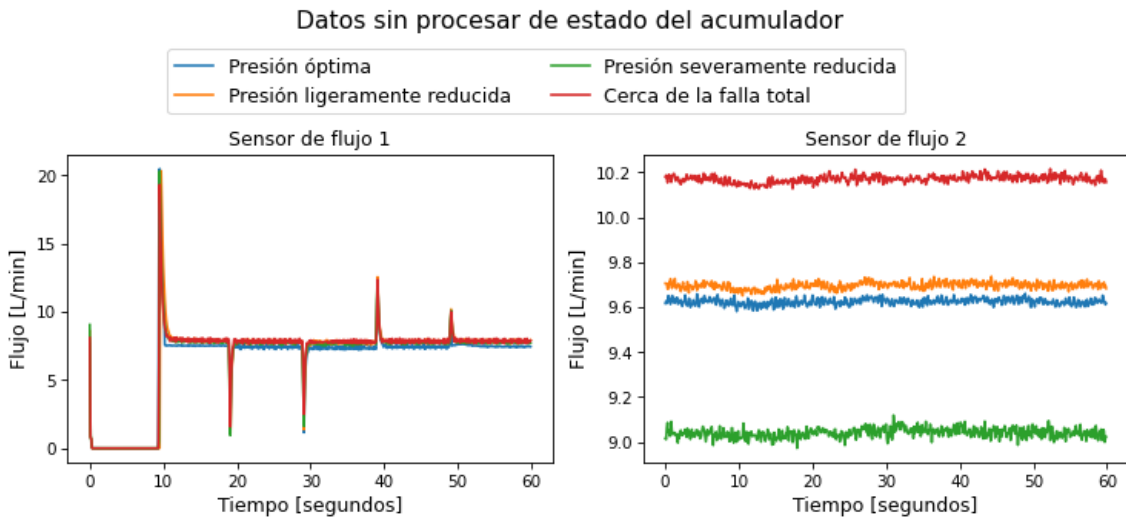


Figura A.45: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases del estado del acumulador, iteración 5.

A.1.2.5. Gráficos de la estabilidad del sistema

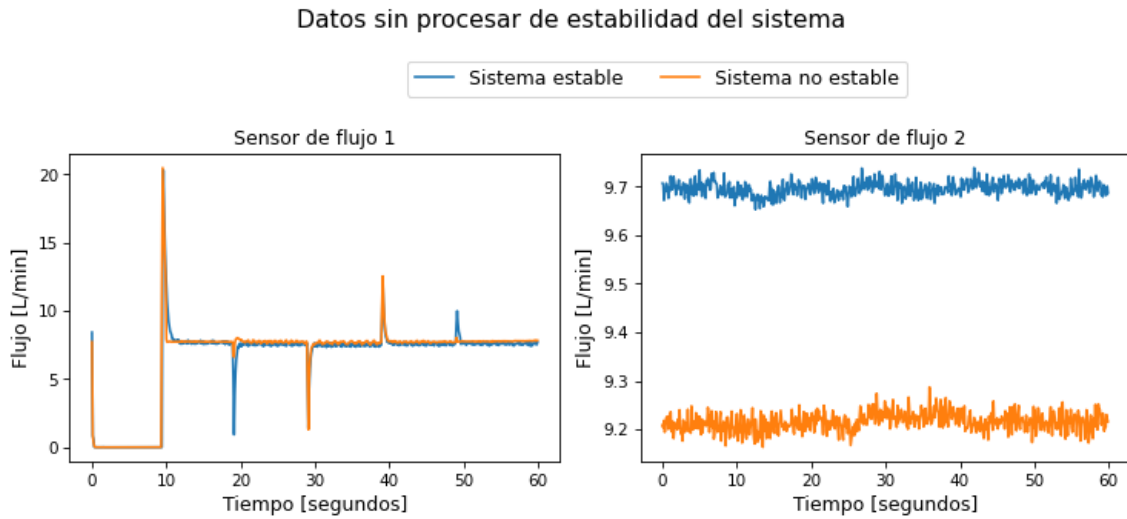


Figura A.46: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 1.

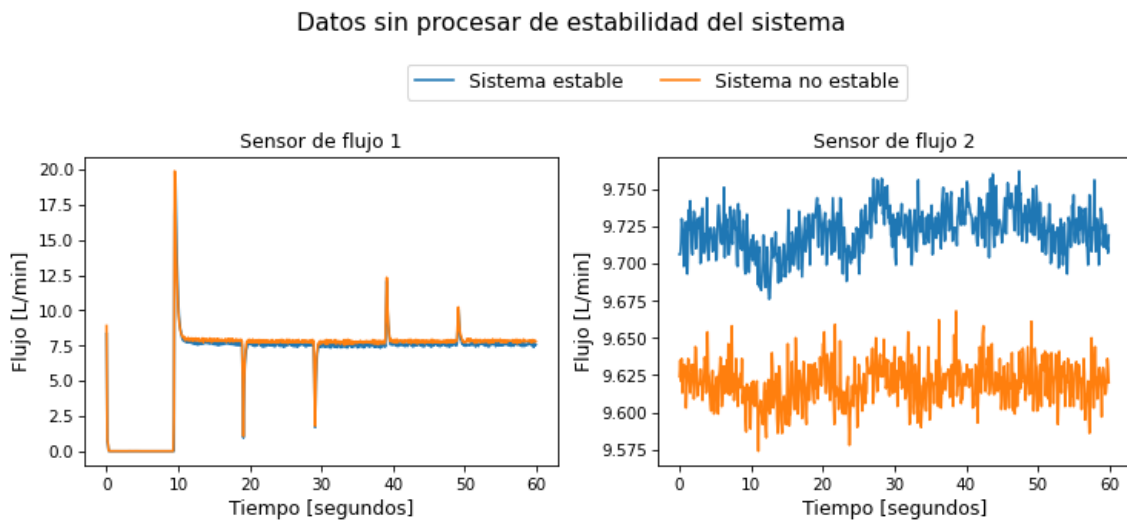


Figura A.47: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 2.

Datos sin procesar de estabilidad del sistema

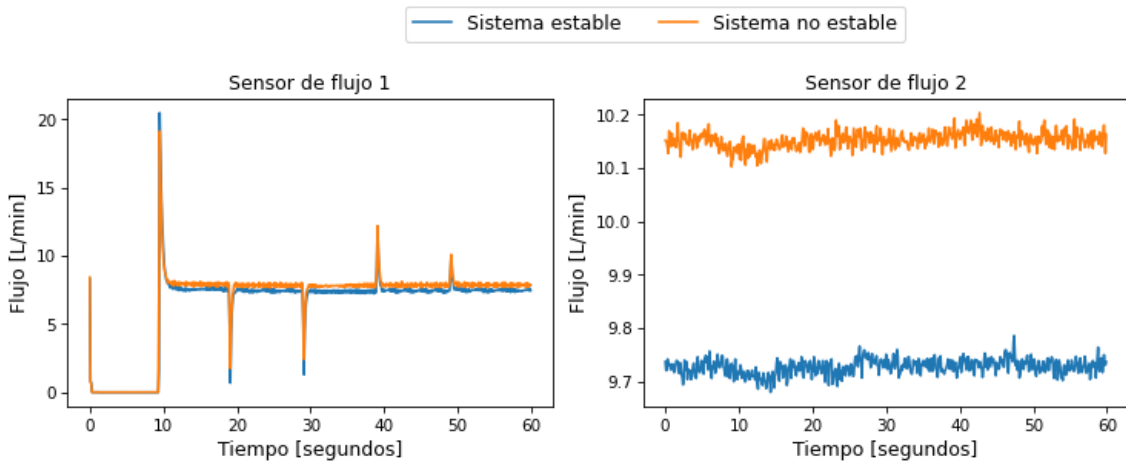


Figura A.48: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 3.

Datos sin procesar de estabilidad del sistema

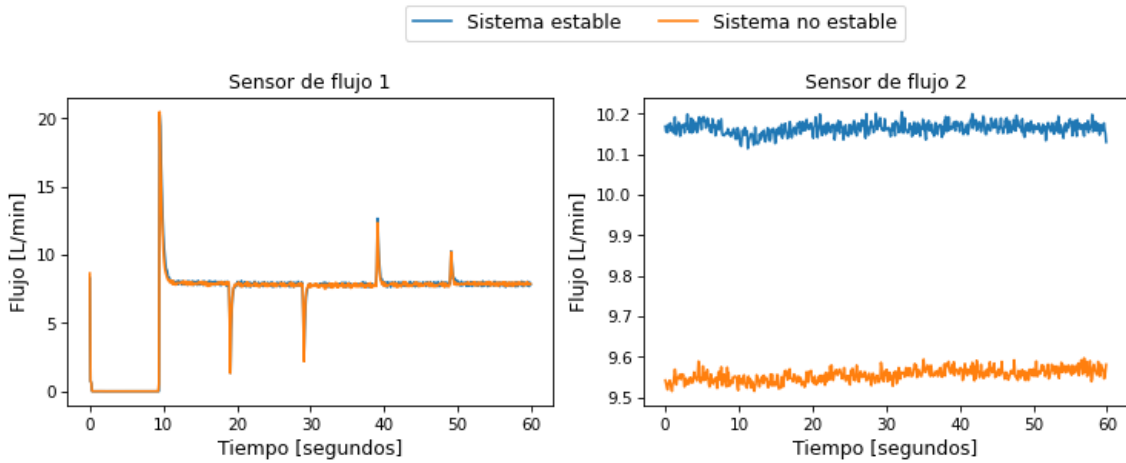


Figura A.49: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 4.

Datos sin procesar de estabilidad del sistema

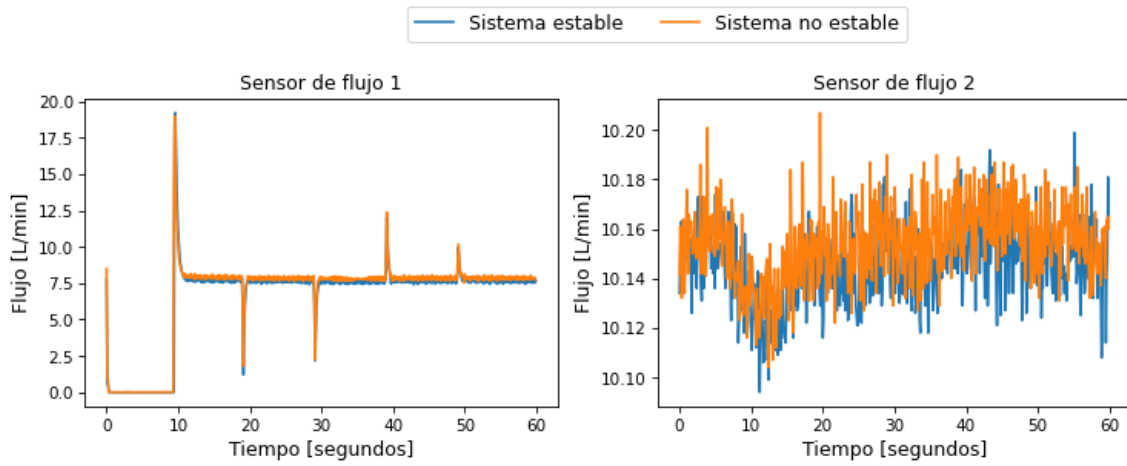


Figura A.50: Gráficos de los datos sin procesar obtenidos por los sensores de flujo, separando según las clases de la estabilidad del sistema, iteración 5.

A.1.3. Gráficos de presión

A.1.3.1. Gráficos del enfriador

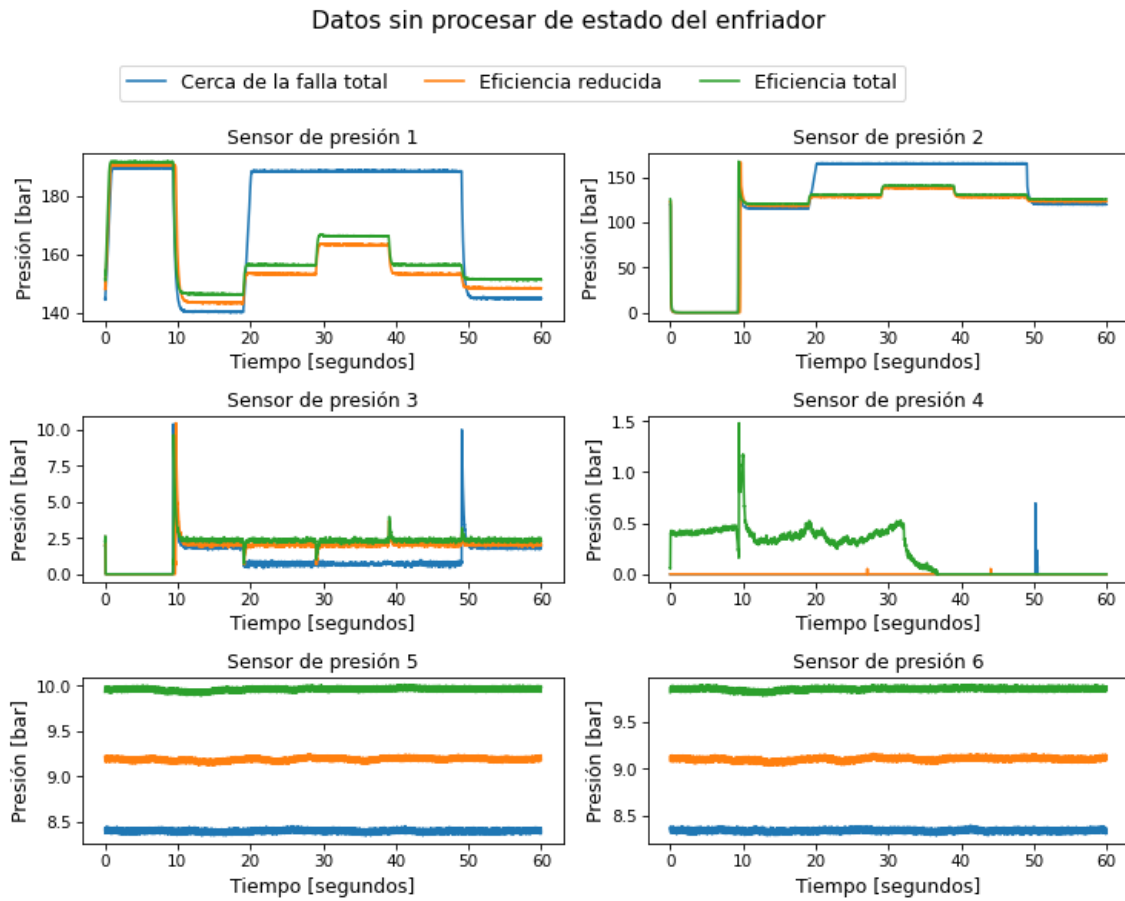


Figura A.51: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 1.

Datos sin procesar de estado del enfriador

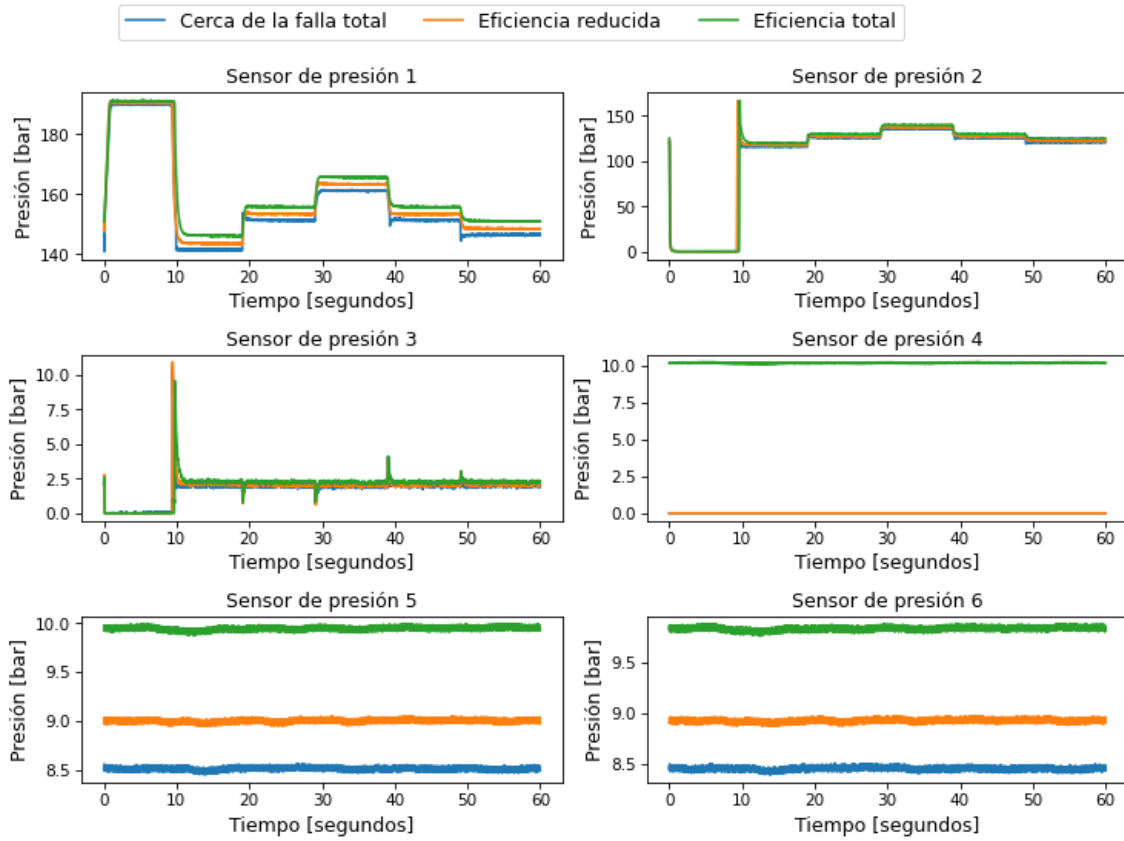


Figura A.52: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 2.

Datos sin procesar de estado del enfriador

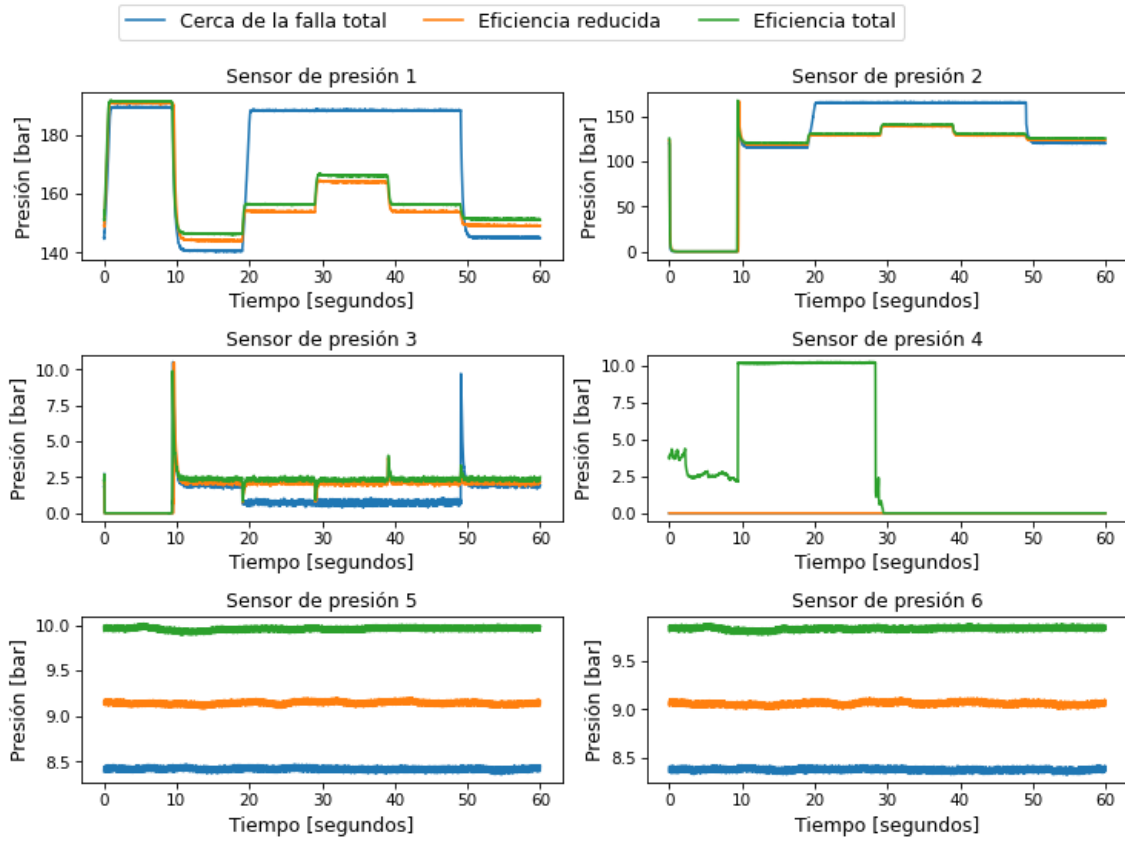


Figura A.53: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 3.

Datos sin procesar de estado del enfriador

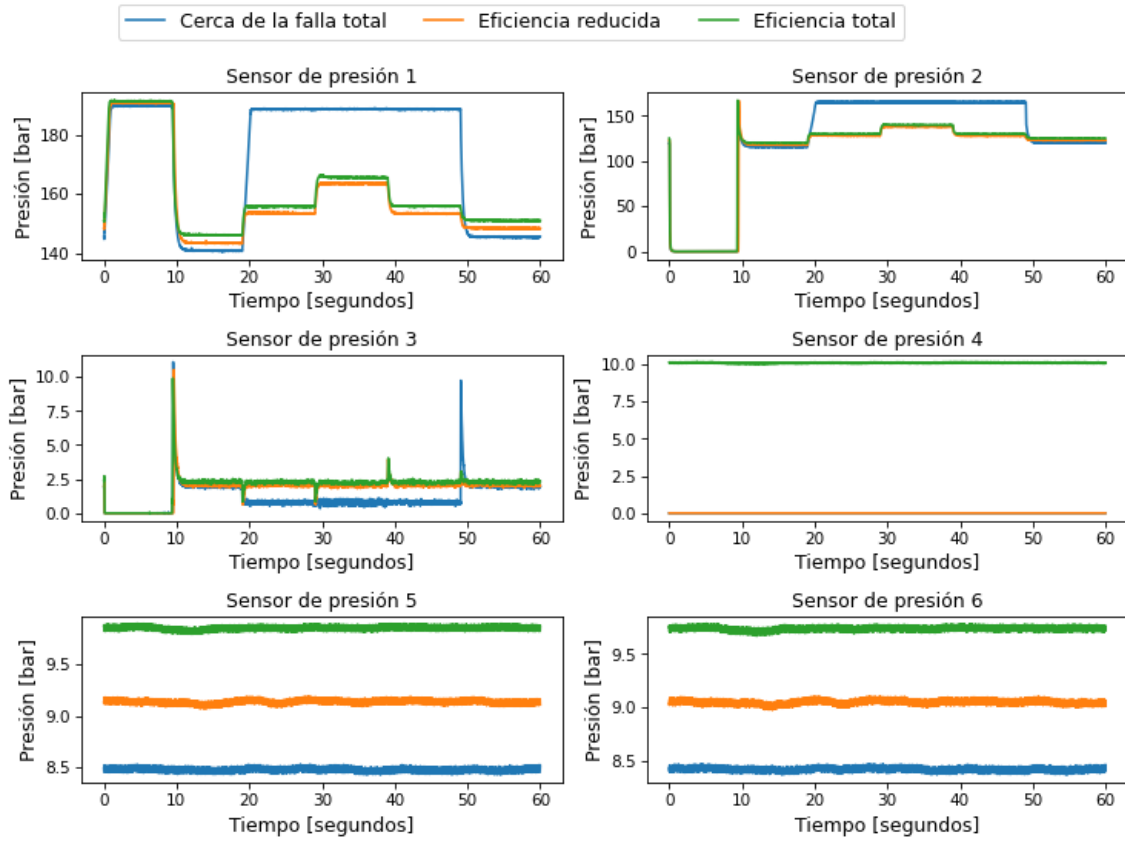


Figura A.54: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 4.

Datos sin procesar de estado del enfriador

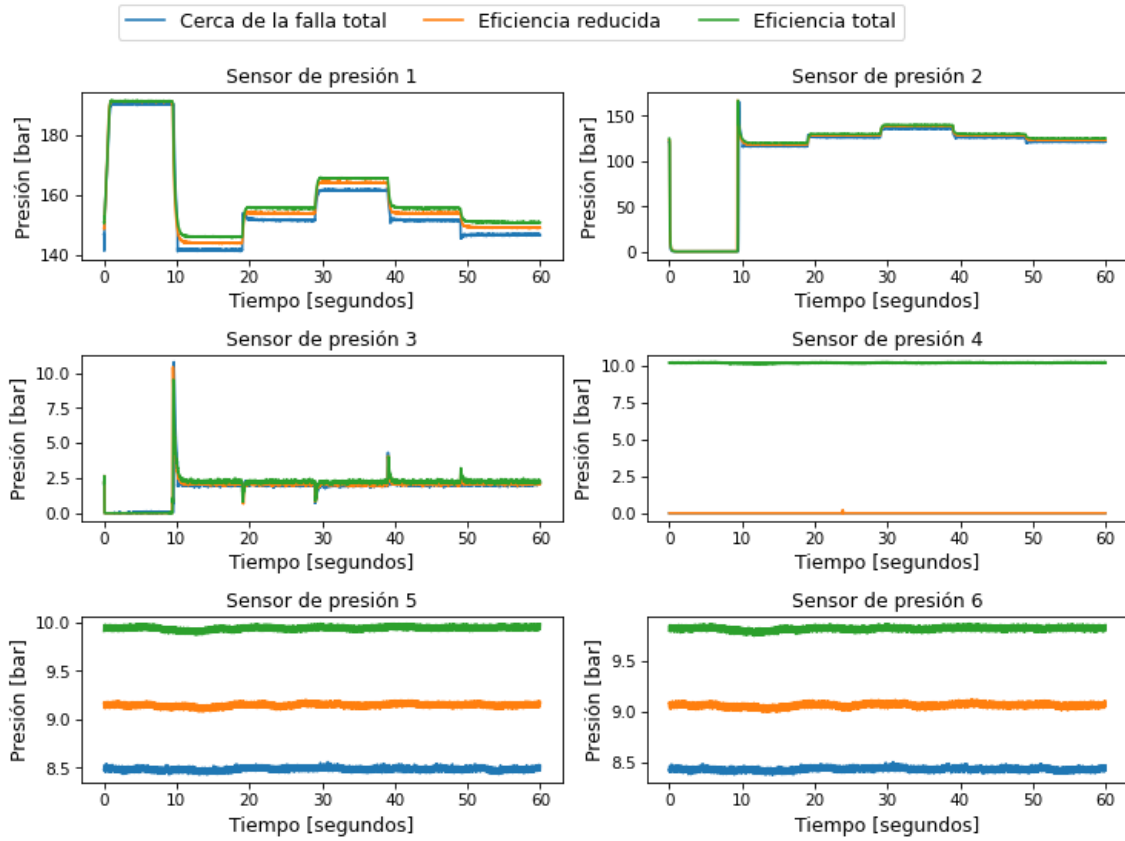


Figura A.55: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del enfriador, iteración 5.

A.1.3.2. Gráficos de la válvula

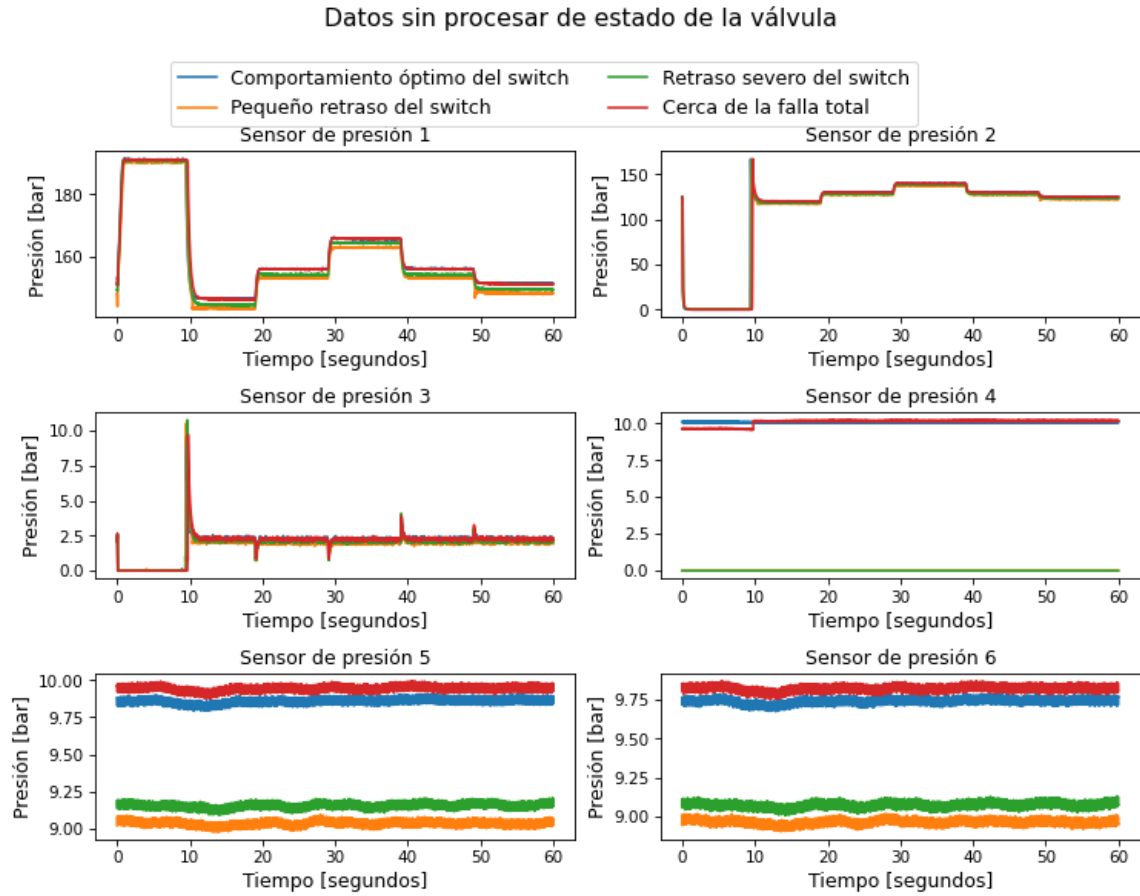


Figura A.56: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 1.

Datos sin procesar de estado de la válvula

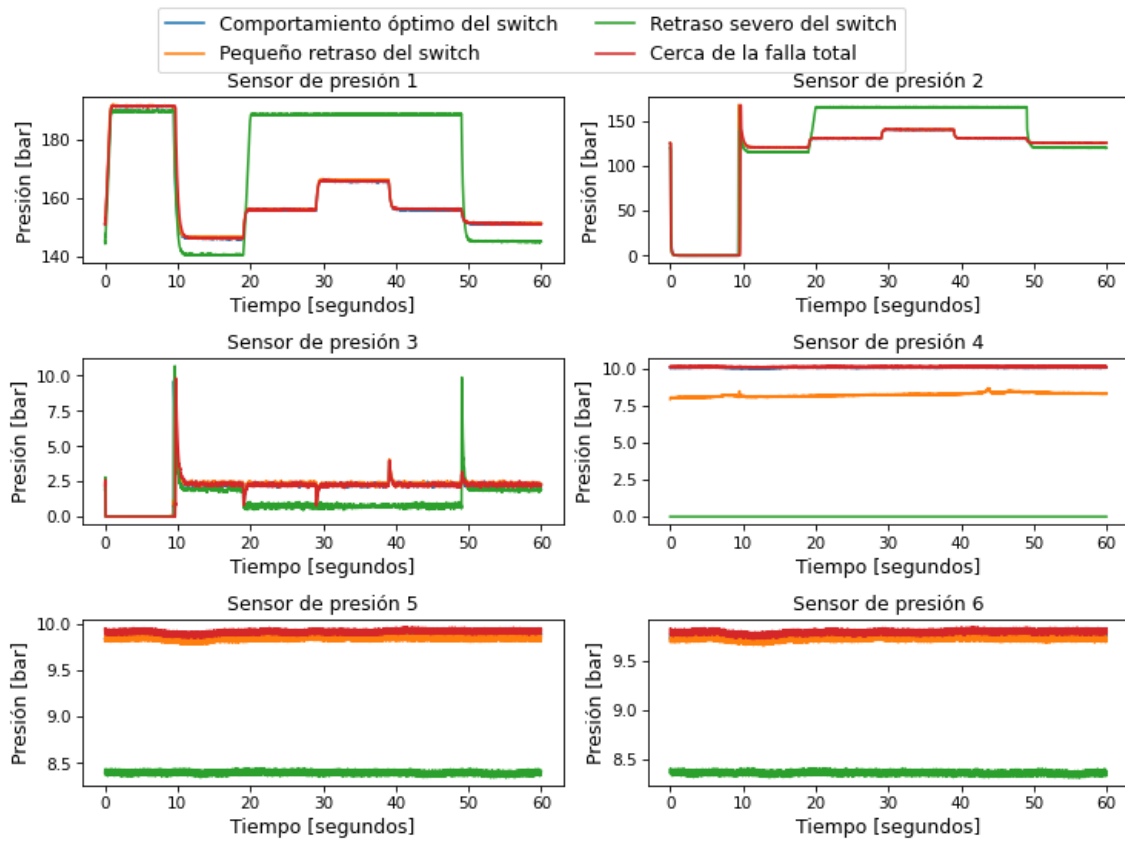


Figura A.57: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 2.

Datos sin procesar de estado de la válvula

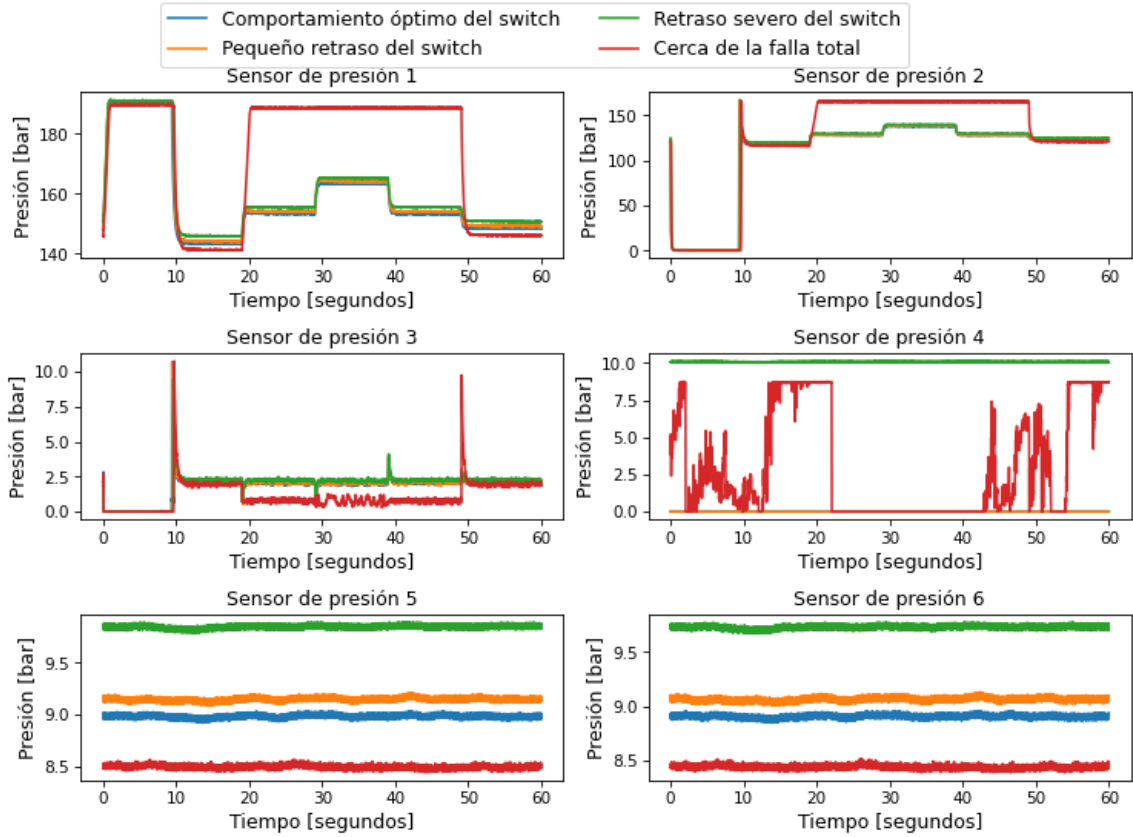


Figura A.58: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 3.

Datos sin procesar de estado de la válvula

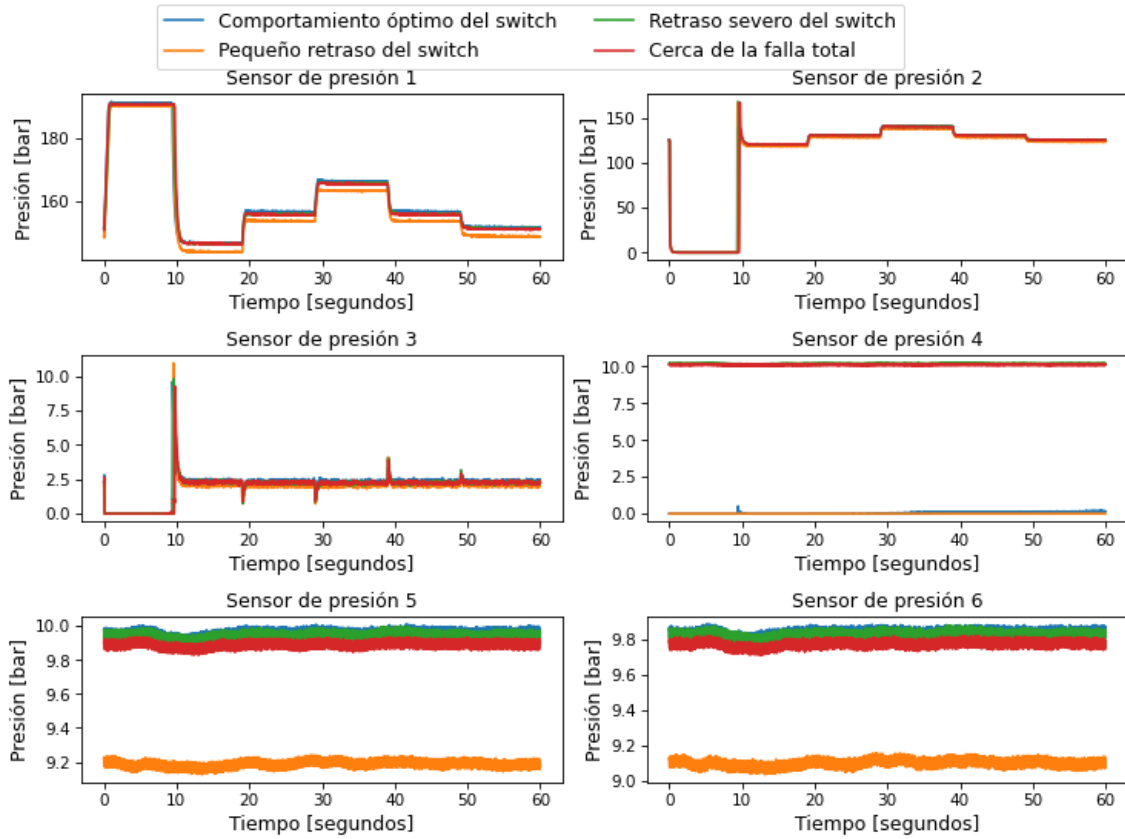


Figura A.59: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 4.

Datos sin procesar de estado de la válvula

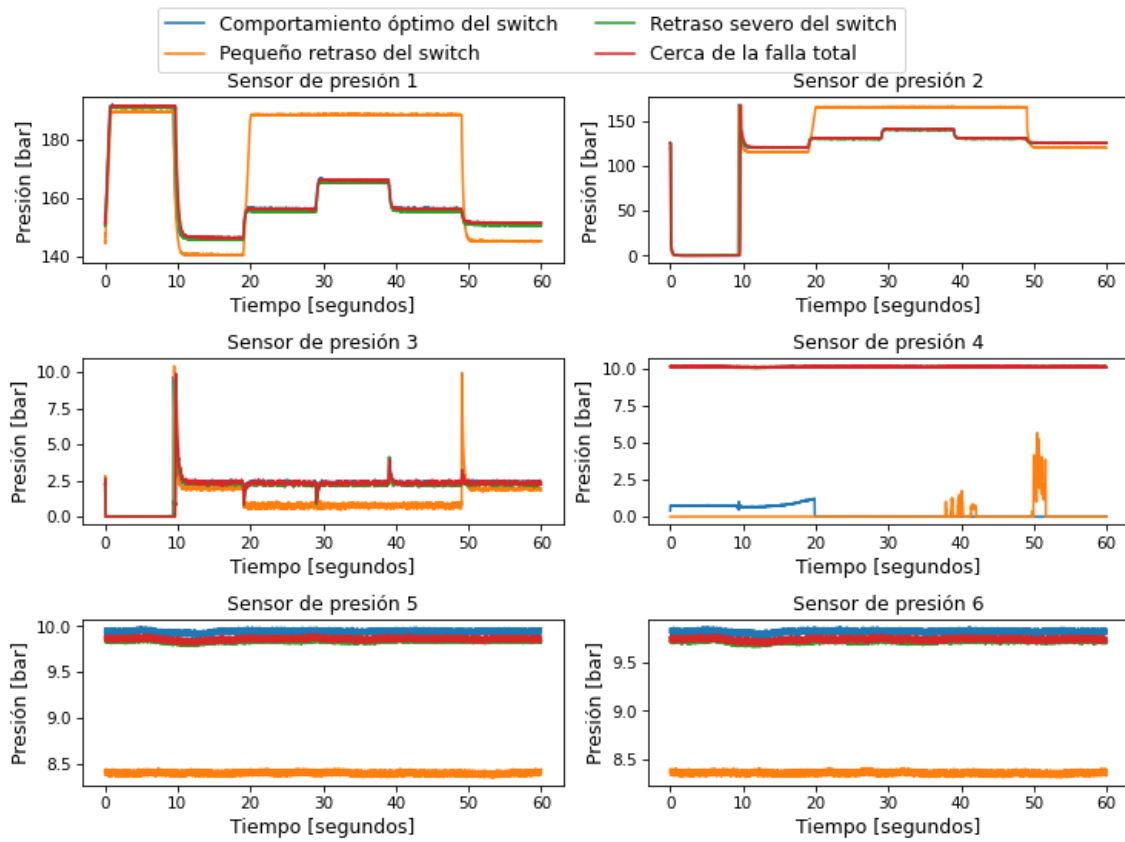


Figura A.60: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado de la válvula, iteración 5.

A.1.3.3. Gráficos de la bomba

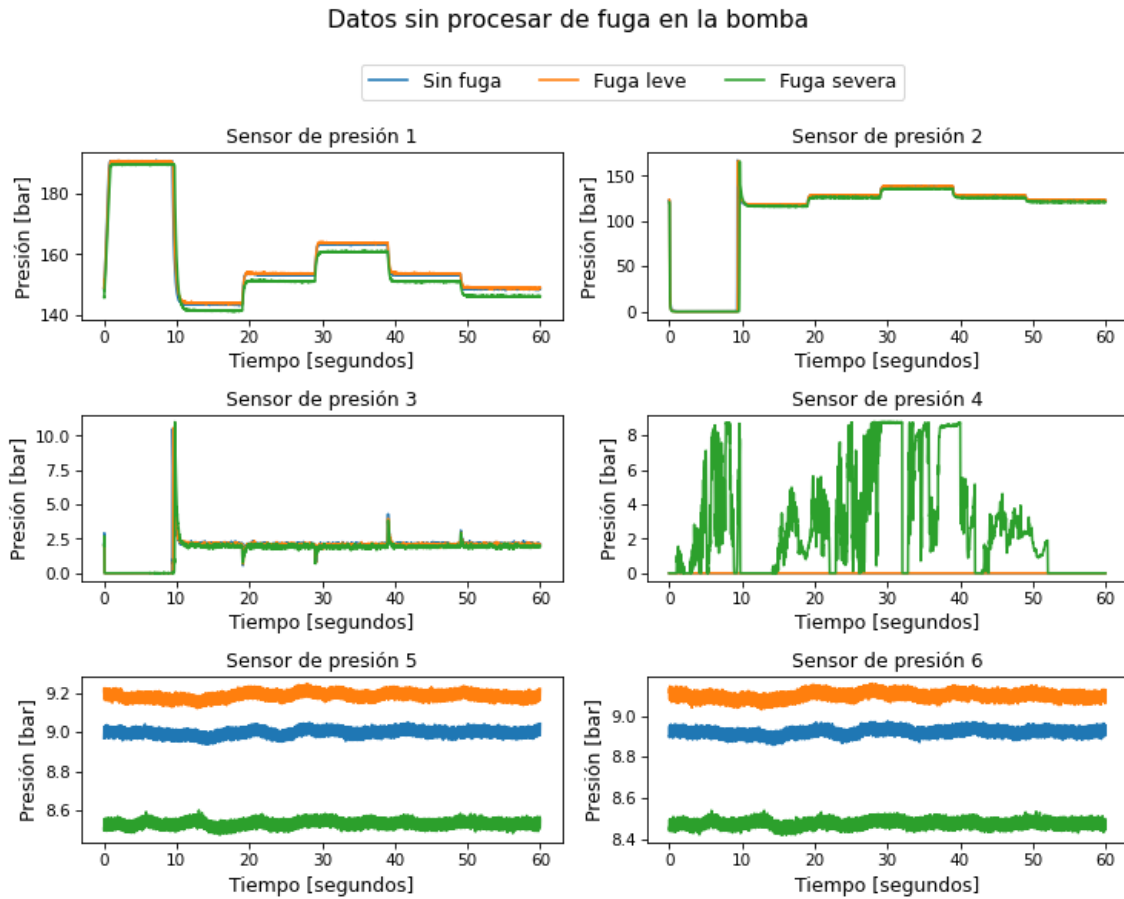


Figura A.61: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 1.

Datos sin procesar de fuga en la bomba

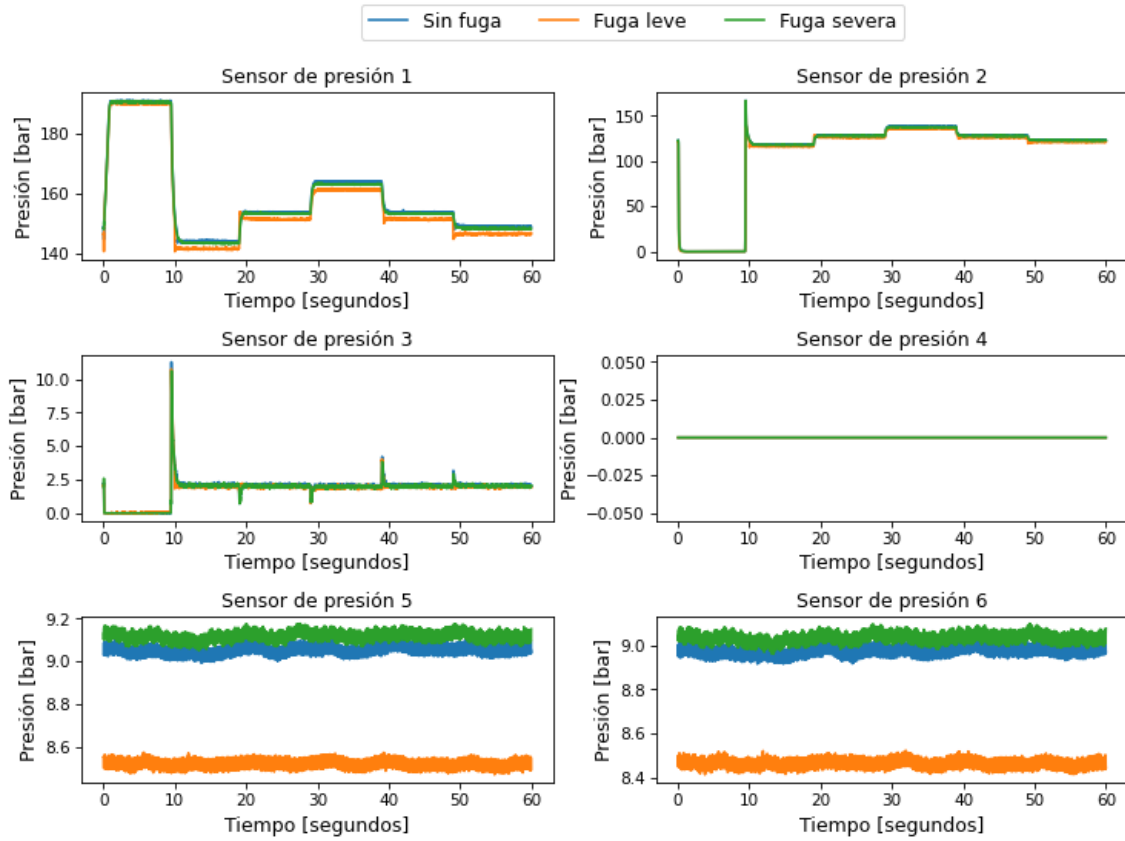


Figura A.62: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 2.

Datos sin procesar de fuga en la bomba

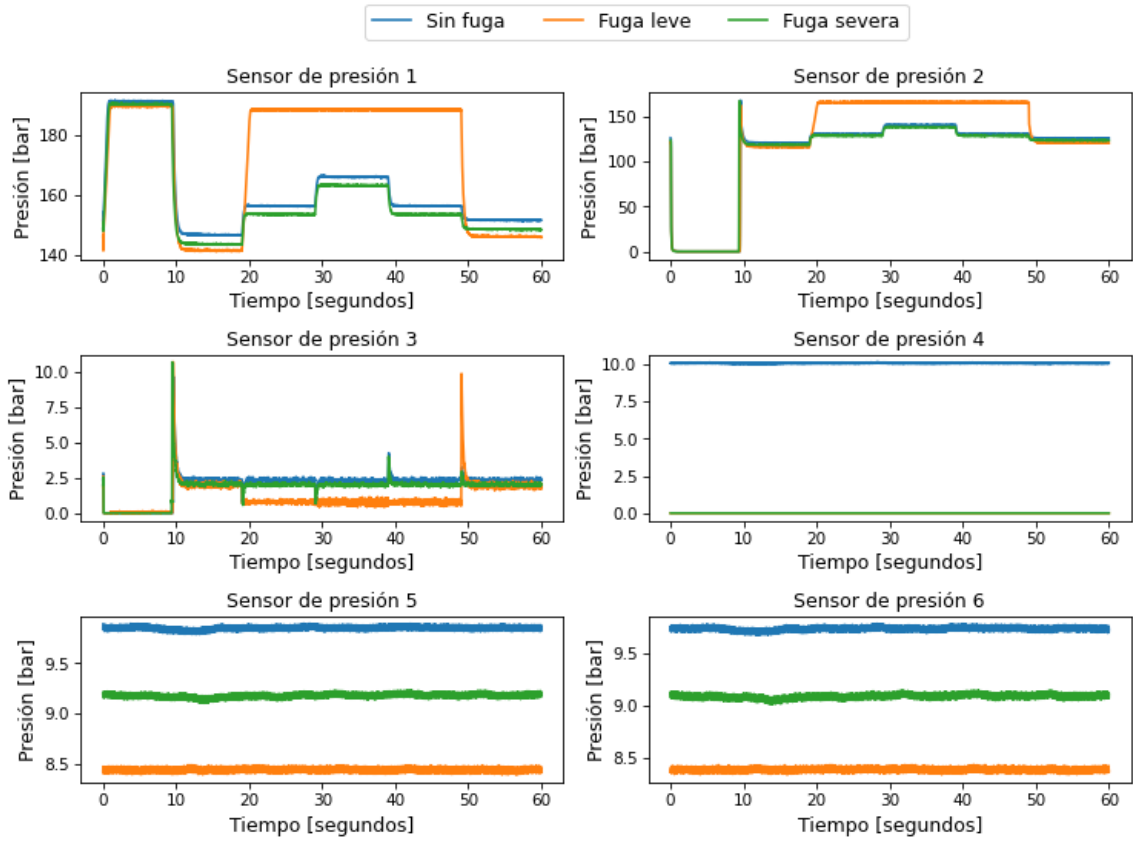


Figura A.63: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 3.

Datos sin procesar de fuga en la bomba

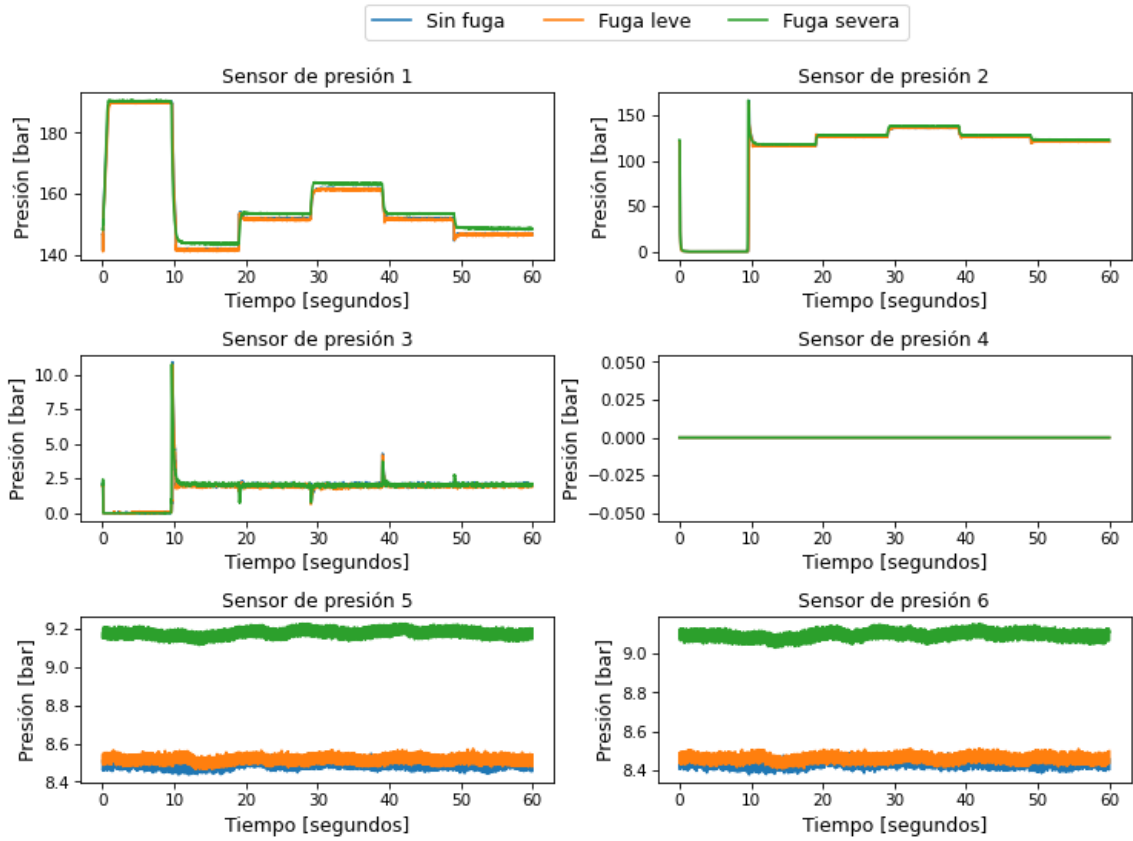


Figura A.64: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 4.

Datos sin procesar de fuga en la bomba

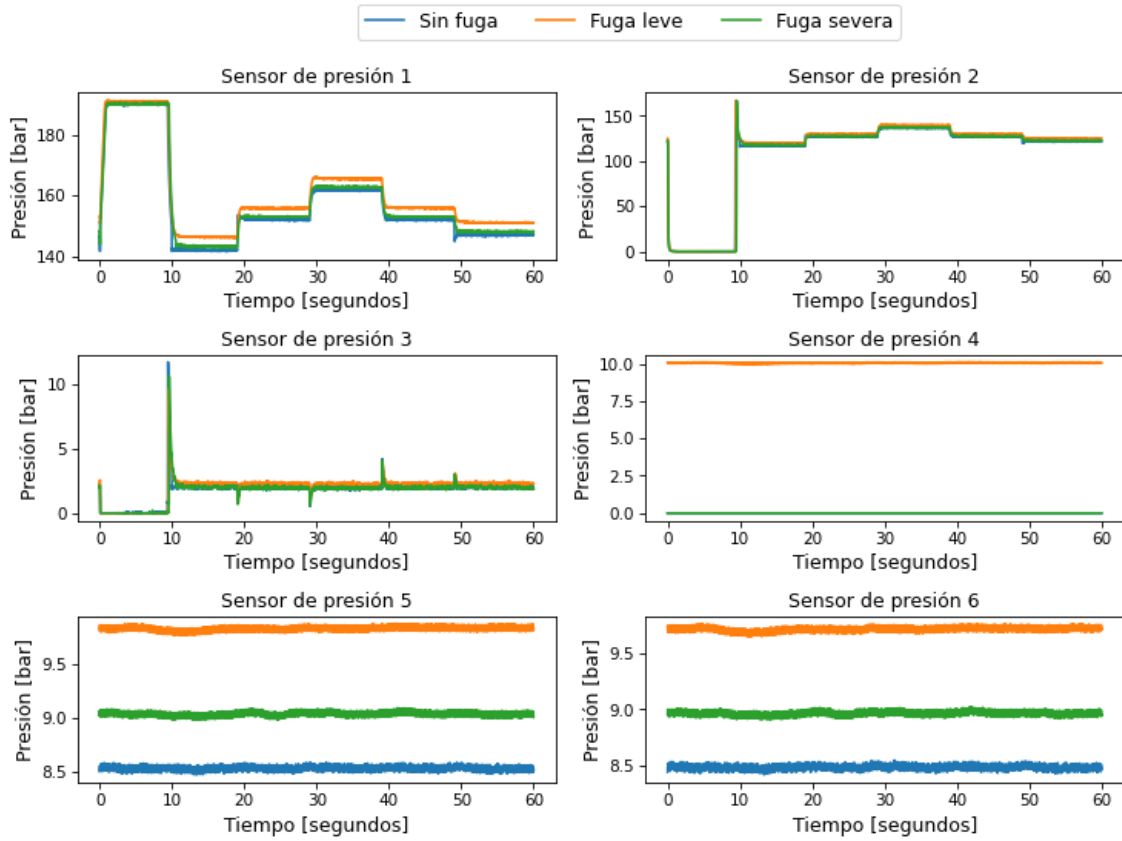


Figura A.65: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la fuga en la bomba, iteración 5.

A.1.3.4. Gráficos del acumulador

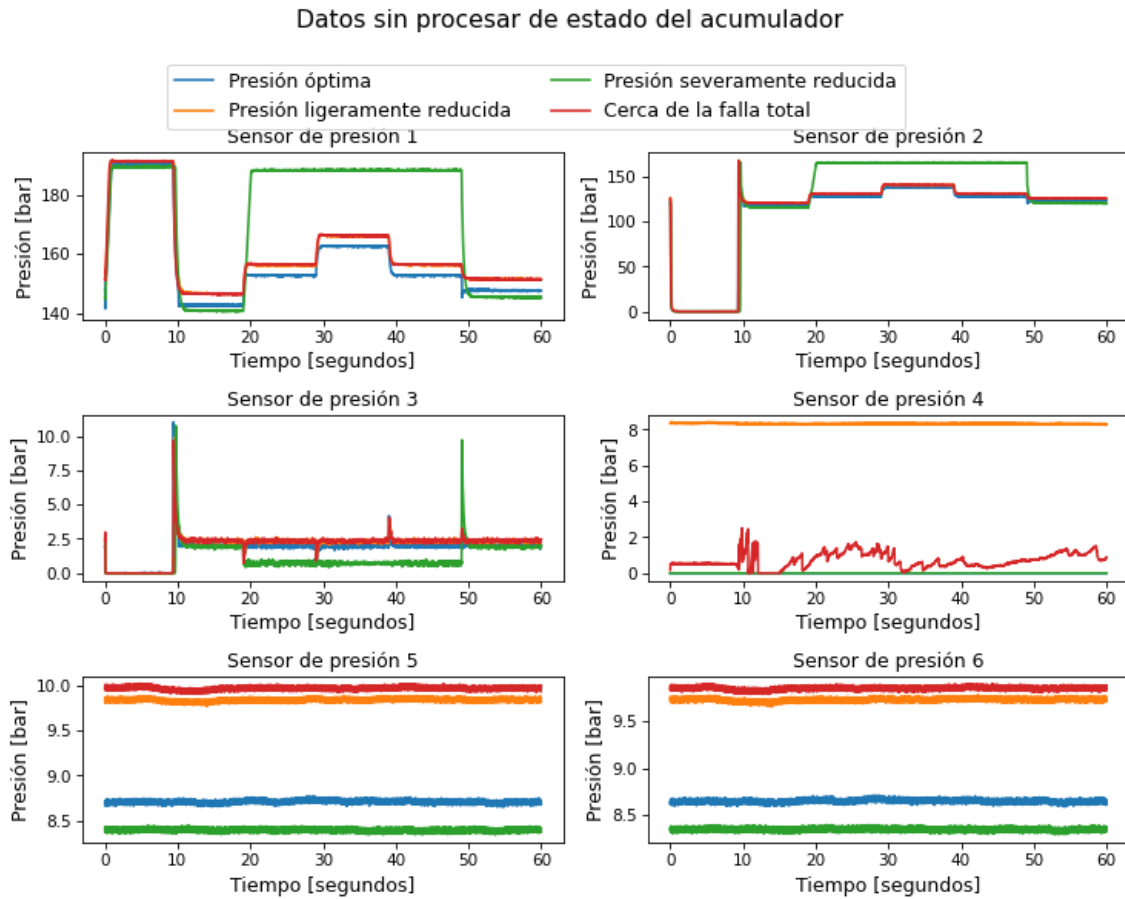


Figura A.66: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 1.

Datos sin procesar de estado del acumulador

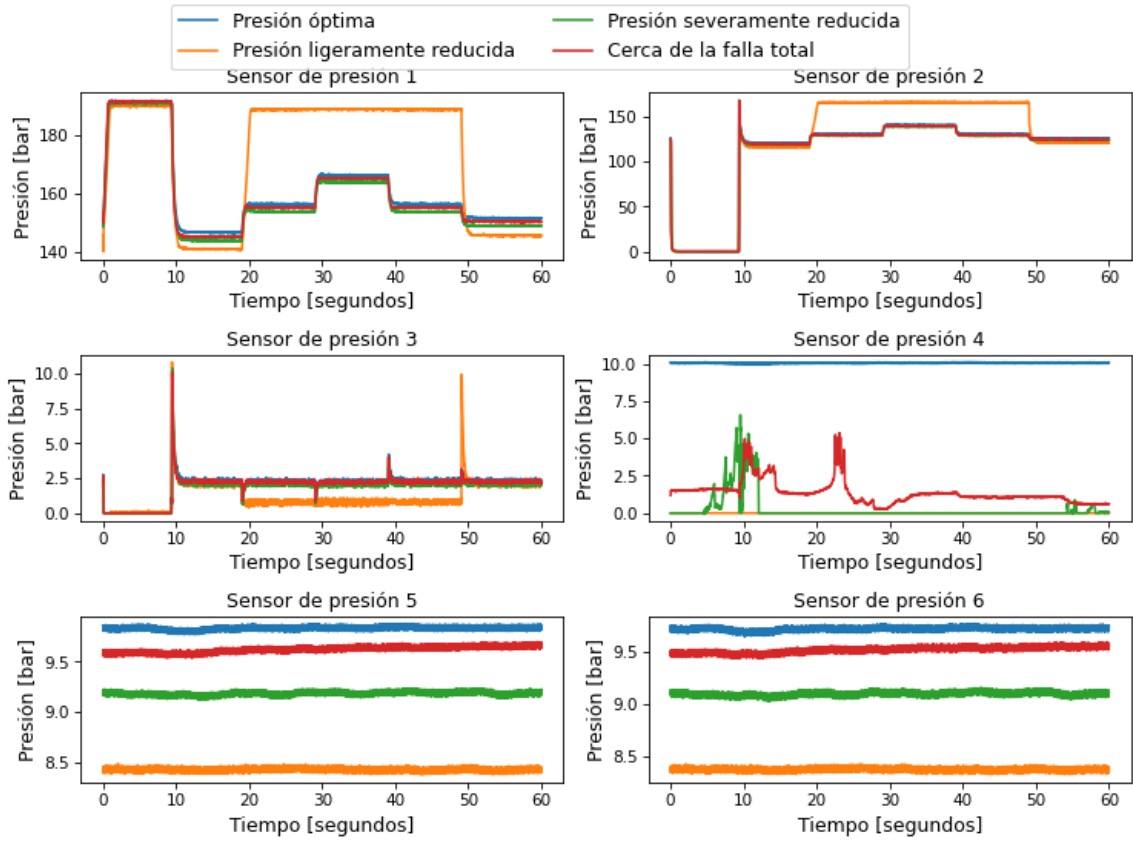


Figura A.67: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 2.

Datos sin procesar de estado del acumulador

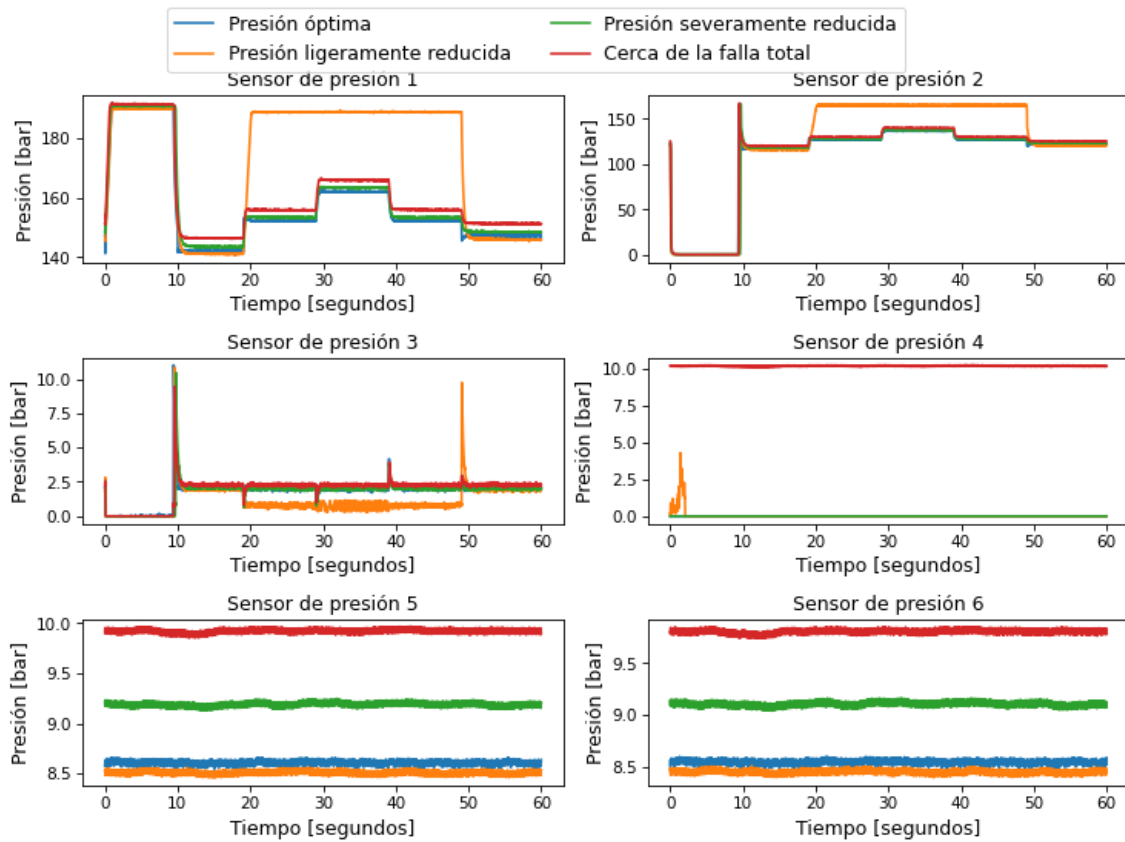


Figura A.68: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 3.

Datos sin procesar de estado del acumulador

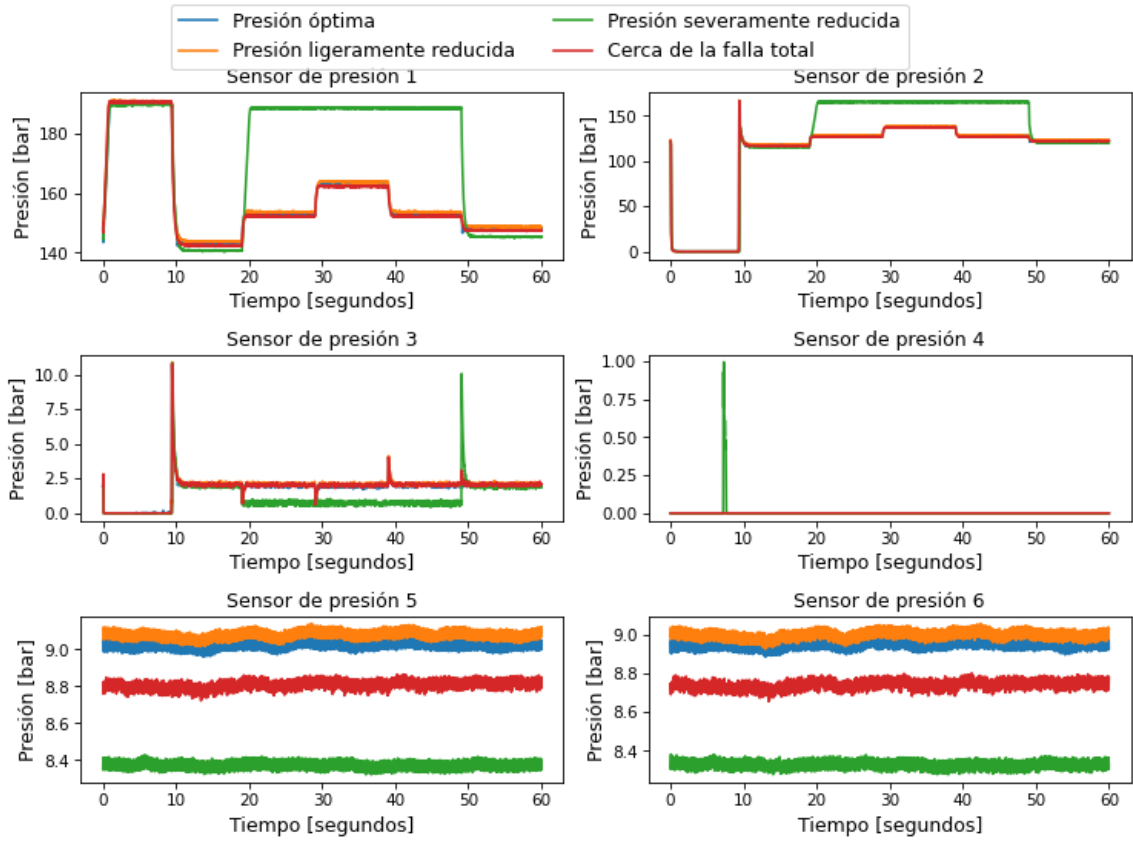


Figura A.69: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 4.

Datos sin procesar de estado del acumulador

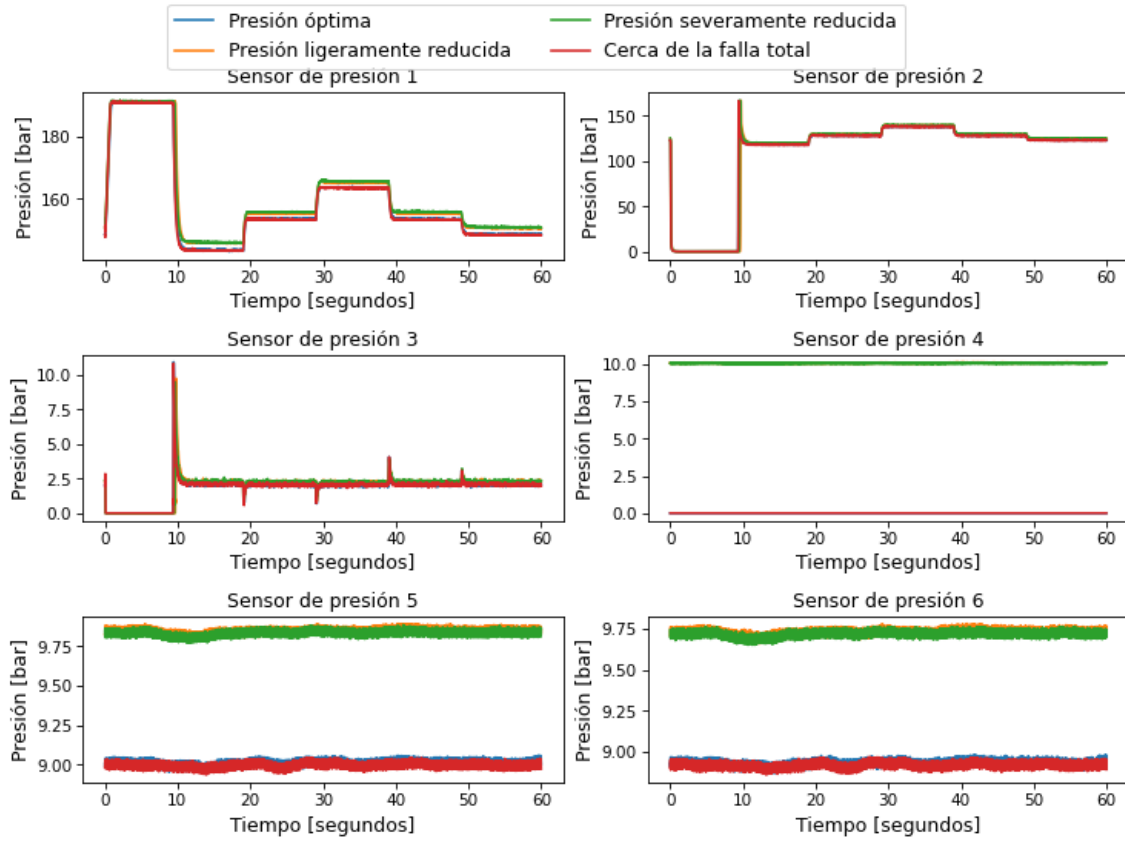


Figura A.70: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases del estado del acumulador, iteración 5.

A.1.3.5. Gráficos de la estabilidad del sistema

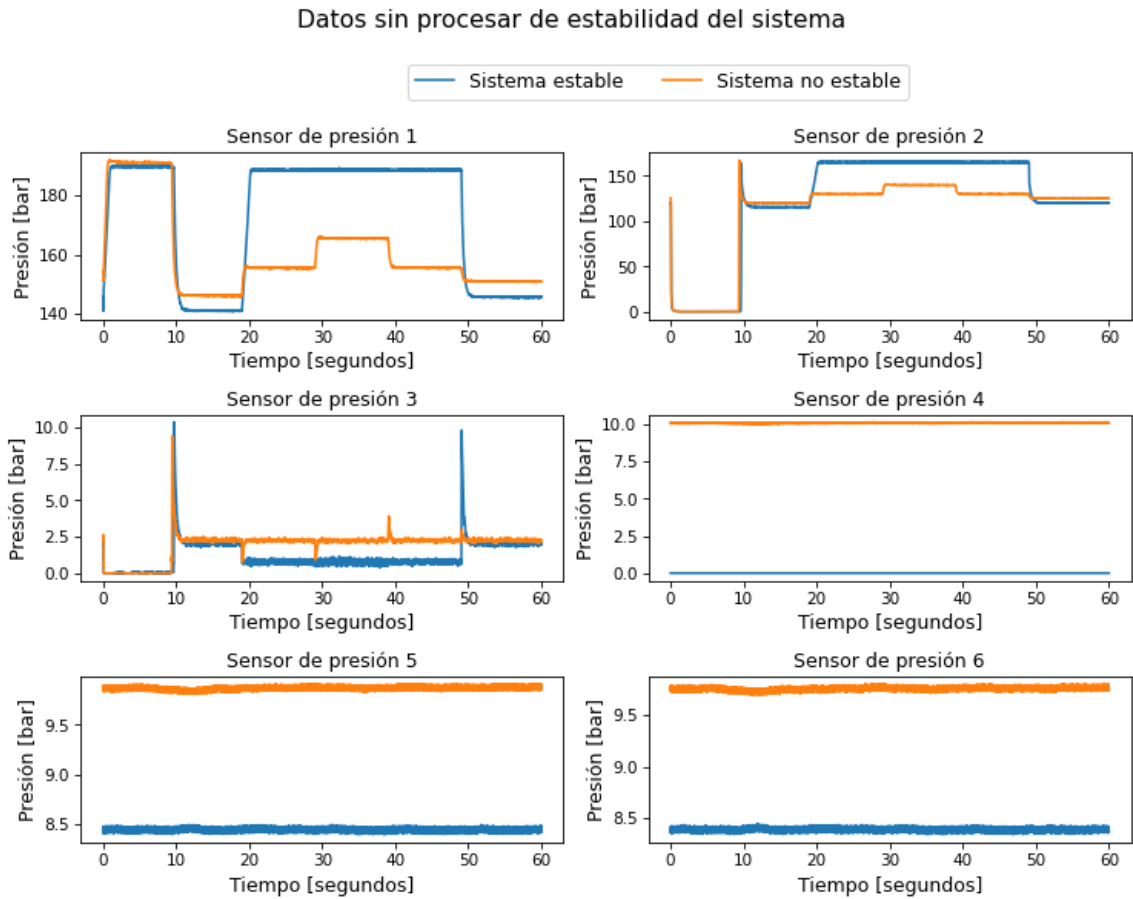


Figura A.71: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 1.

Datos sin procesar de estabilidad del sistema

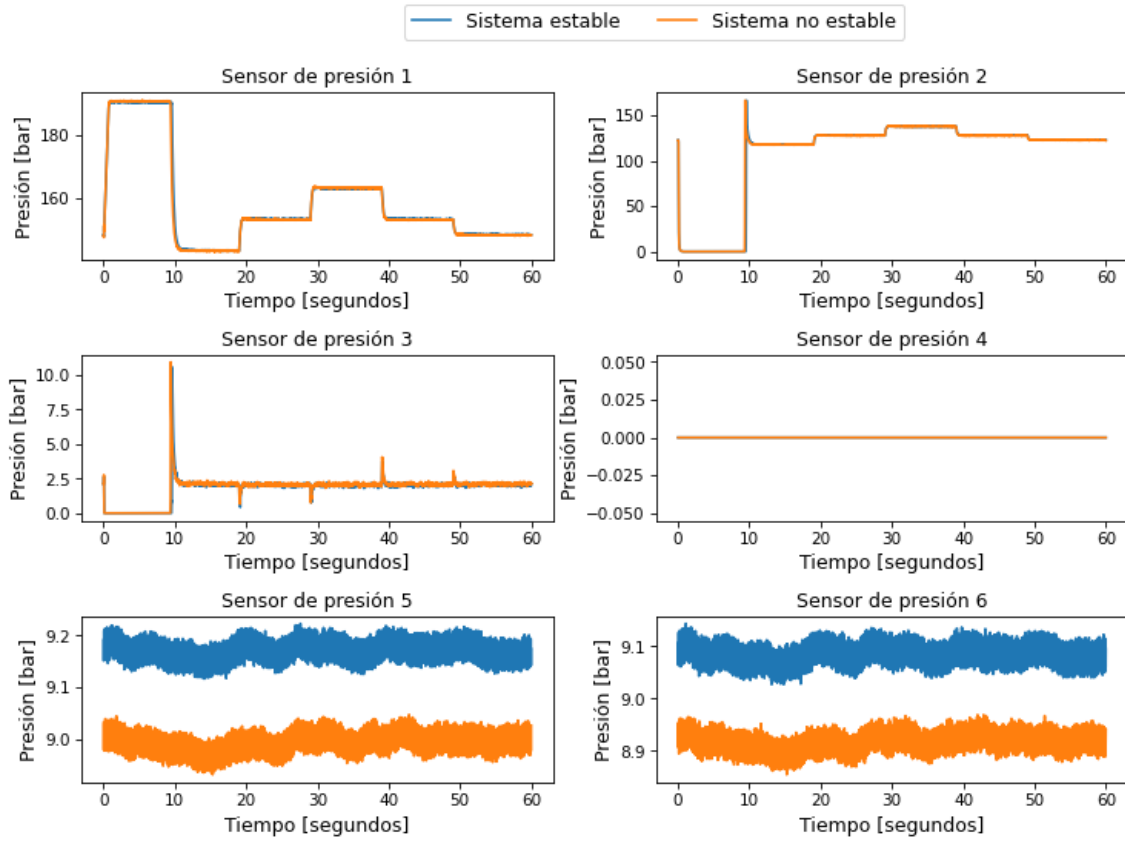


Figura A.72: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 2.

Datos sin procesar de estabilidad del sistema

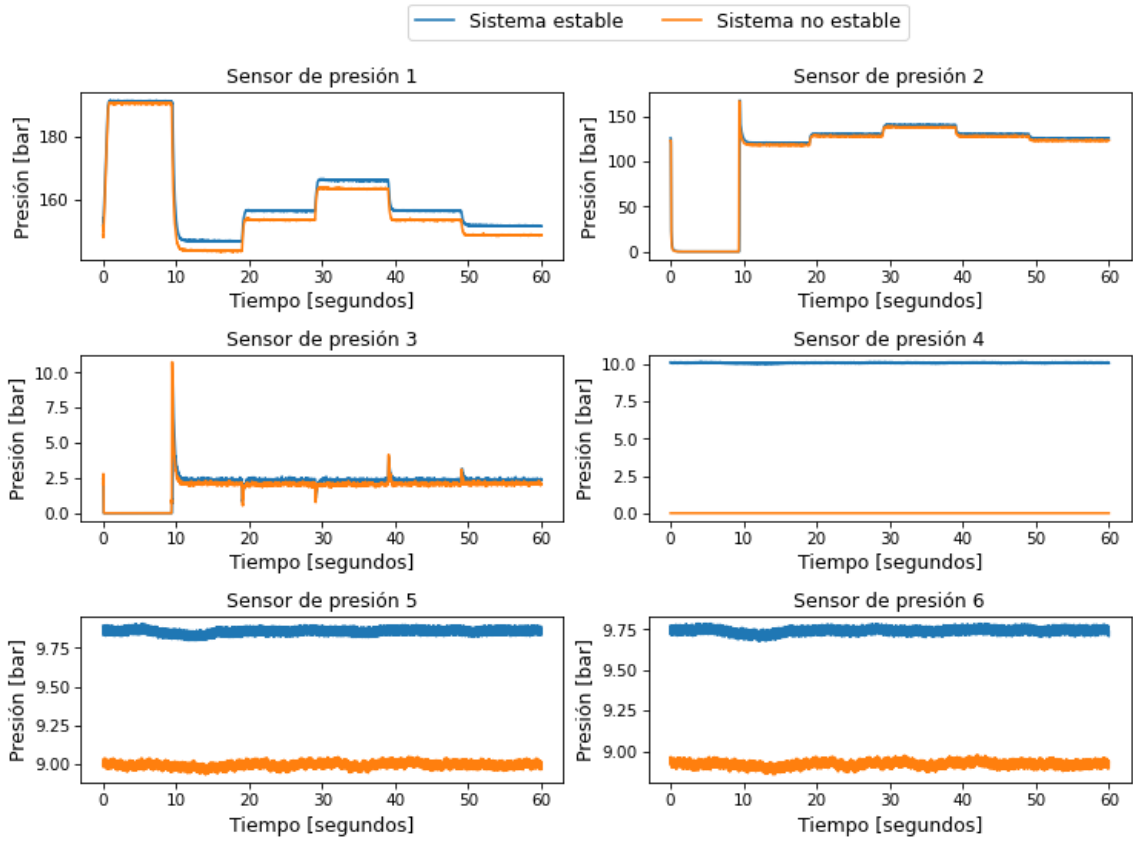


Figura A.73: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 3.

Datos sin procesar de estabilidad del sistema

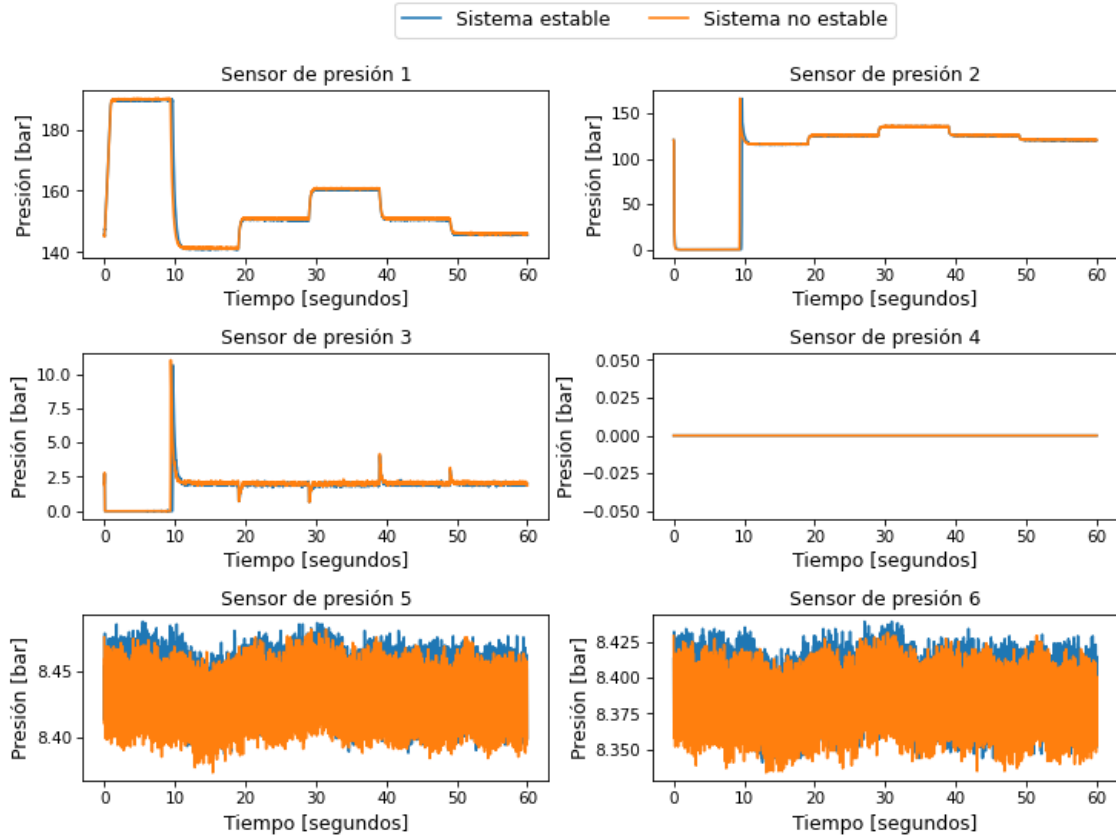


Figura A.74: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 4.

Datos sin procesar de estabilidad del sistema

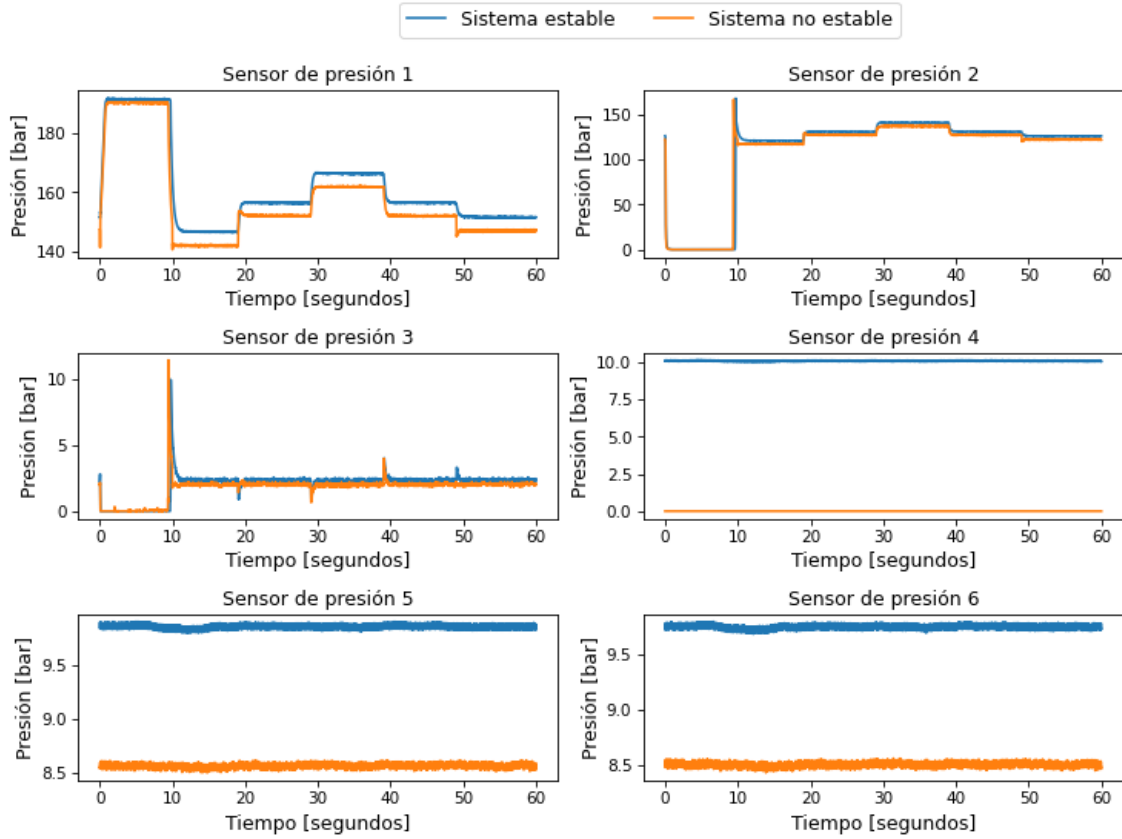


Figura A.75: Gráficos de los datos sin procesar obtenidos por los sensores de presión, separando según las clases de la estabilidad del sistema, iteración 5.

A.1.4. Gráficos de otras variables

A.1.4.1. Gráficos del enfriador

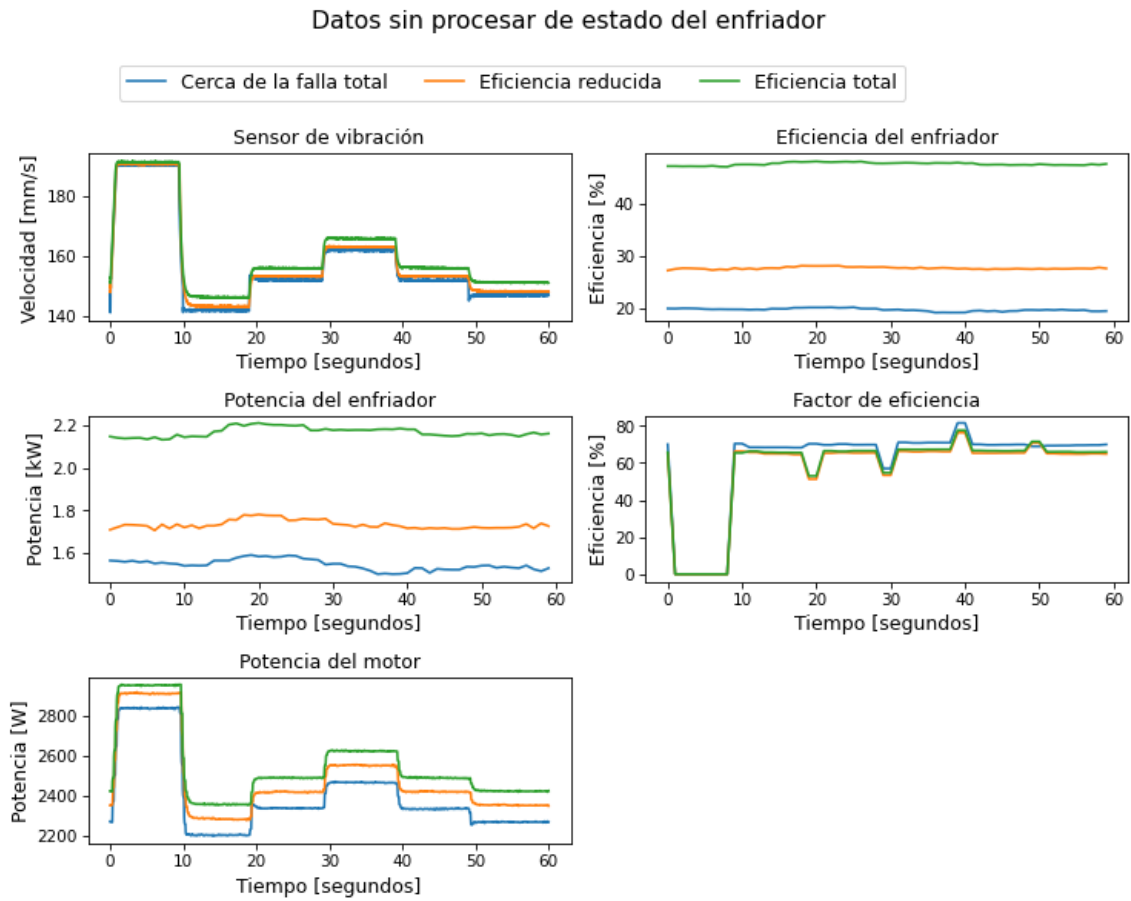


Figura A.76: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 1.

Datos sin procesar de estado del enfriador

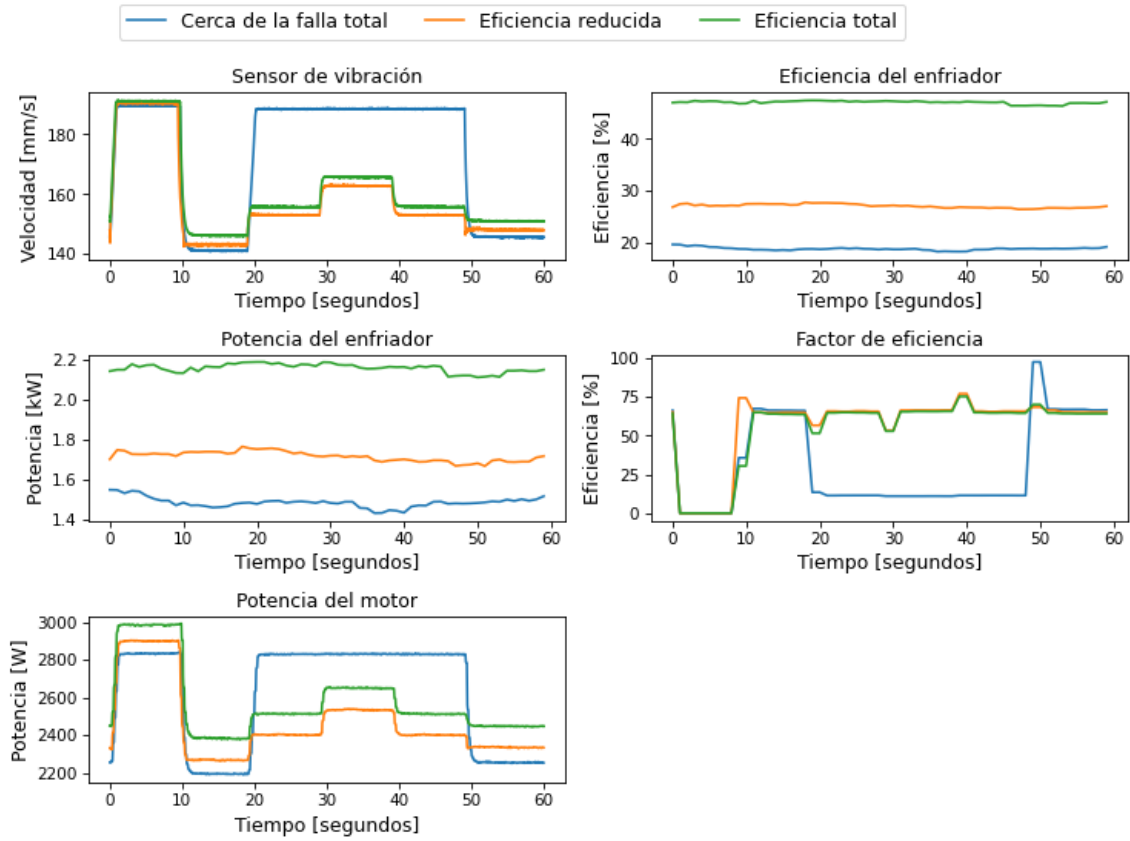


Figura A.77: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 2.

Datos sin procesar de estado del enfriador

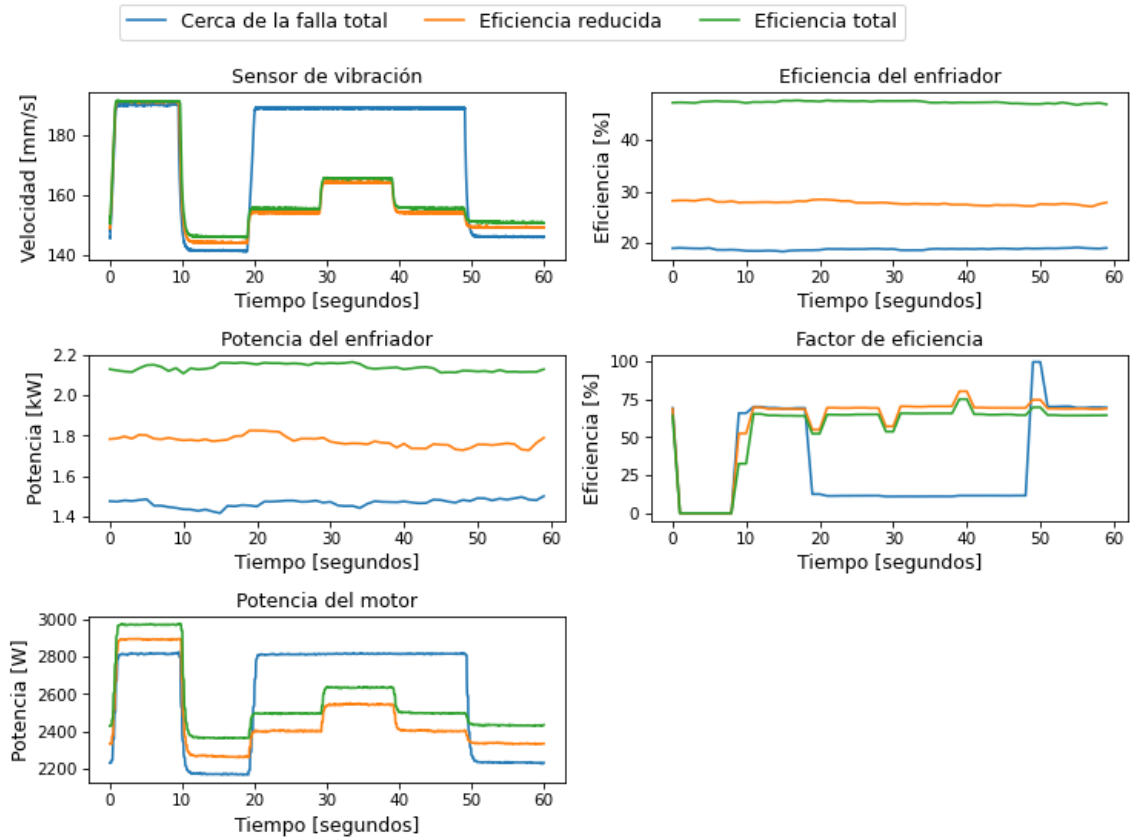


Figura A.78: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 3.

Datos sin procesar de estado del enfriador

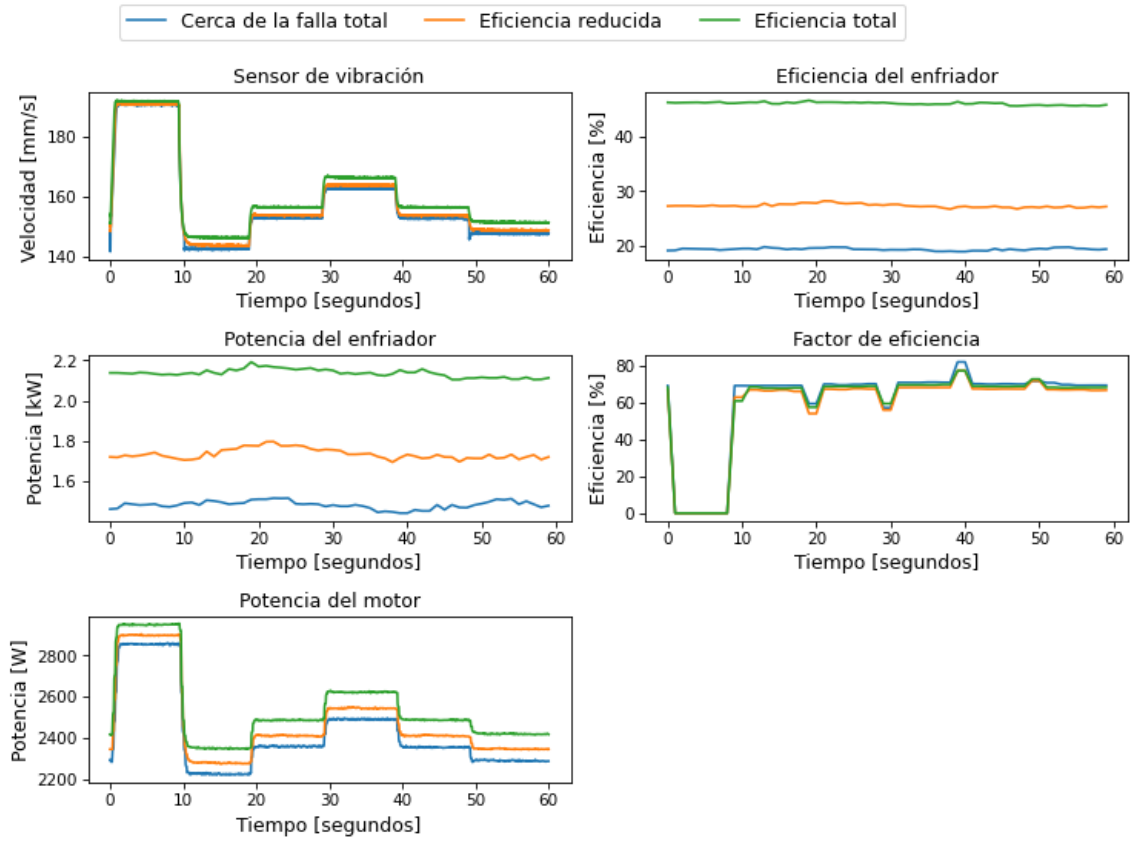


Figura A.79: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 4.

Datos sin procesar de estado del enfriador

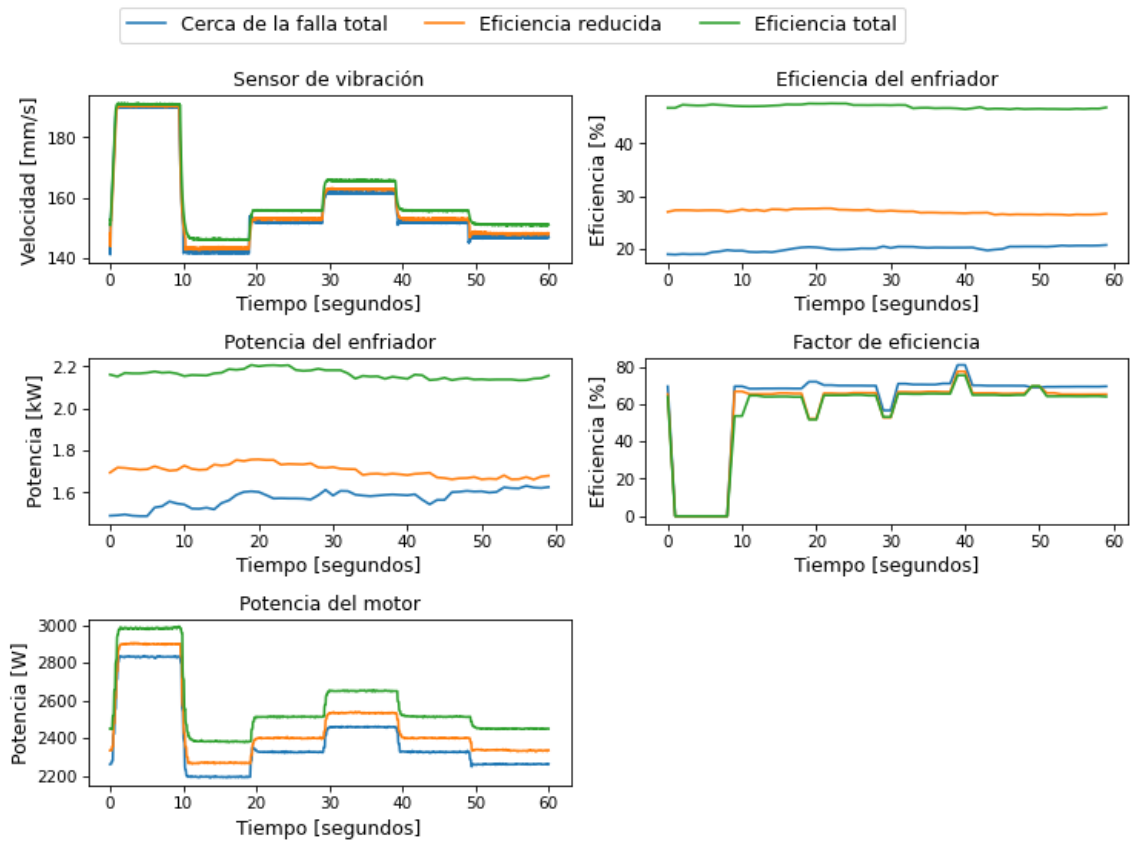


Figura A.80: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del enfriador, iteración 5.

A.1.4.2. Gráficos de la válvula

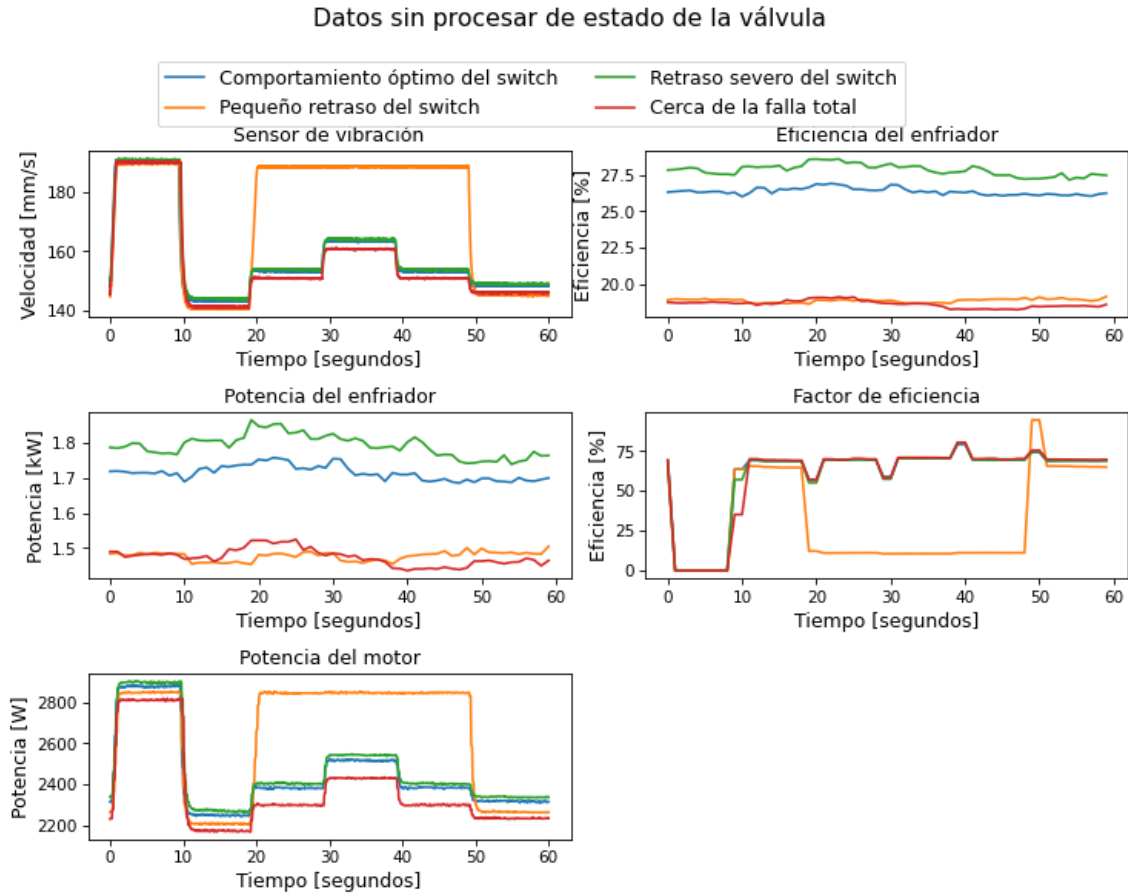


Figura A.81: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 1.

Datos sin procesar de estado de la válvula

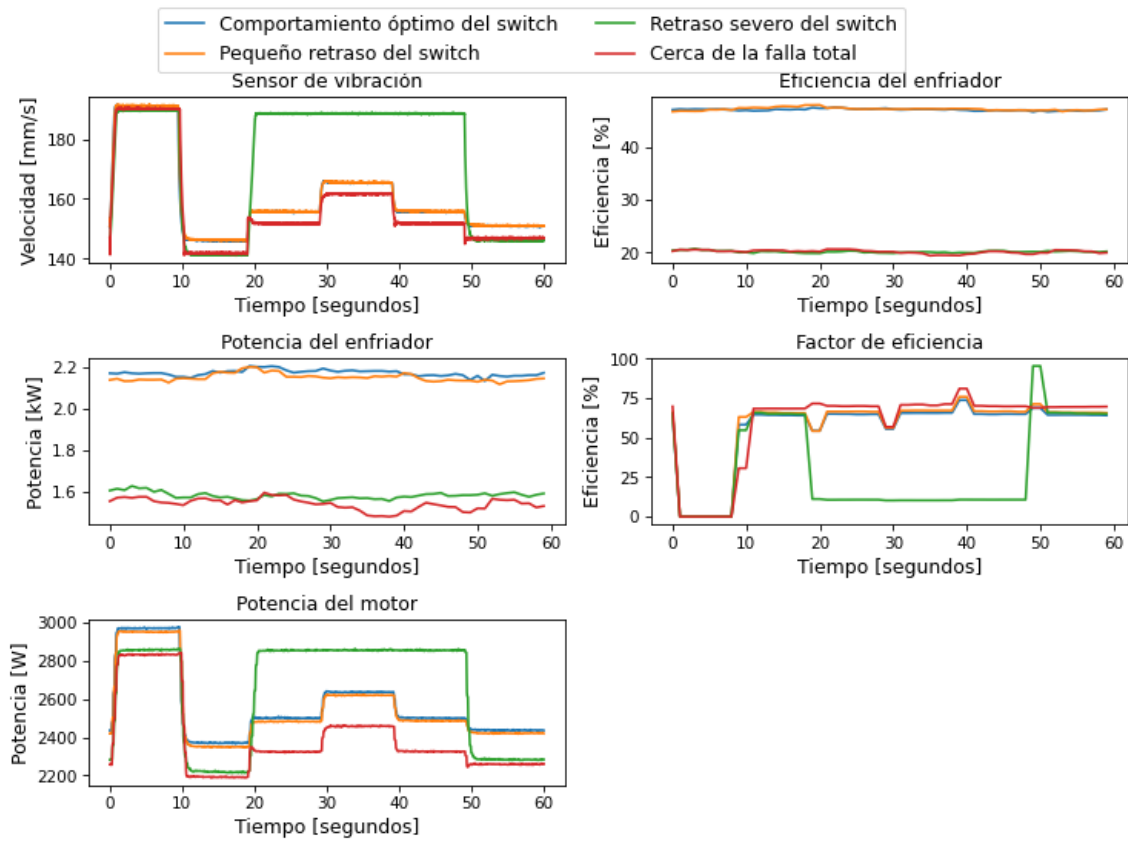


Figura A.82: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 2.

Datos sin procesar de estado de la válvula

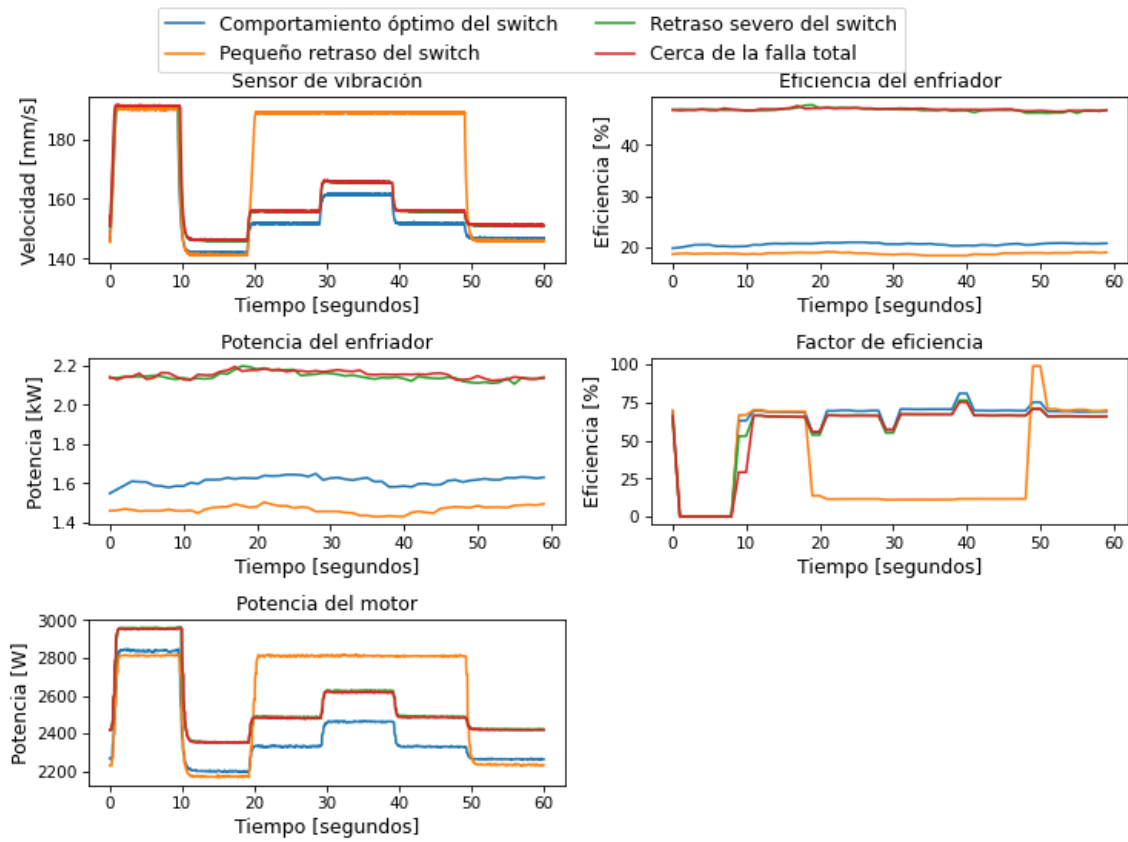


Figura A.83: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 3.

Datos sin procesar de estado de la válvula

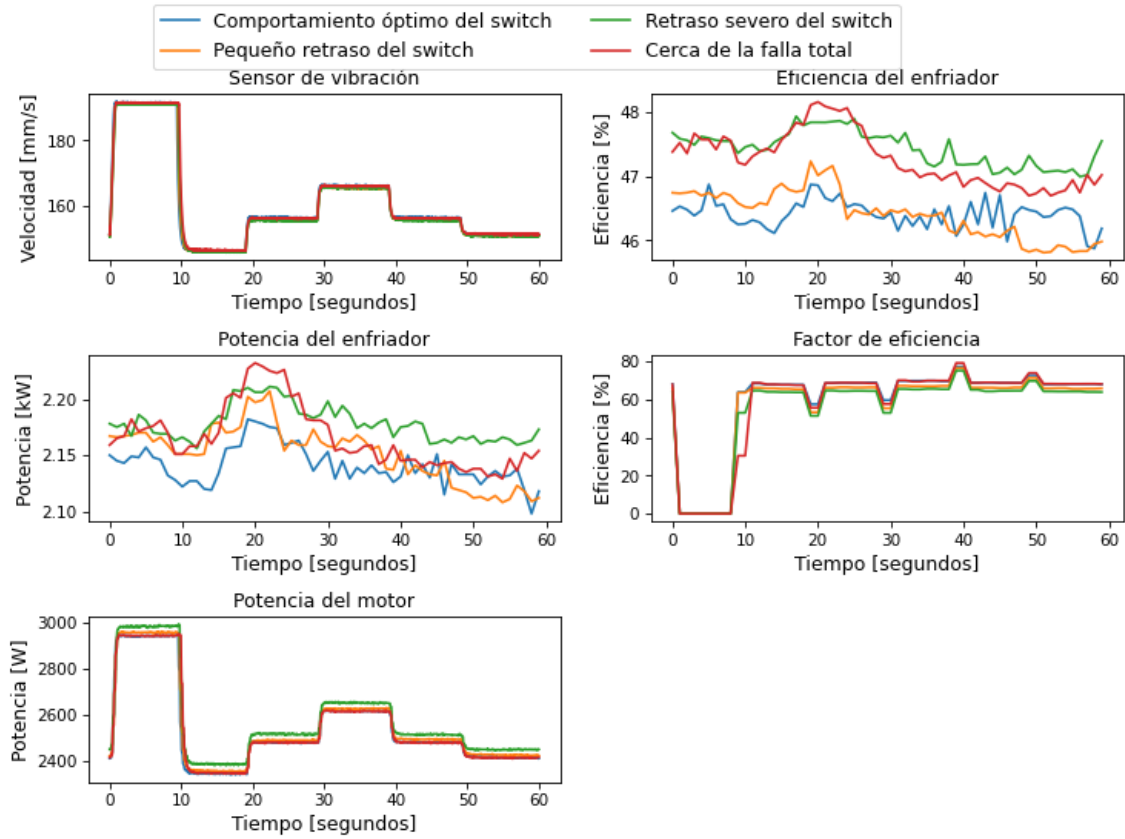


Figura A.84: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 4.

Datos sin procesar de estado de la válvula

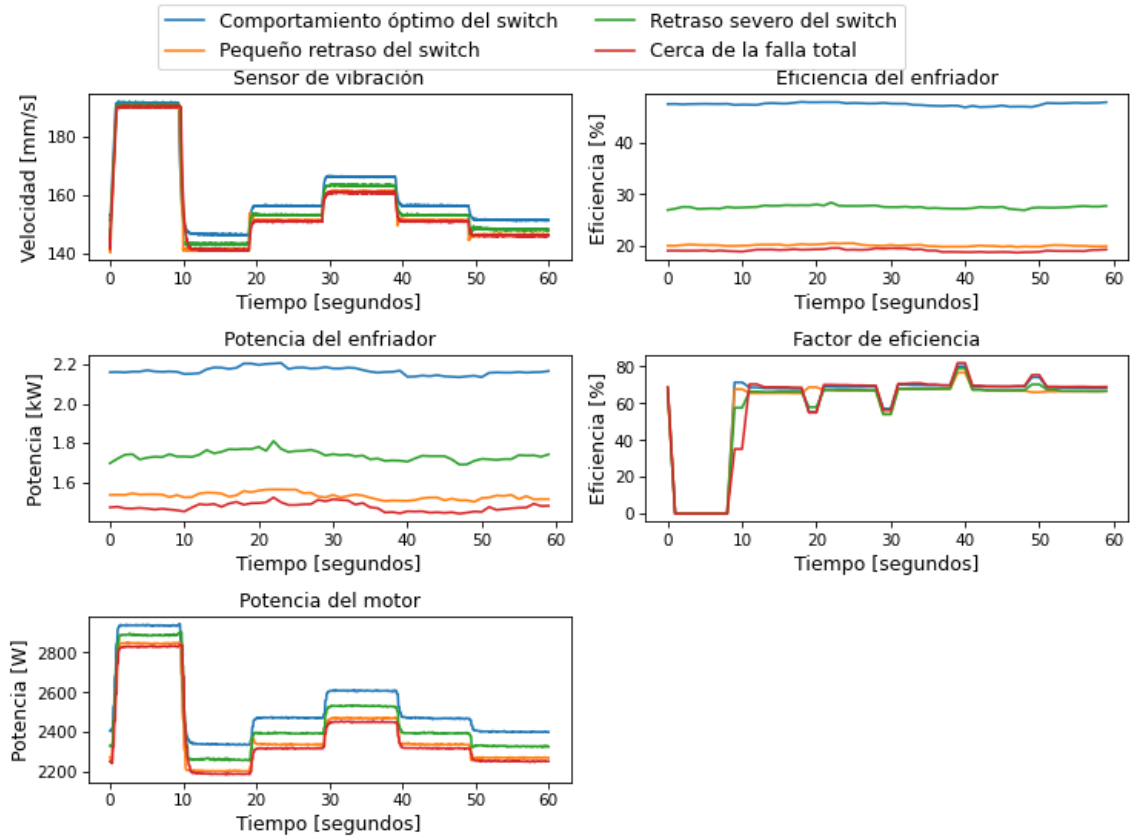


Figura A.85: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado de la válvula, iteración 5.

A.1.4.3. Gráficos de la bomba

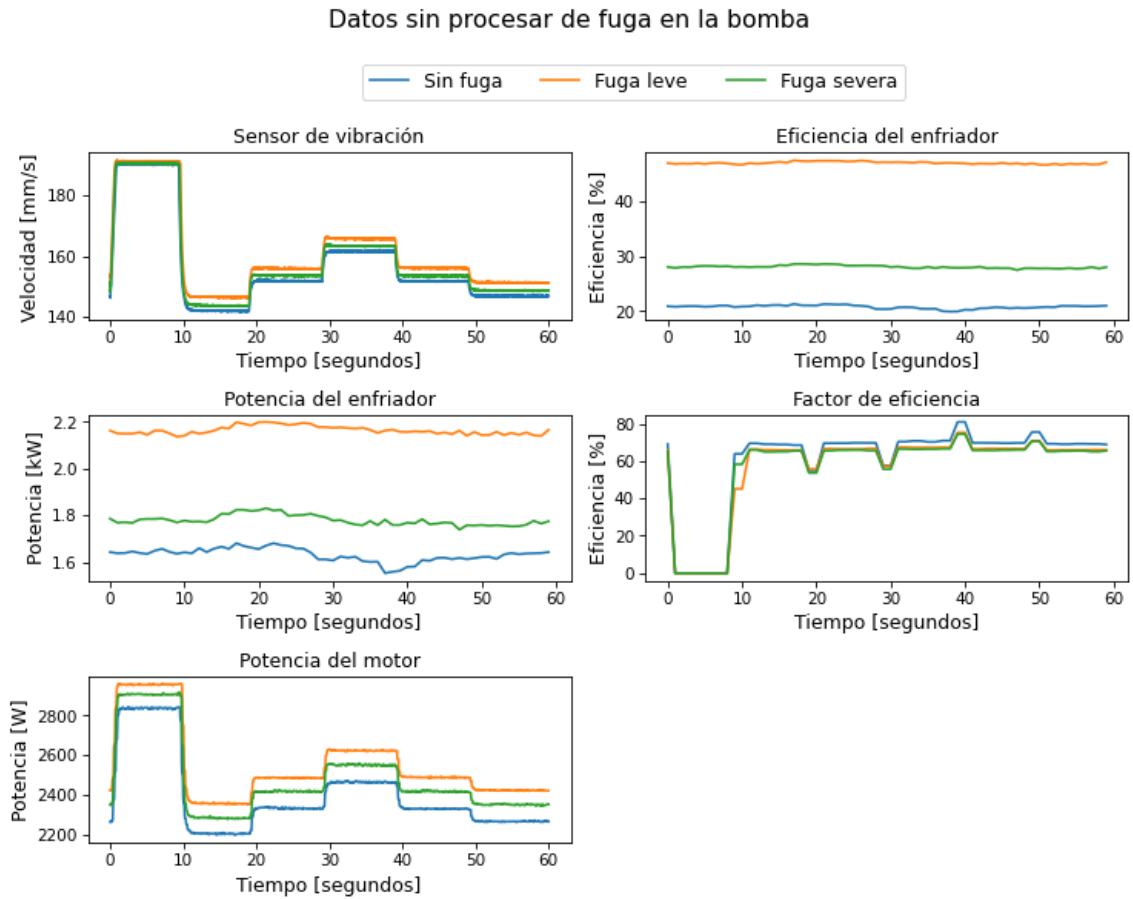


Figura A.86: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 1.

Datos sin procesar de fuga en la bomba

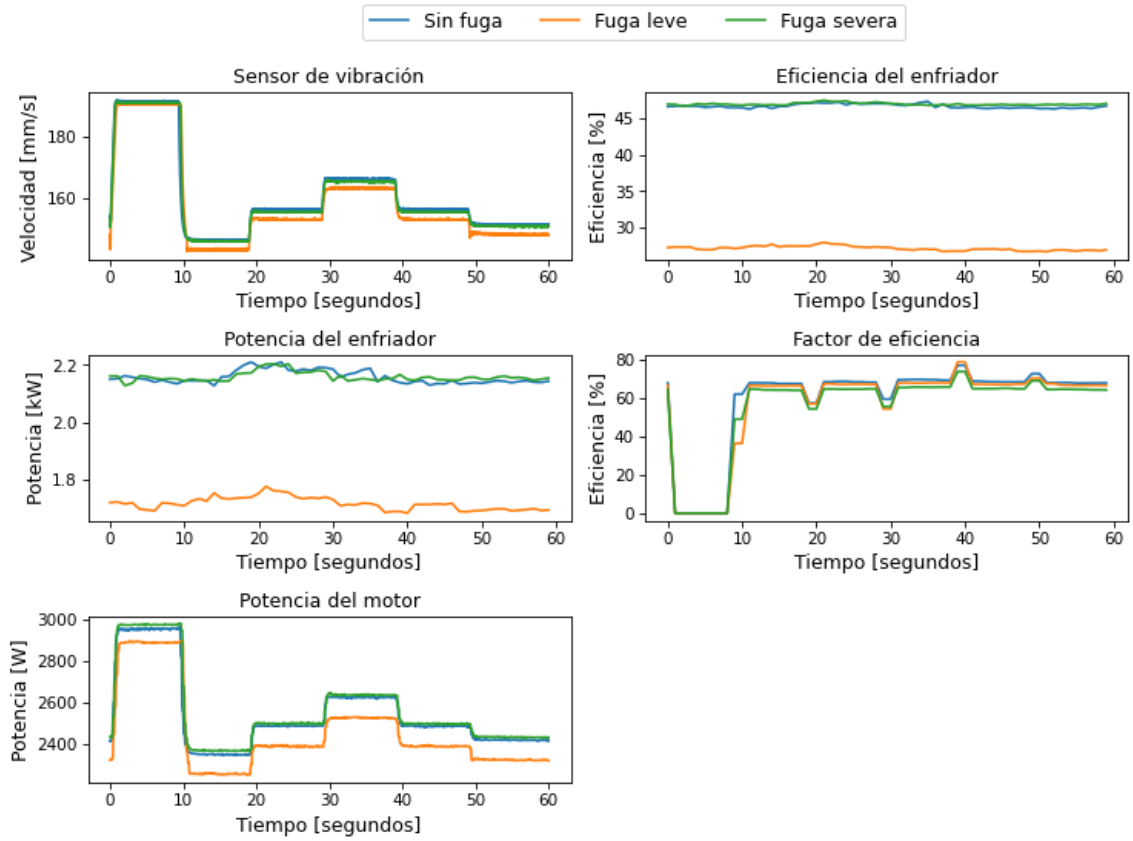


Figura A.87: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 2.

Datos sin procesar de fuga en la bomba

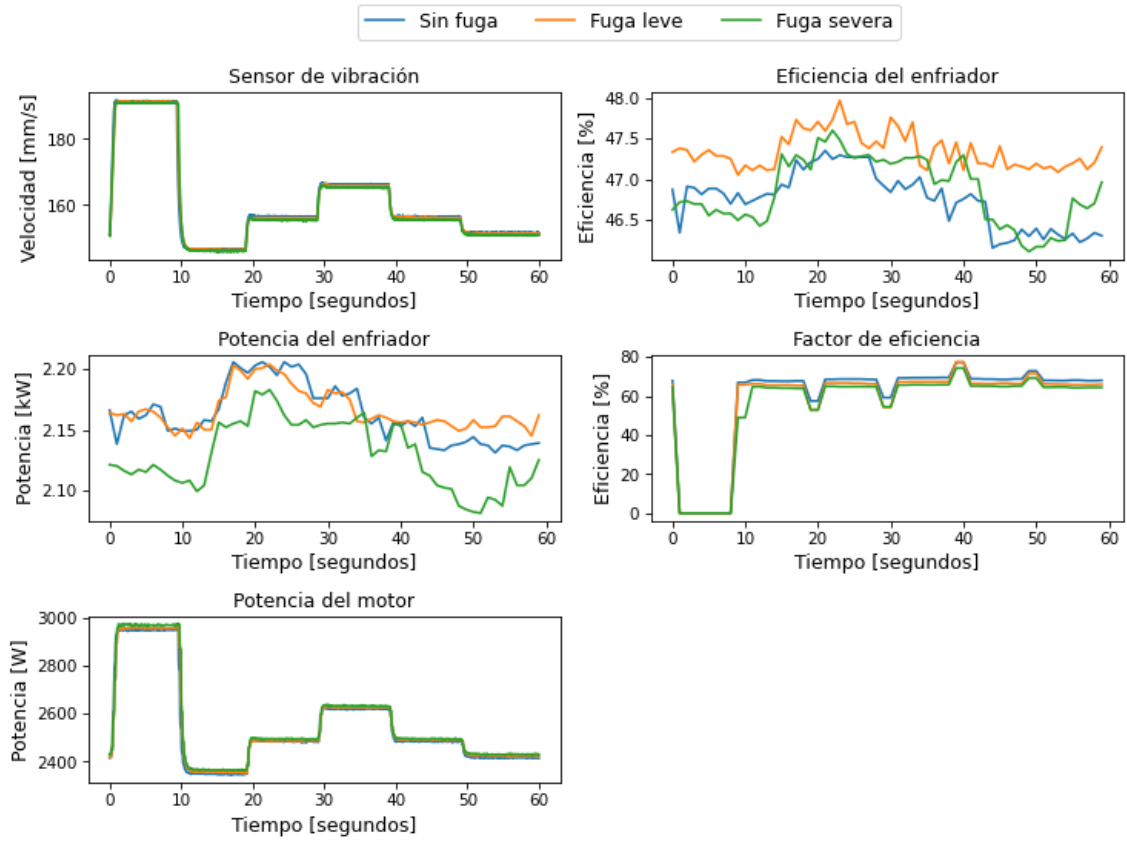


Figura A.88: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 3.

Datos sin procesar de fuga en la bomba

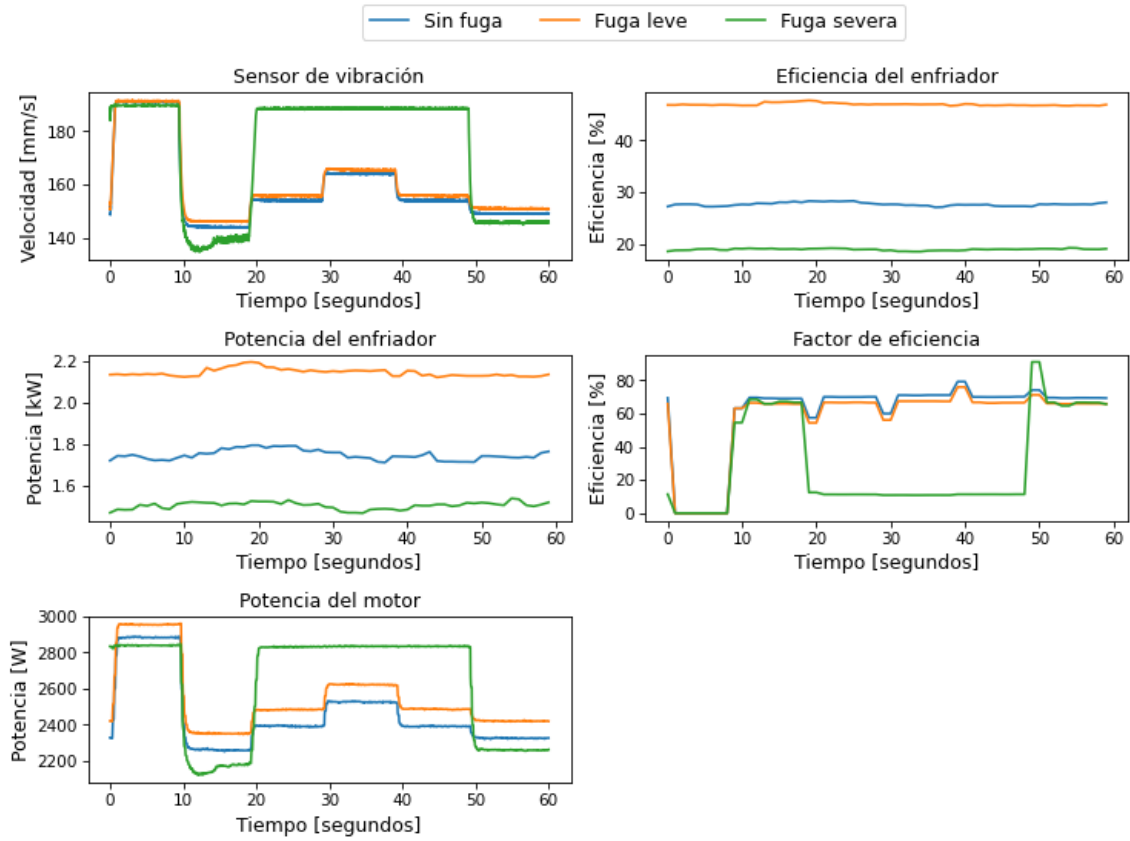


Figura A.89: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 4.

Datos sin procesar de fuga en la bomba

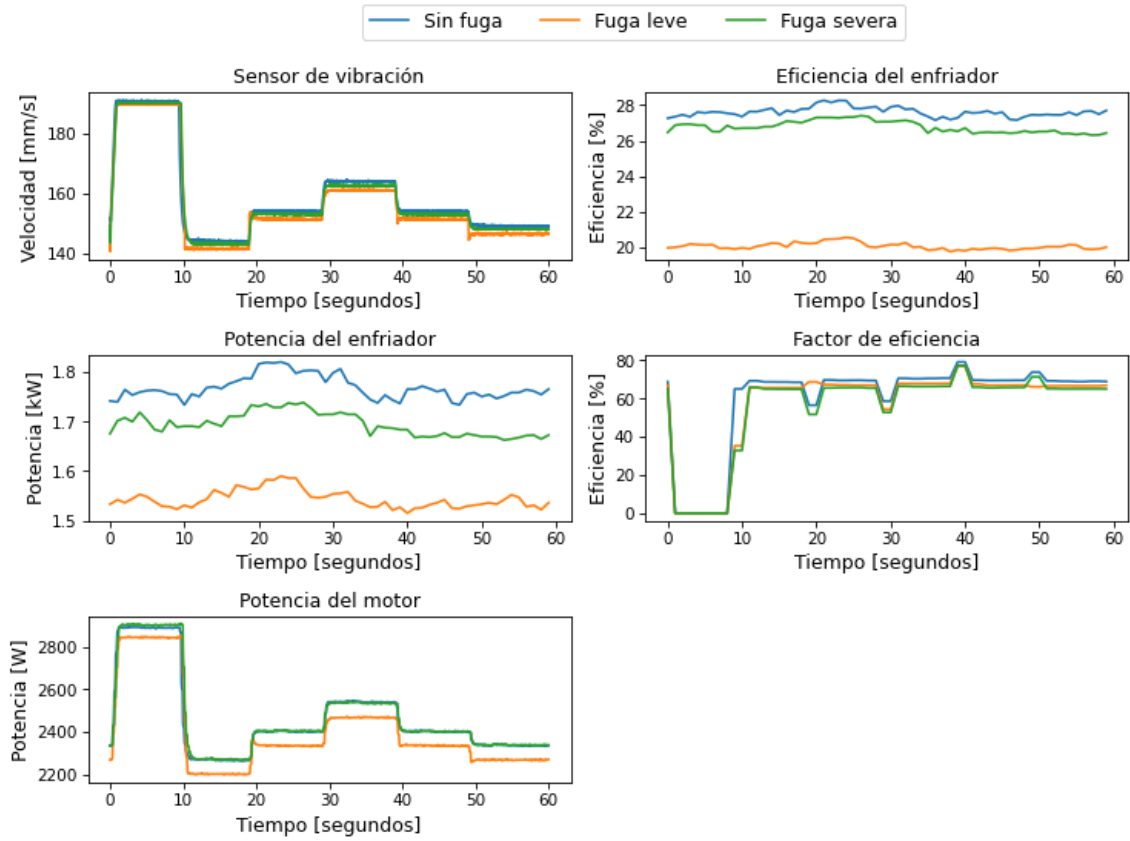


Figura A.90: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la fuga en la bomba, iteración 5.

A.1.4.4. Gráficos del acumulador

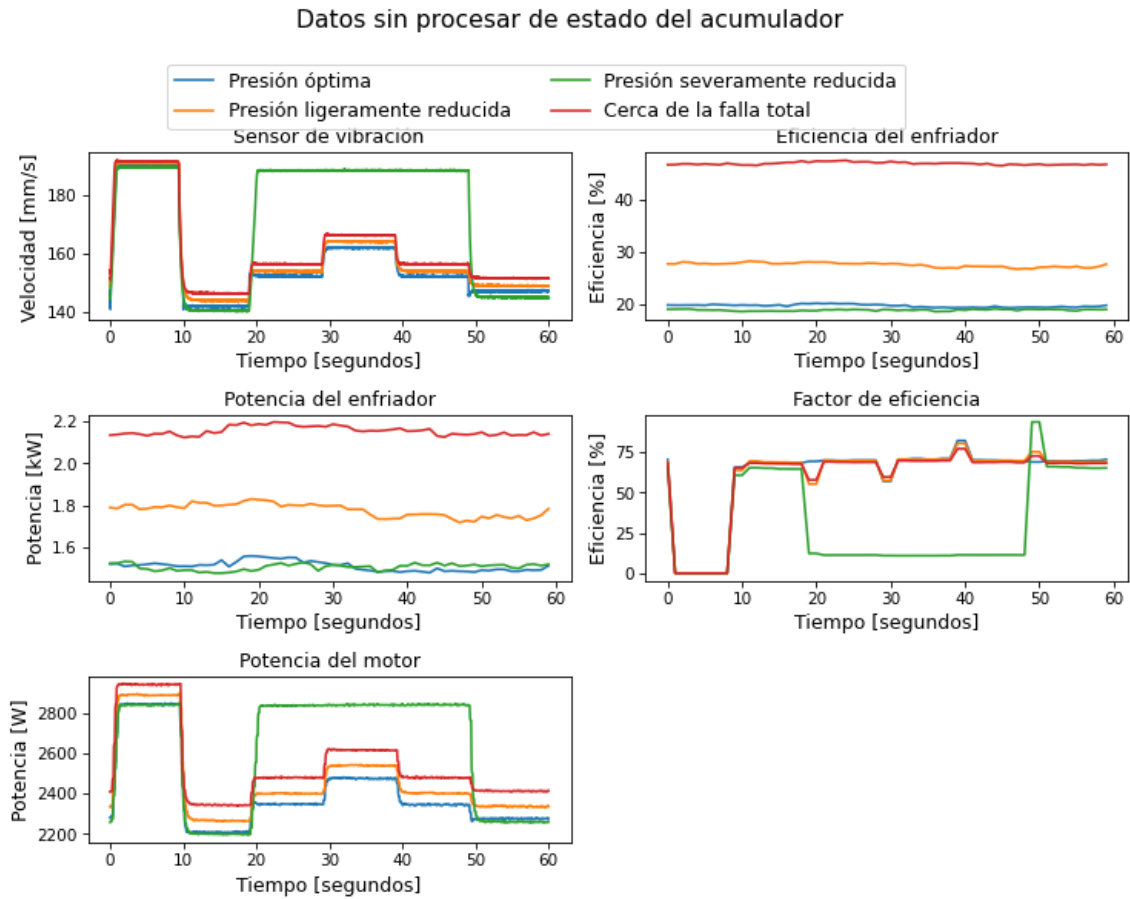


Figura A.91: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 1.

Datos sin procesar de estado del acumulador

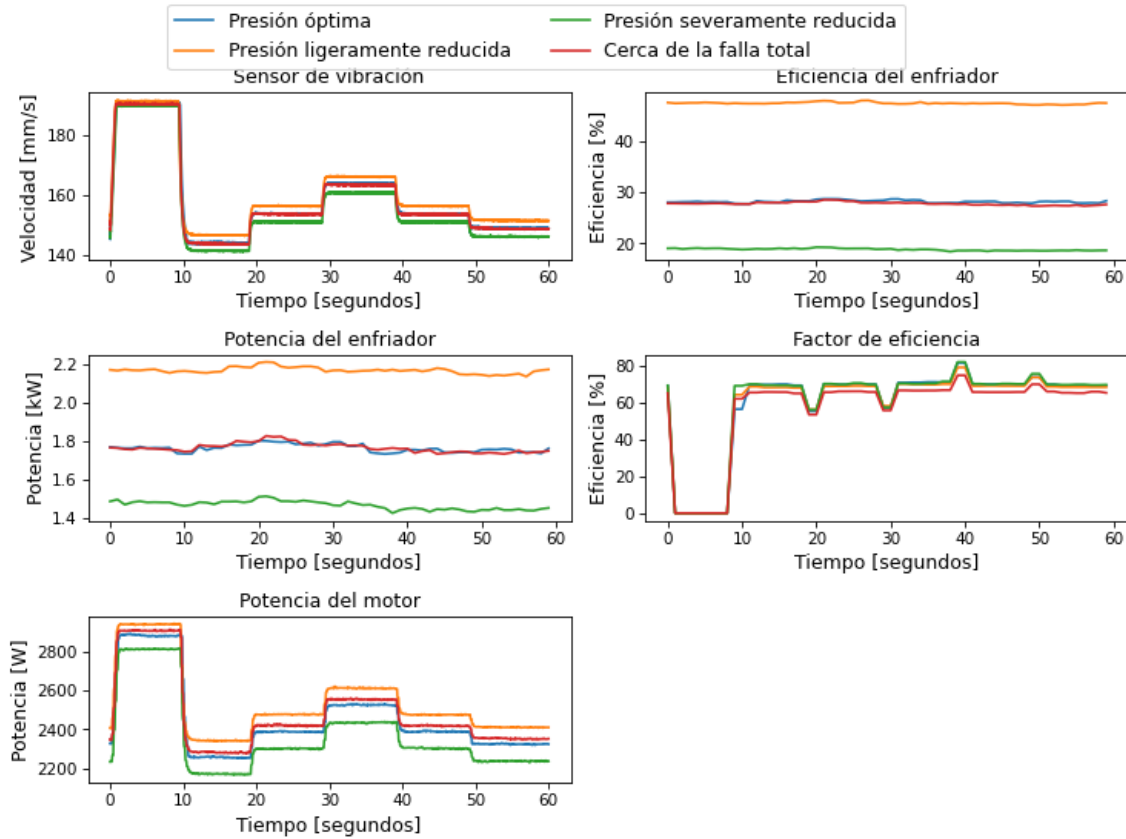


Figura A.92: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 2.

Datos sin procesar de estado del acumulador

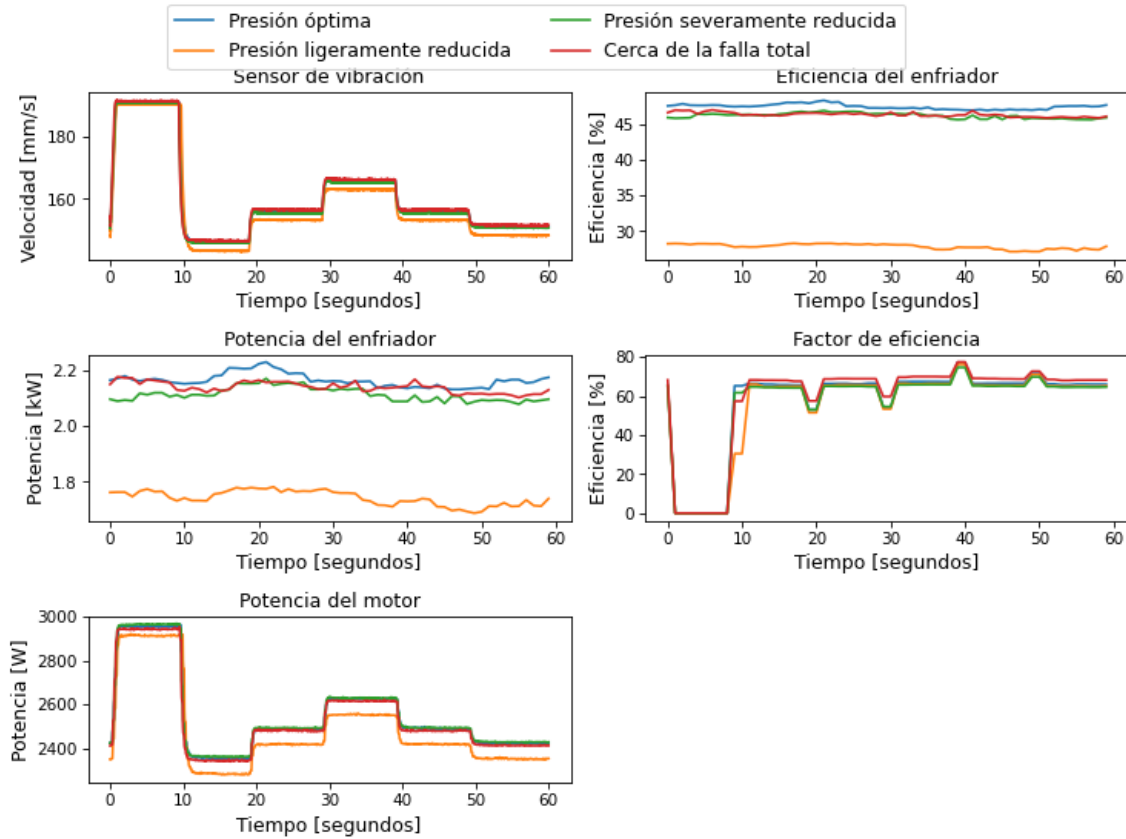


Figura A.93: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 3.

Datos sin procesar de estado del acumulador

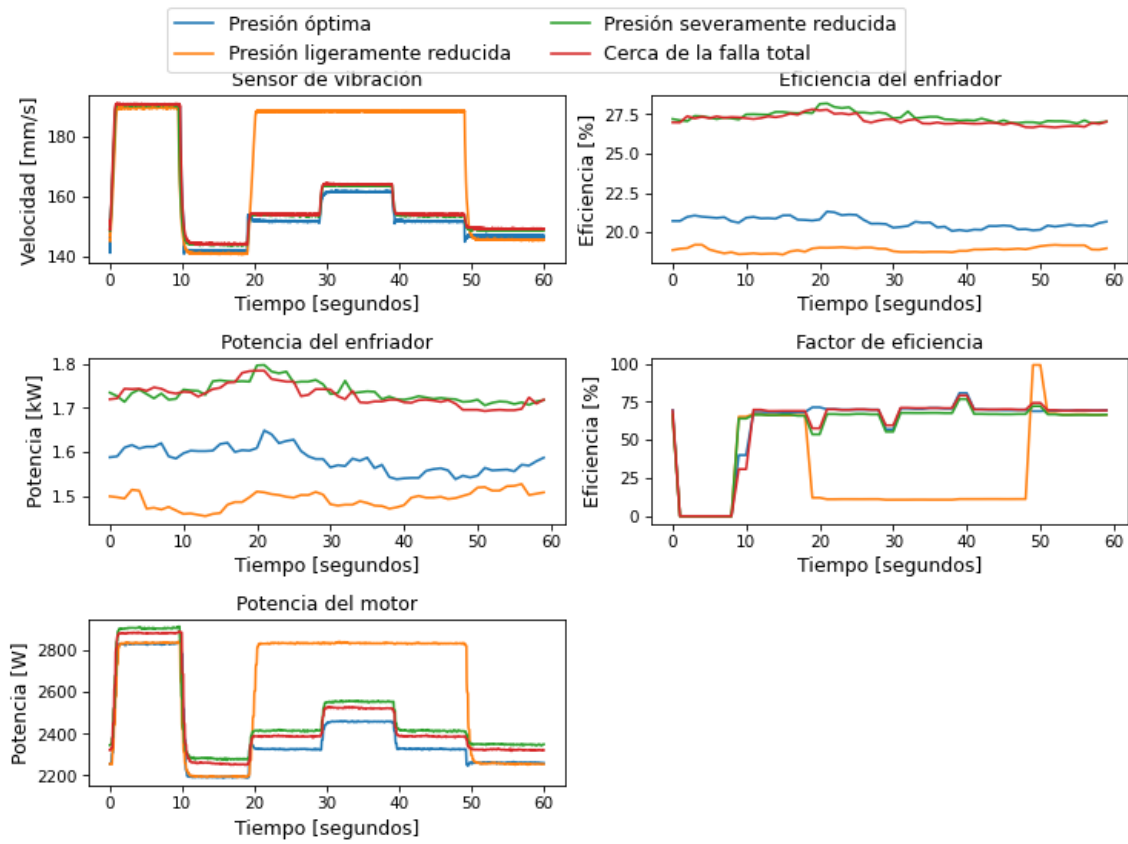


Figura A.94: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 4.

Datos sin procesar de estado del acumulador

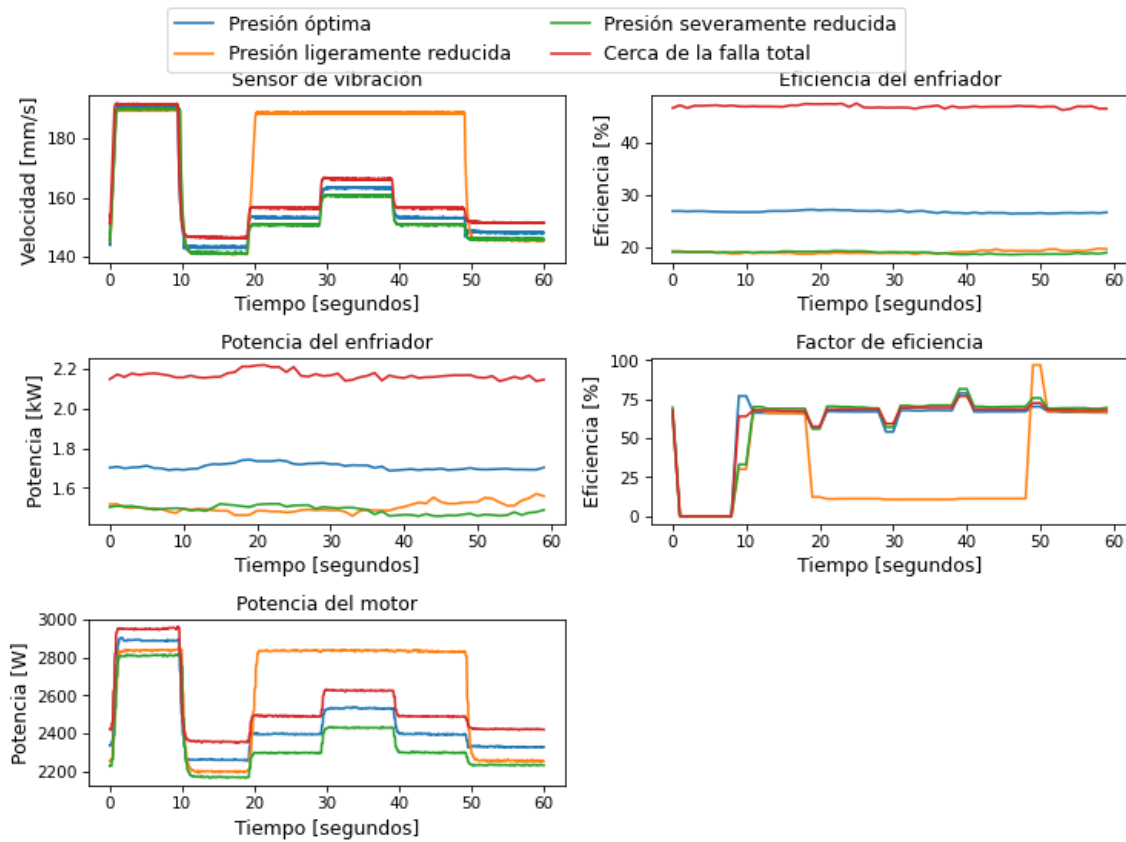


Figura A.95: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases del estado del acumulador, iteración 5.

A.1.4.5. Gráficos de la estabilidad del sistema

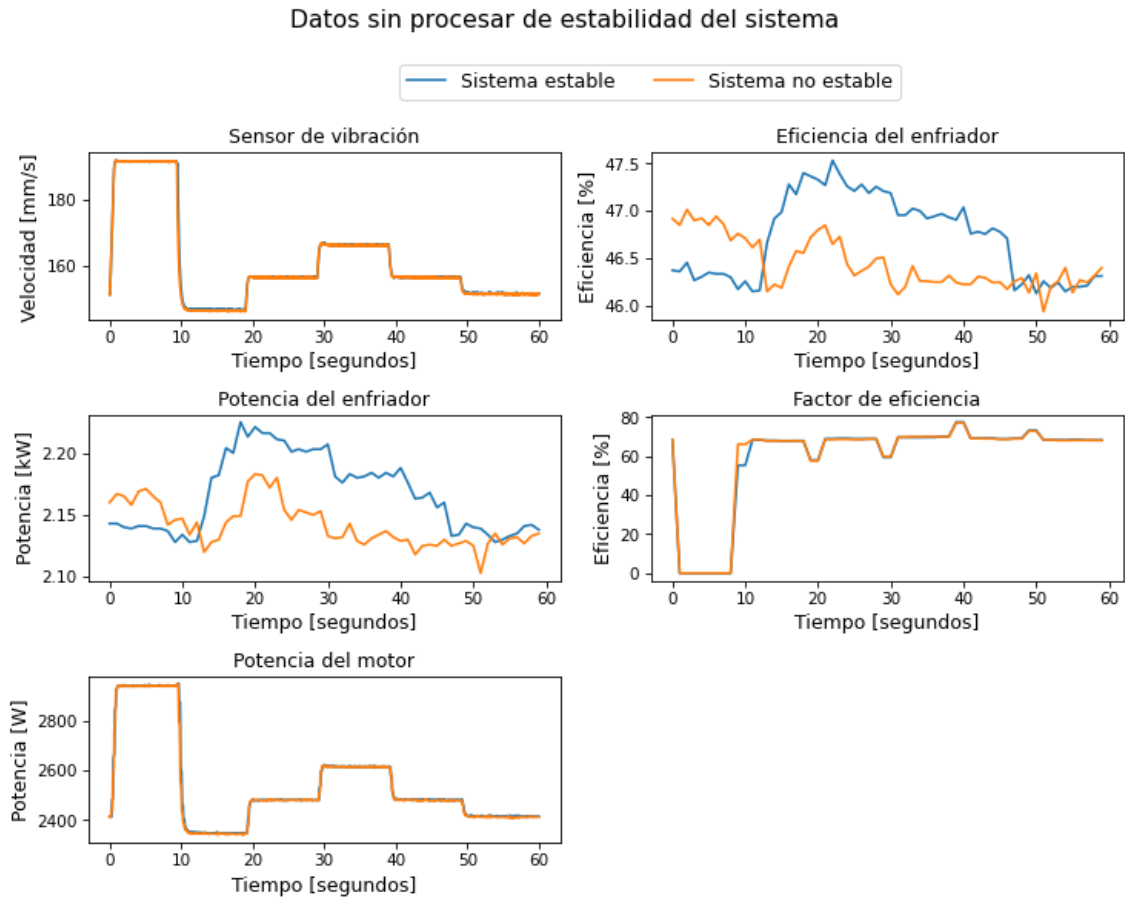


Figura A.96: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 1.

Datos sin procesar de estabilidad del sistema

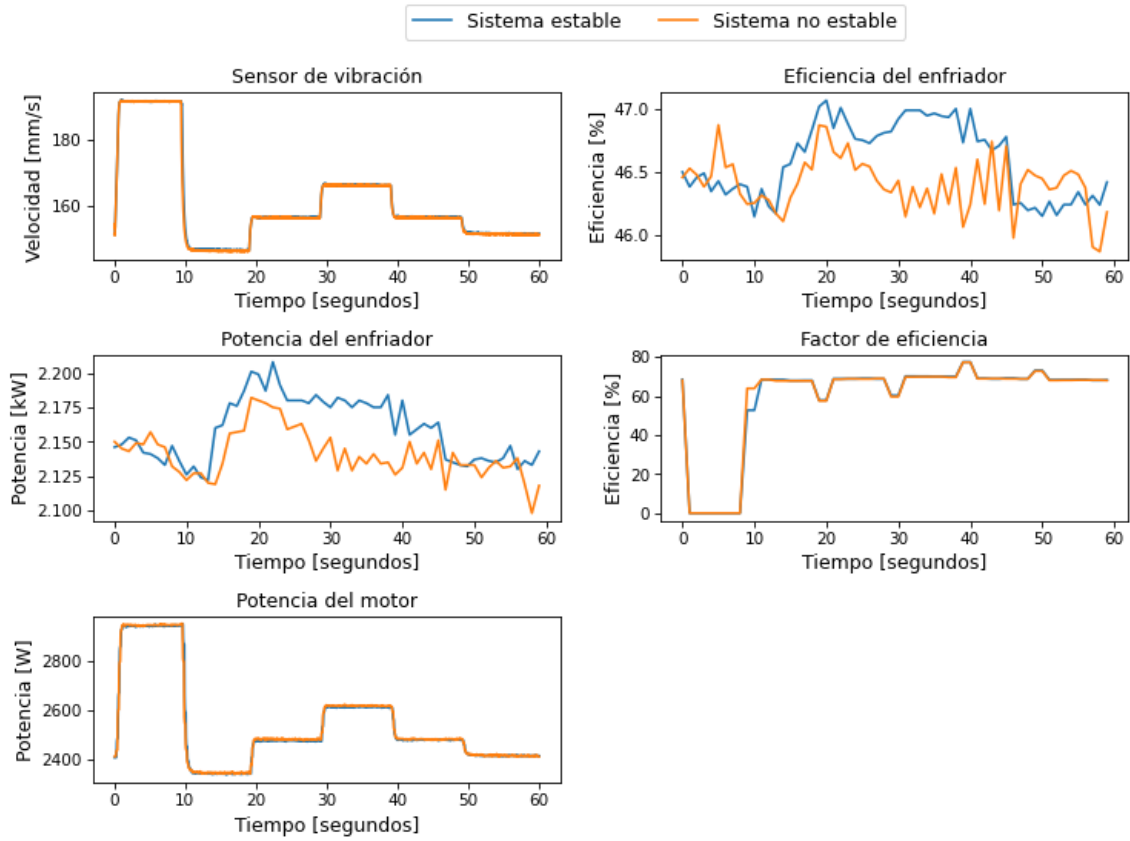


Figura A.97: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 2.

Datos sin procesar de estabilidad del sistema

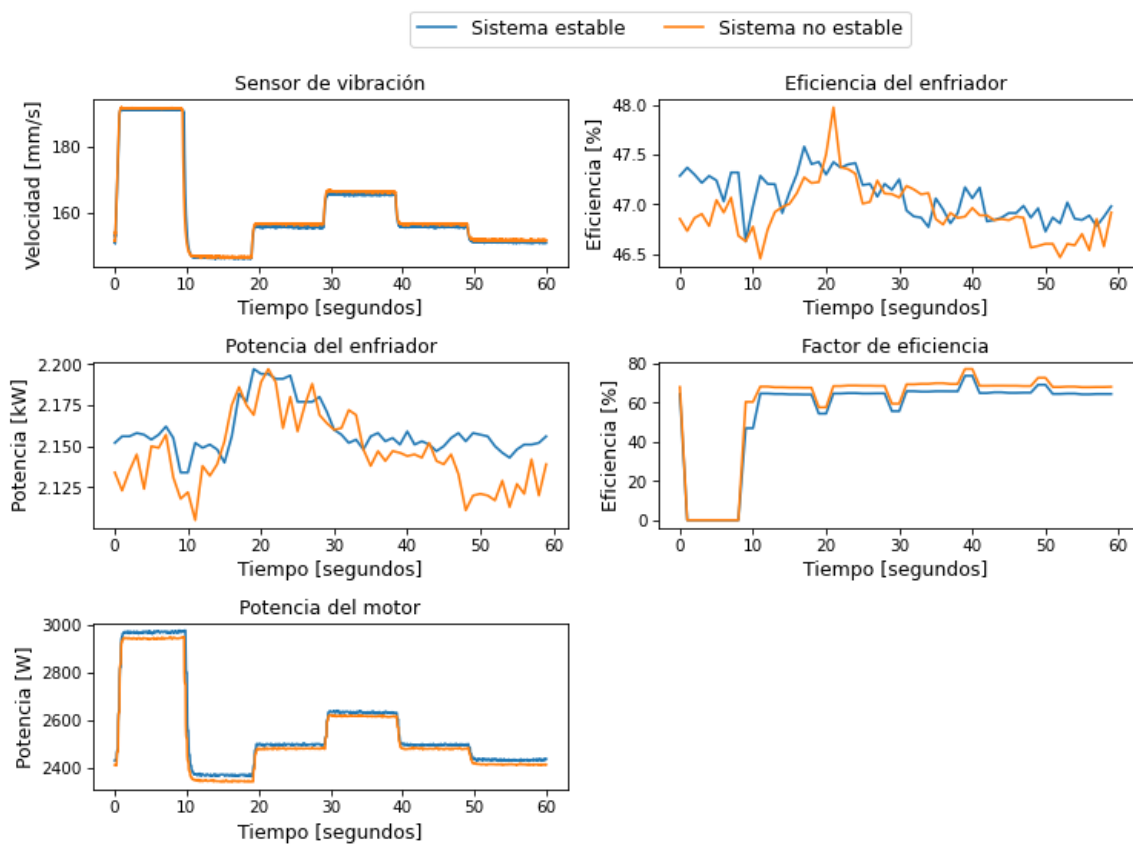


Figura A.98: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 3.

Datos sin procesar de estabilidad del sistema

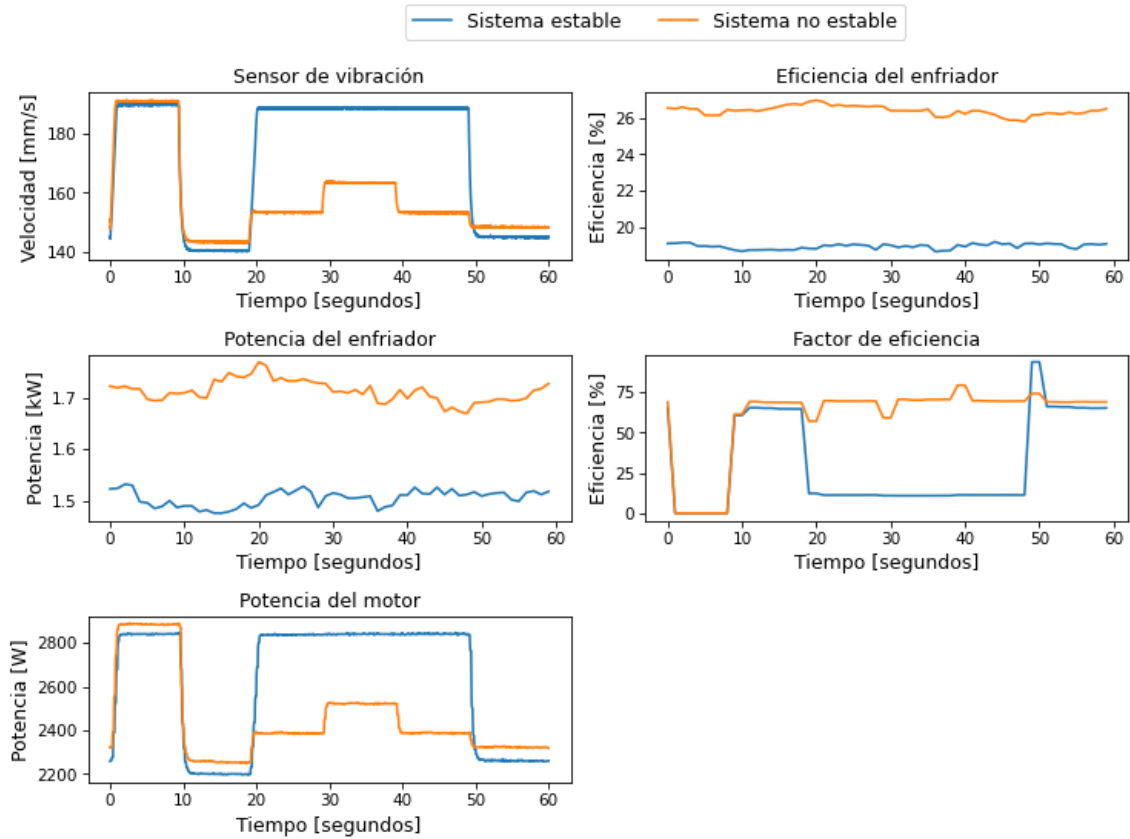


Figura A.99: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 4.

Datos sin procesar de estabilidad del sistema

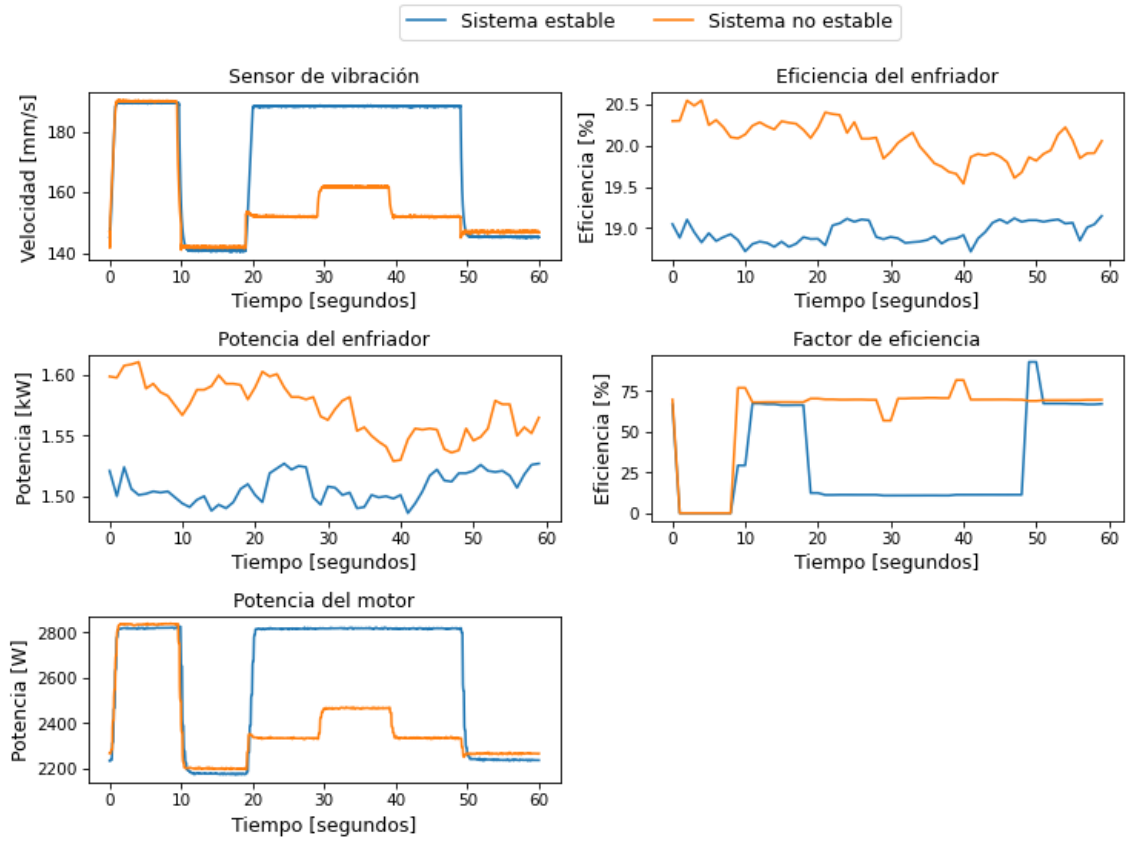


Figura A.100: Gráficos de los datos sin procesar obtenidos por los sensores de diferentes variables de operación, separando según las clases de la estabilidad del sistema, iteración 5.

A.2. Gráficos de los parámetros de tiempo

A.2.1. Gráficos del enfriador

A.2.1.1. Gráficos usando 1 dato por ciclo

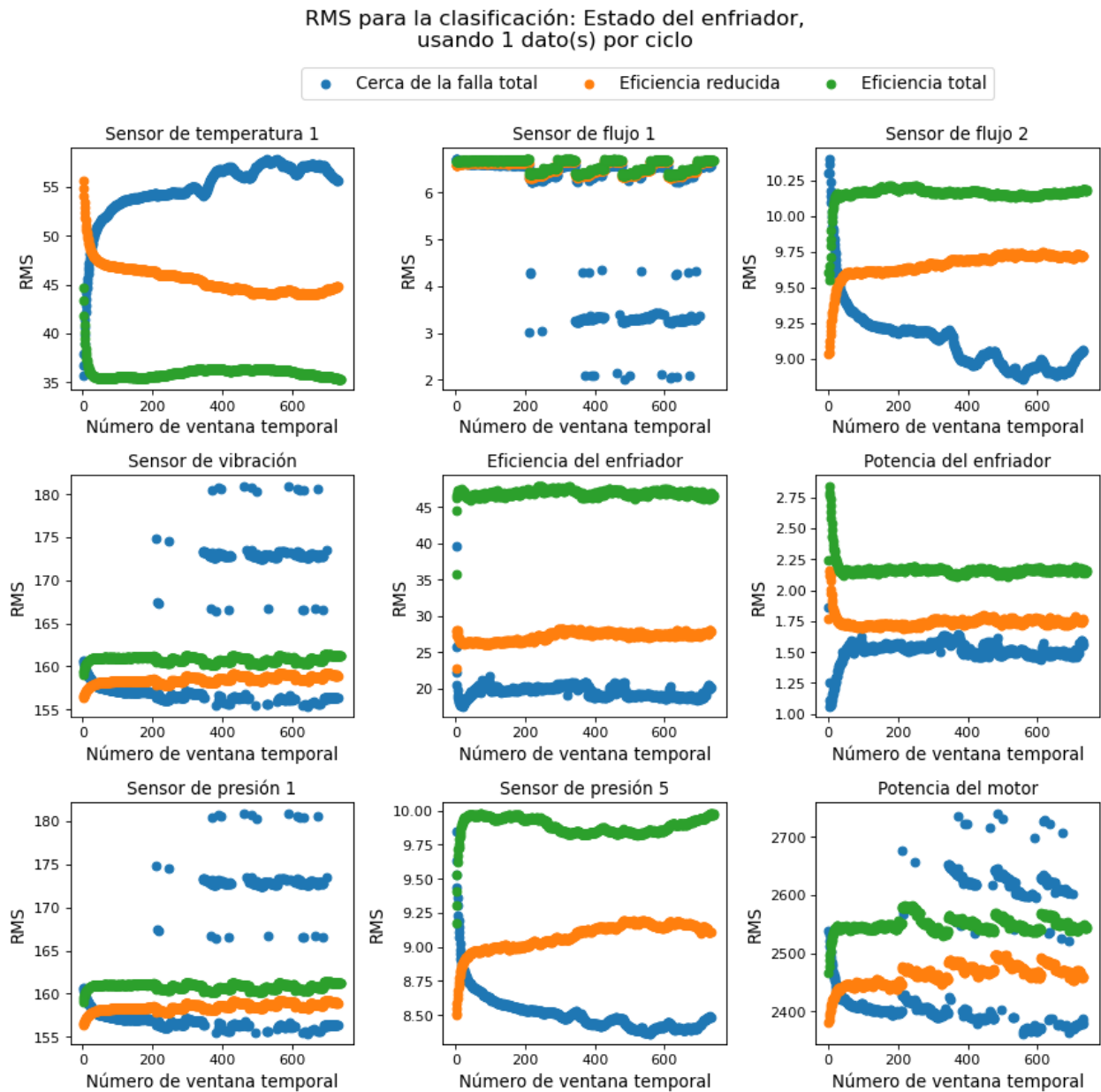


Figura A.101: Gráficos del RMS obtenido según las clases del estado del enfriador.

Varianza para la clasificación: Estado del enfriador,
usando 1 dato(s) por ciclo

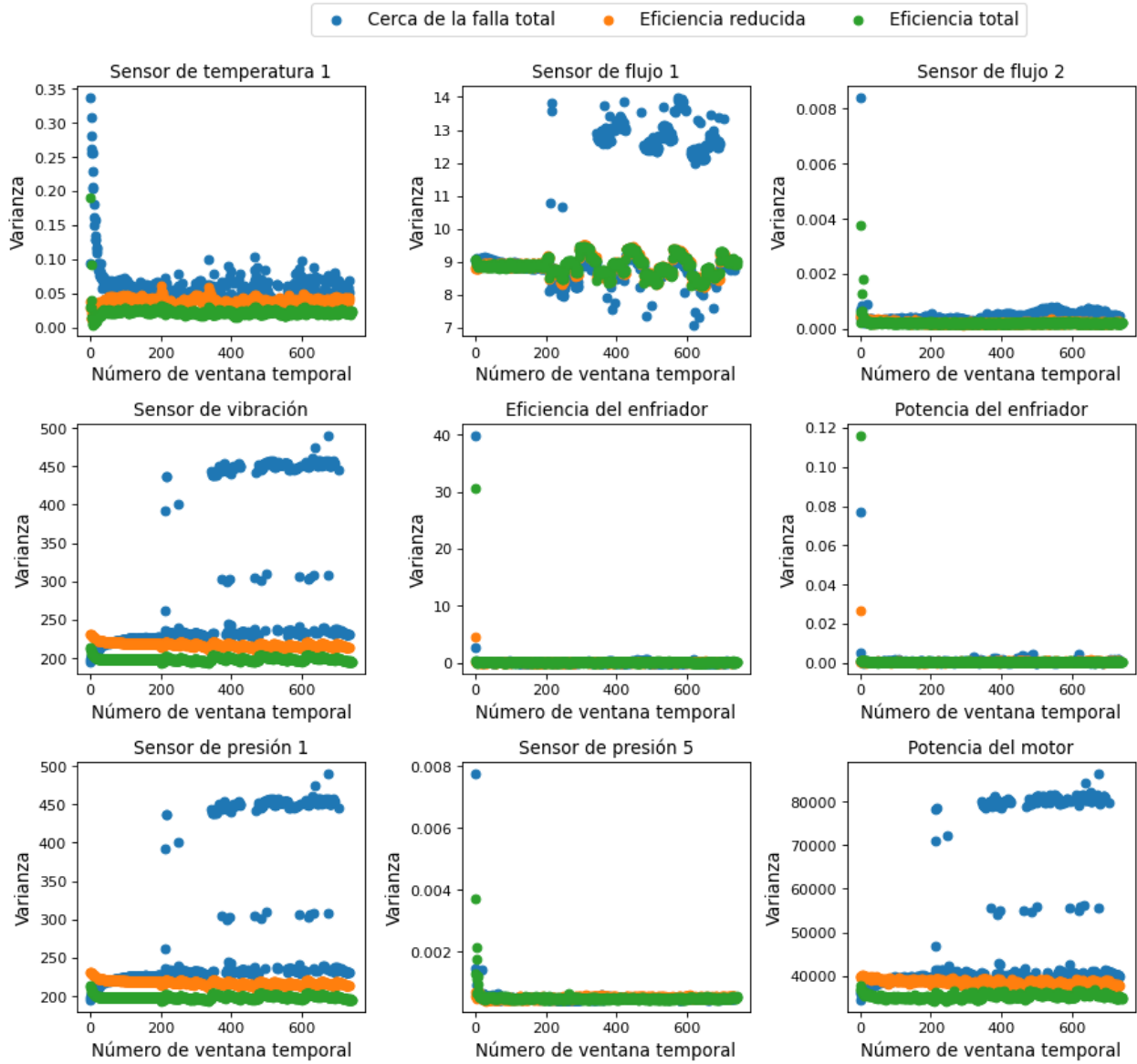


Figura A.102: Gráficos de la varianza obtenida según las clases del estado del enfriador.

Media para la clasificación: Estado del enfriador,
usando 1 dato(s) por ciclo

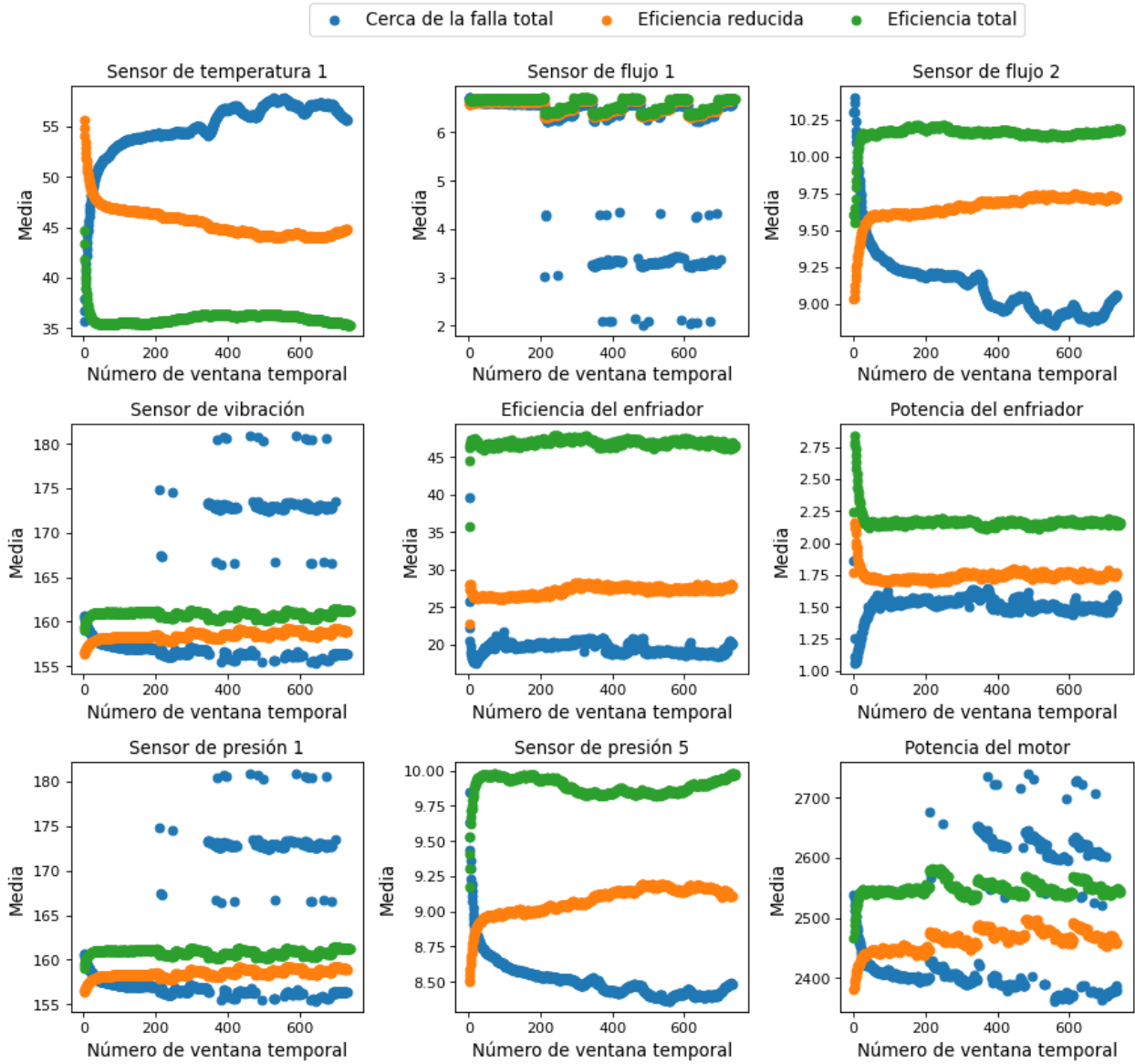


Figura A.103: Gráficos de la media obtenida según las clases del estado del enfriador.

Valor peak to peak para la clasificación: Estado del enfriador,
usando 1 dato(s) por ciclo

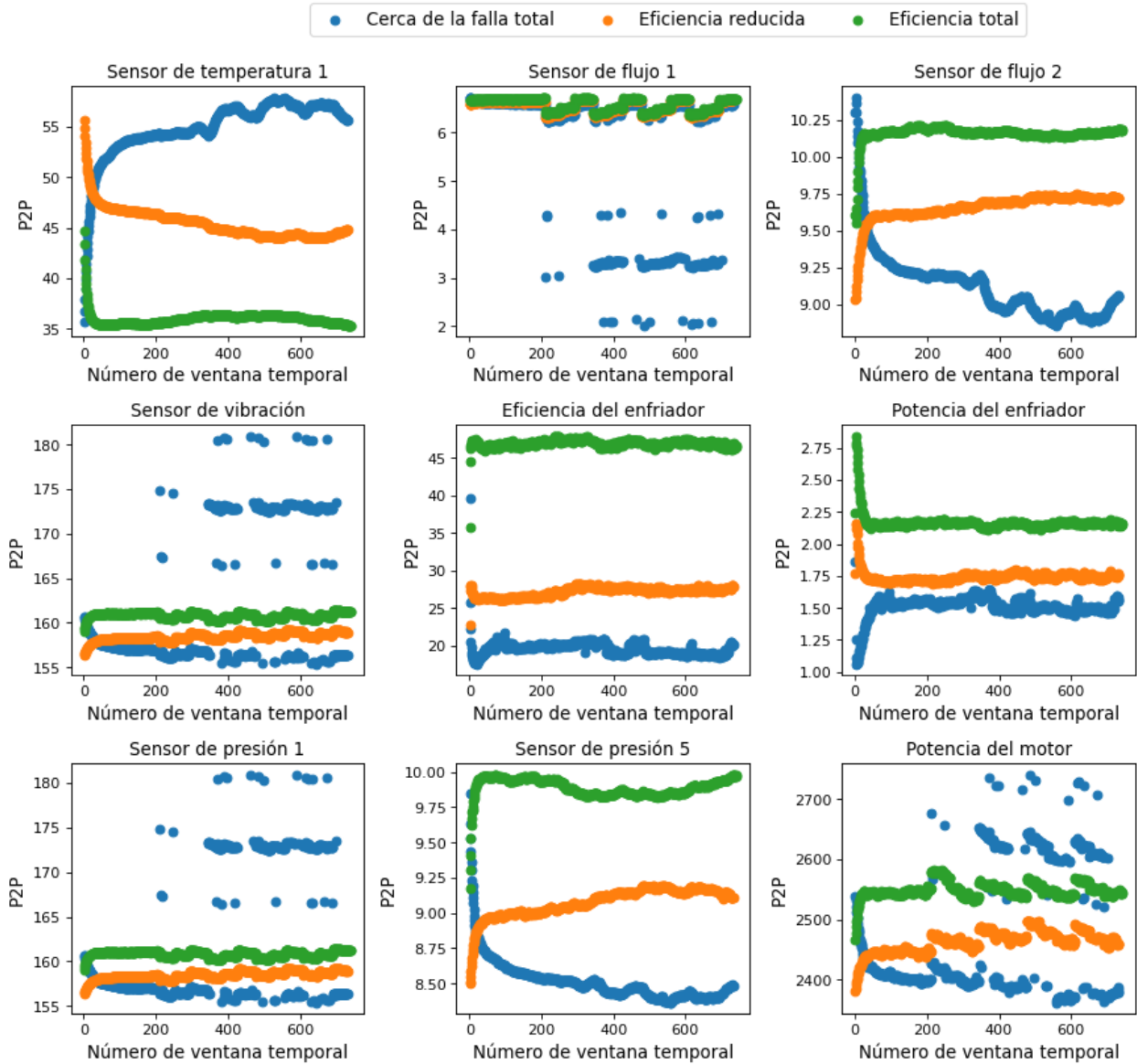


Figura A.104: Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.

A.2.1.2. Gráficos usando 2 datos por ciclo

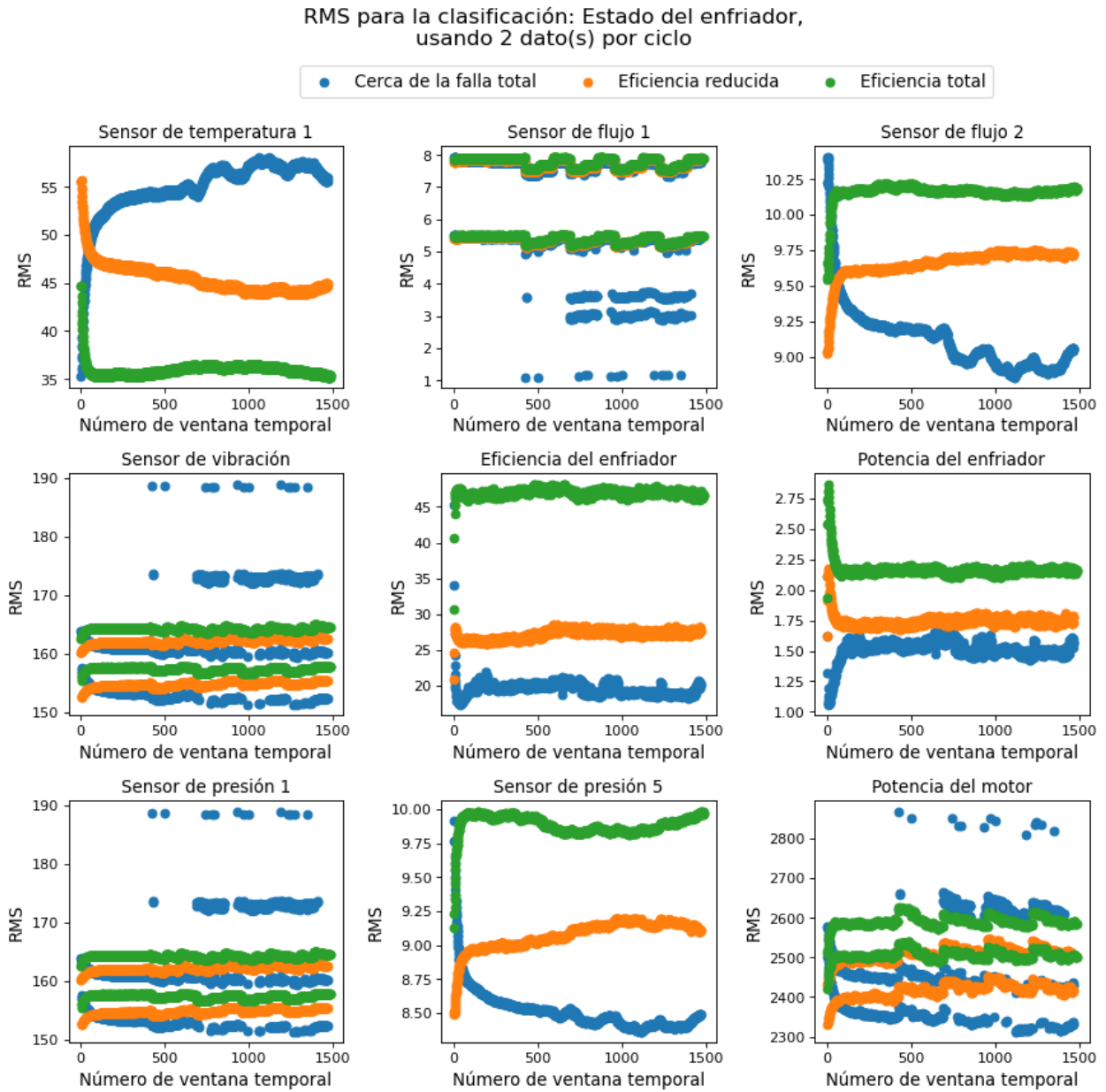


Figura A.105: Gráficos del RMS obtenido según las clases del estado del enfriador.

Varianza para la clasificación: Estado del enfriador, usando 2 dato(s) por ciclo

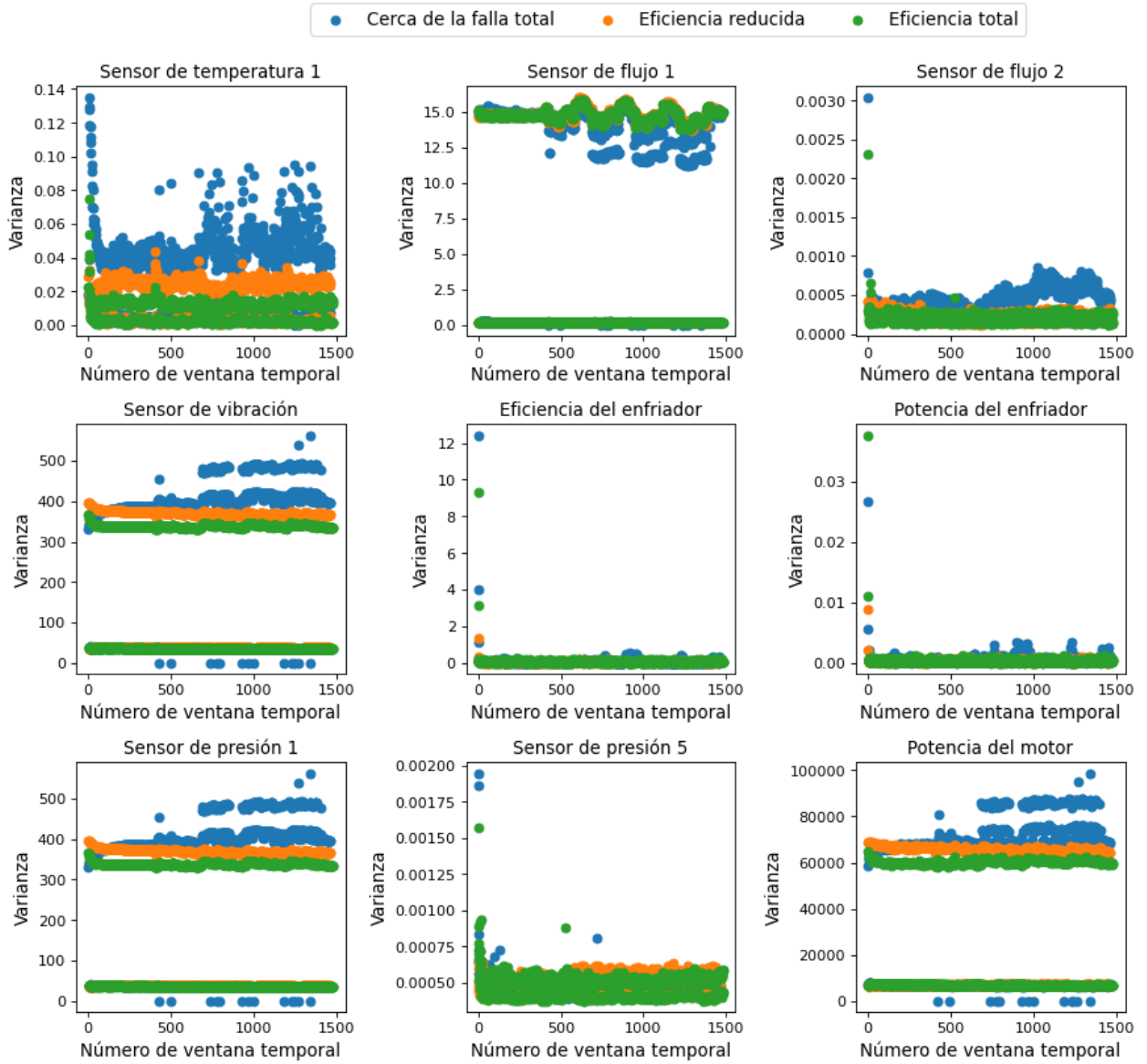


Figura A.106: Gráficos de la varianza obtenida según las clases del estado del enfriador.

Media para la clasificación: Estado del enfriador,
usando 2 dato(s) por ciclo

● Cerca de la falla total ● Eficiencia reducida ● Eficiencia total

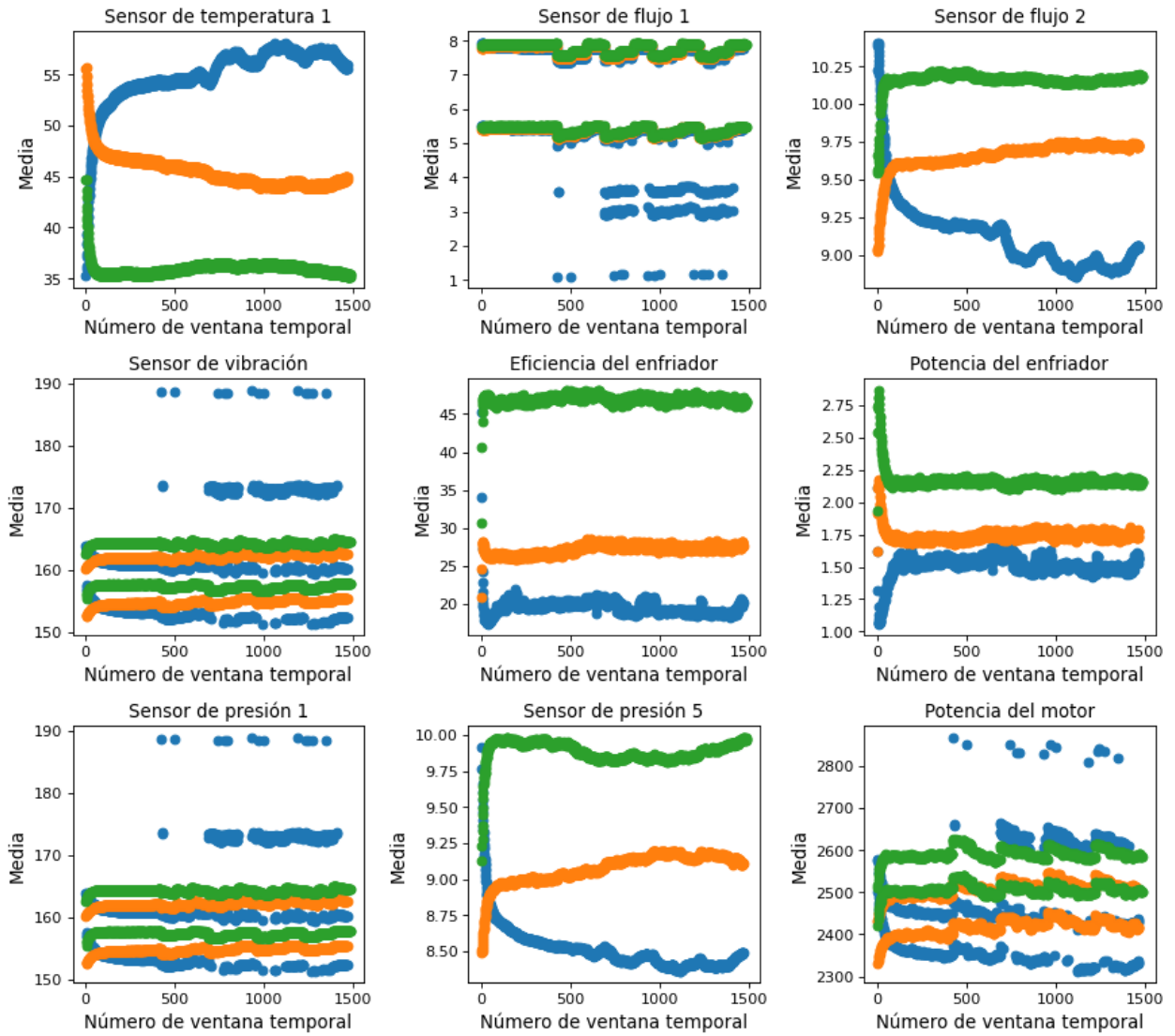


Figura A.107: Gráficos de la media obtenida según las clases del estado del enfriador.

Valor peak to peak para la clasificación: Estado del enfriador,
usando 2 dato(s) por ciclo

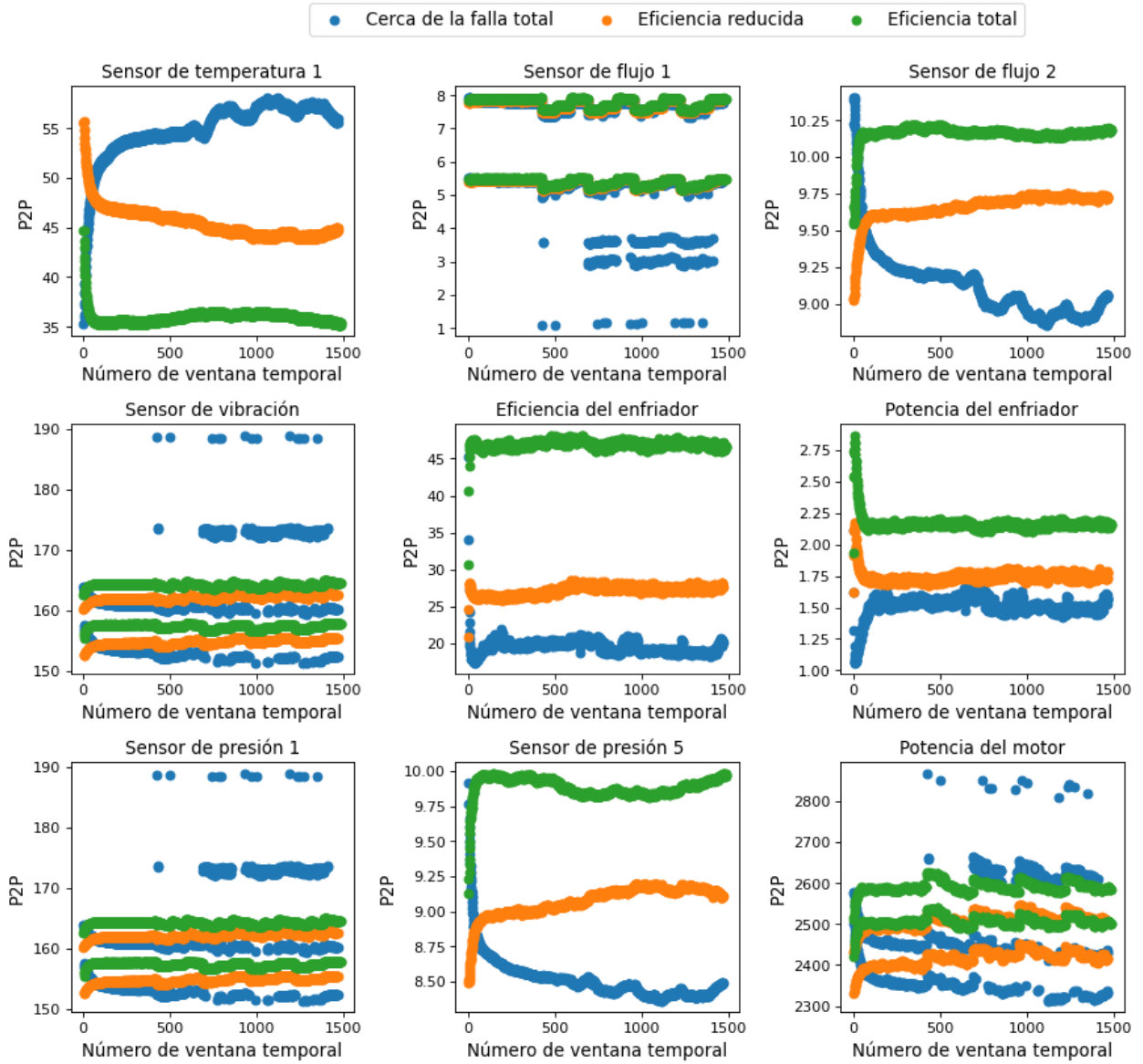


Figura A.108: Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.

A.2.1.3. Gráficos usando 6 datos por ciclo



Figura A.109: Gráficos del RMS obtenido según las clases del estado del enfriador.

Varianza para la clasificación: Estado del enfriador, usando 6 dato(s) por ciclo

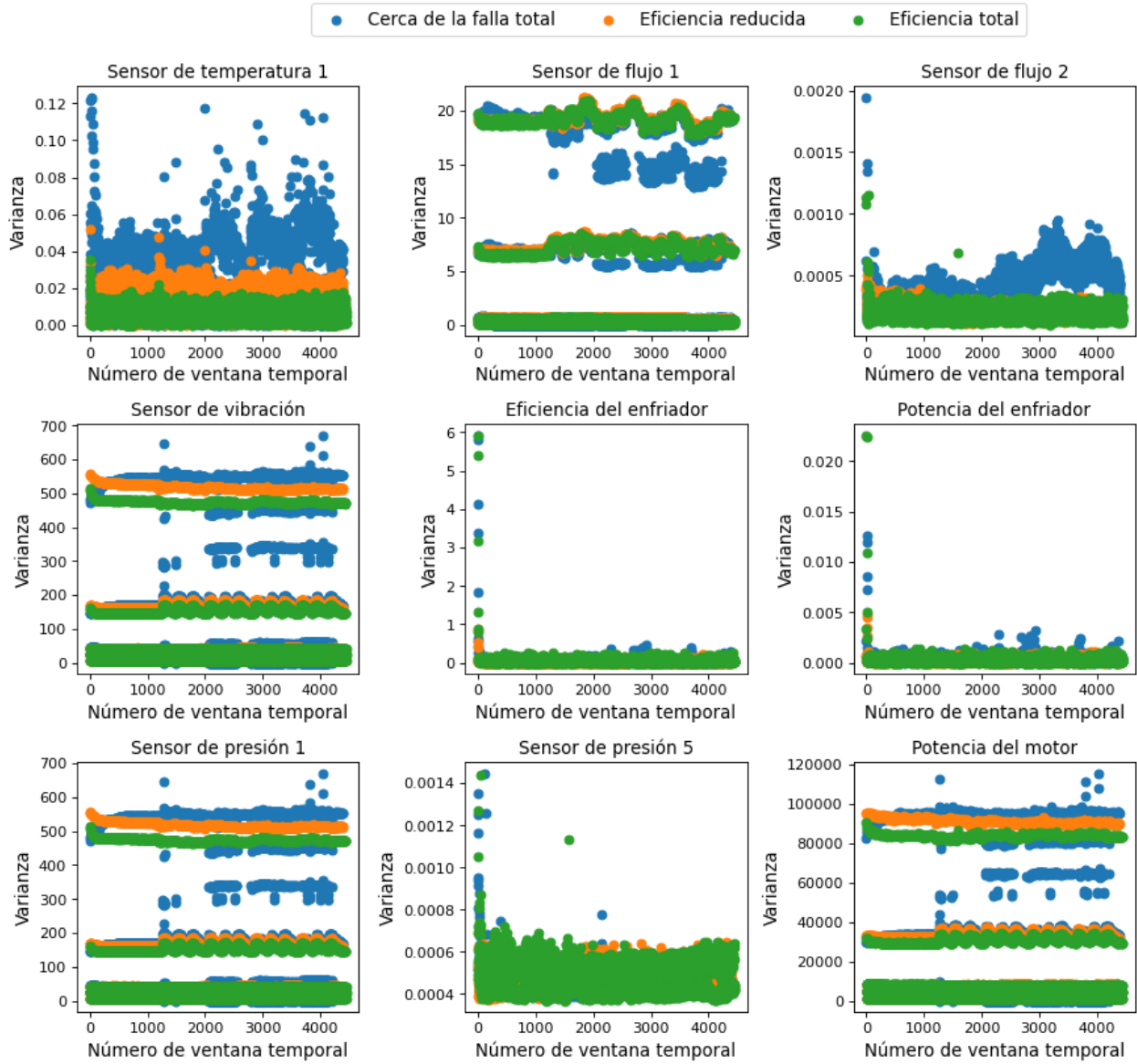


Figura A.110: Gráficos de la varianza obtenida según las clases del estado del enfriador.

Media para la clasificación: Estado del enfriador,
usando 6 dato(s) por ciclo

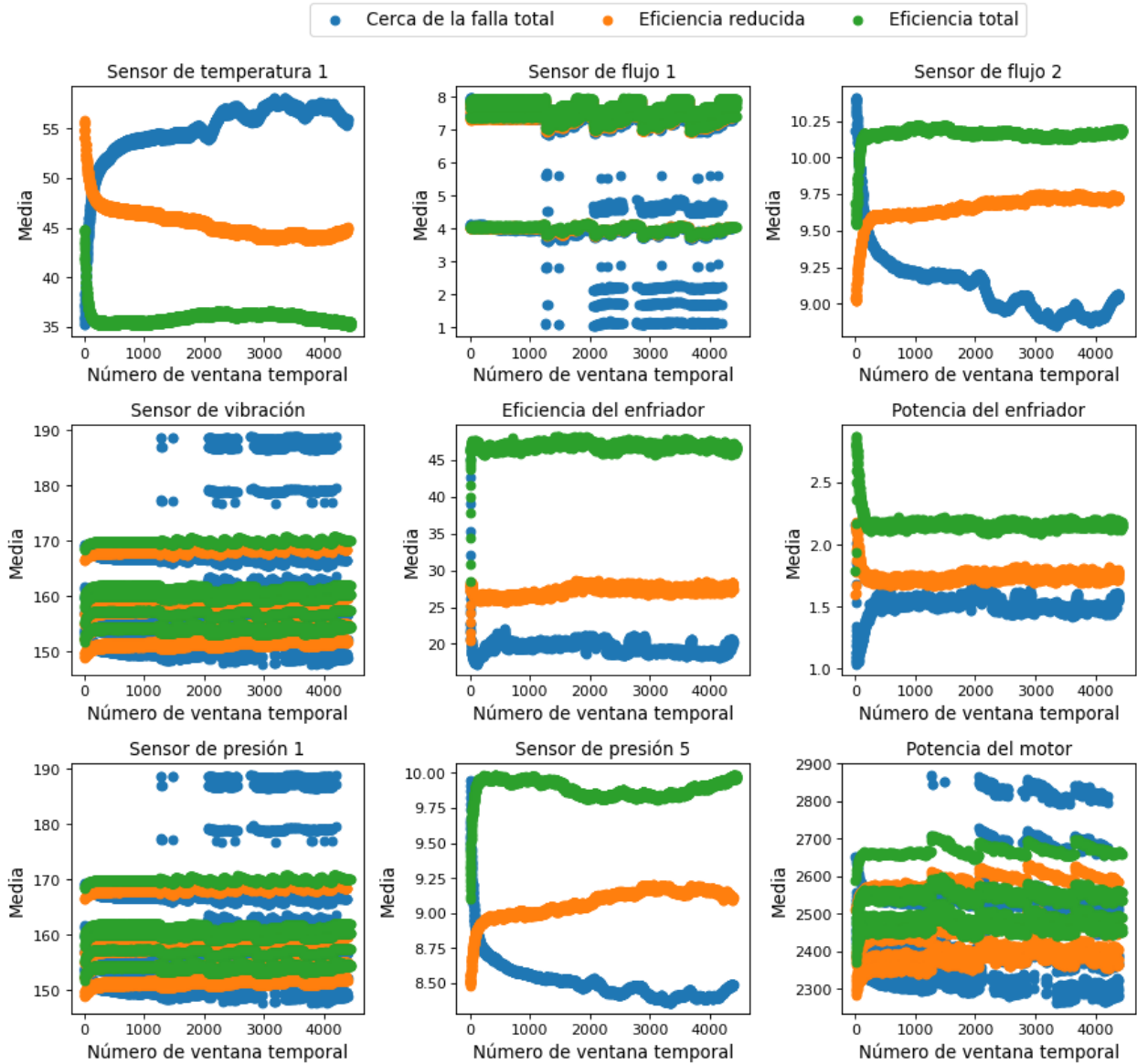


Figura A.111: Gráficos de la media obtenida según las clases del estado del enfriador.

Valor peak to peak para la clasificación: Estado del enfriador,
usando 6 dato(s) por ciclo

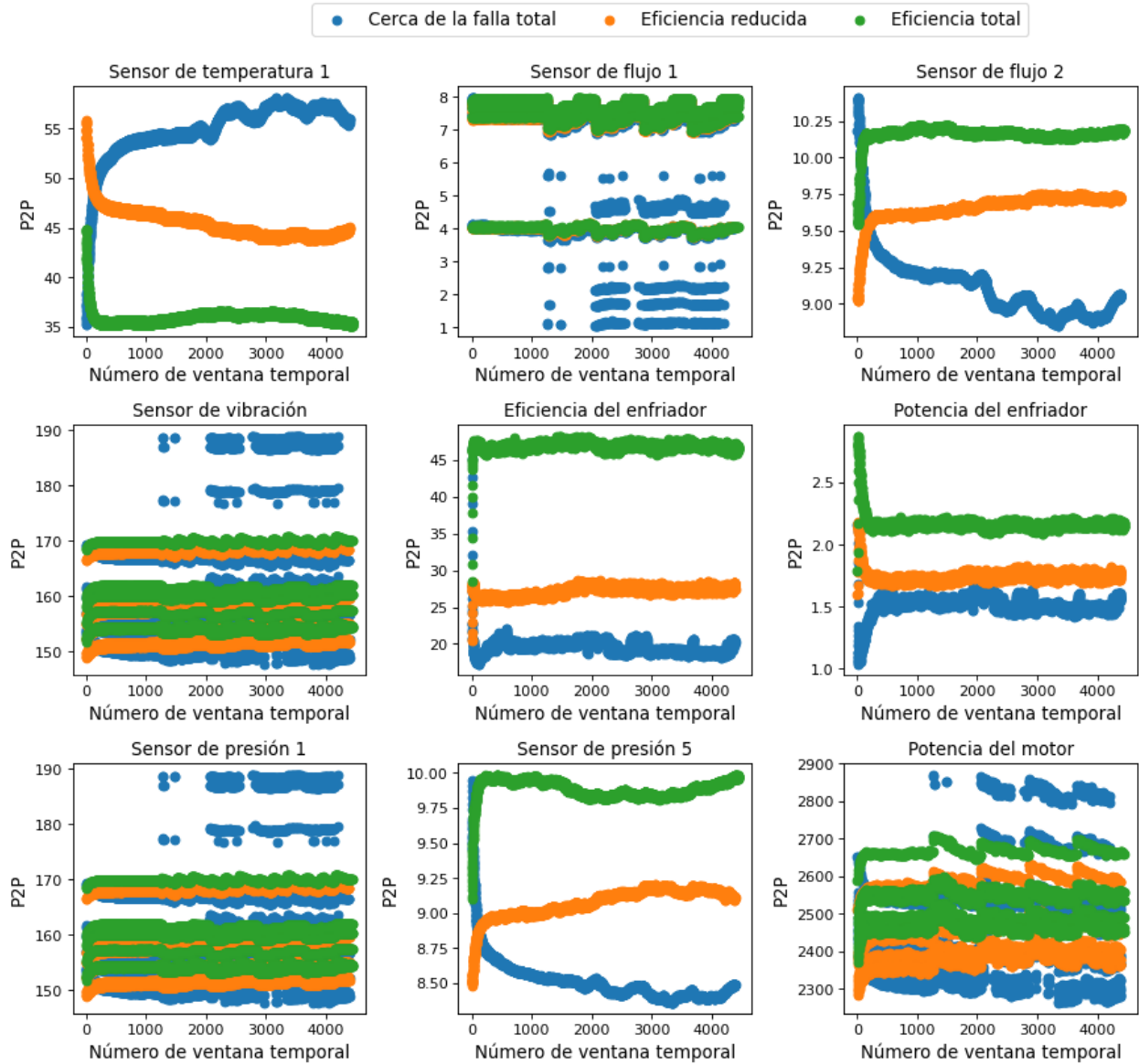


Figura A.112: Gráficos del valor peak to peak obtenido según las clases del estado del enfriador.

A.2.2. Gráficos de la válvula

A.2.2.1. Gráficos usando 1 dato por ciclo

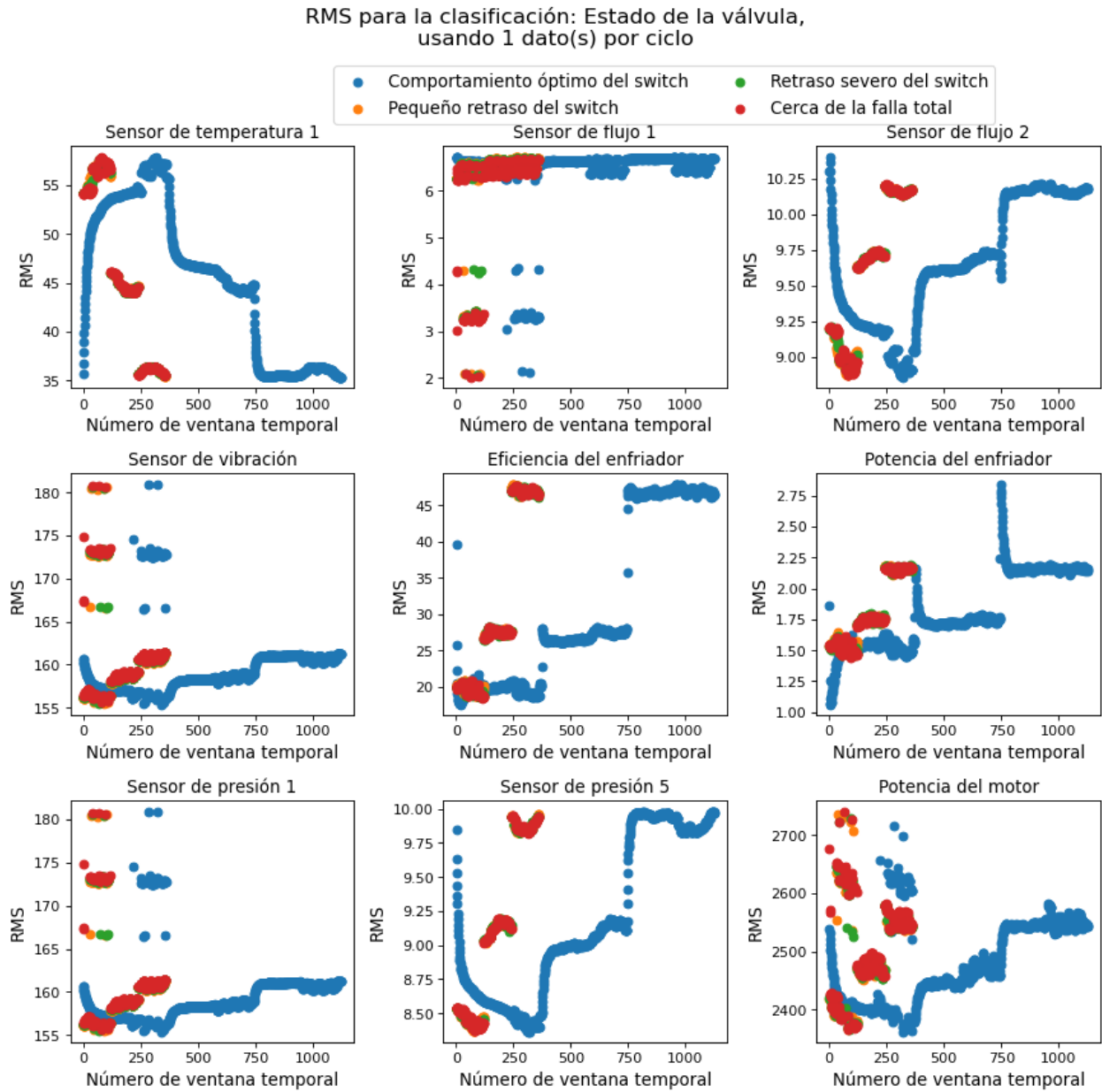


Figura A.113: Gráficos del RMS obtenido según las clases del estado de la válvula.

Varianza para la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo

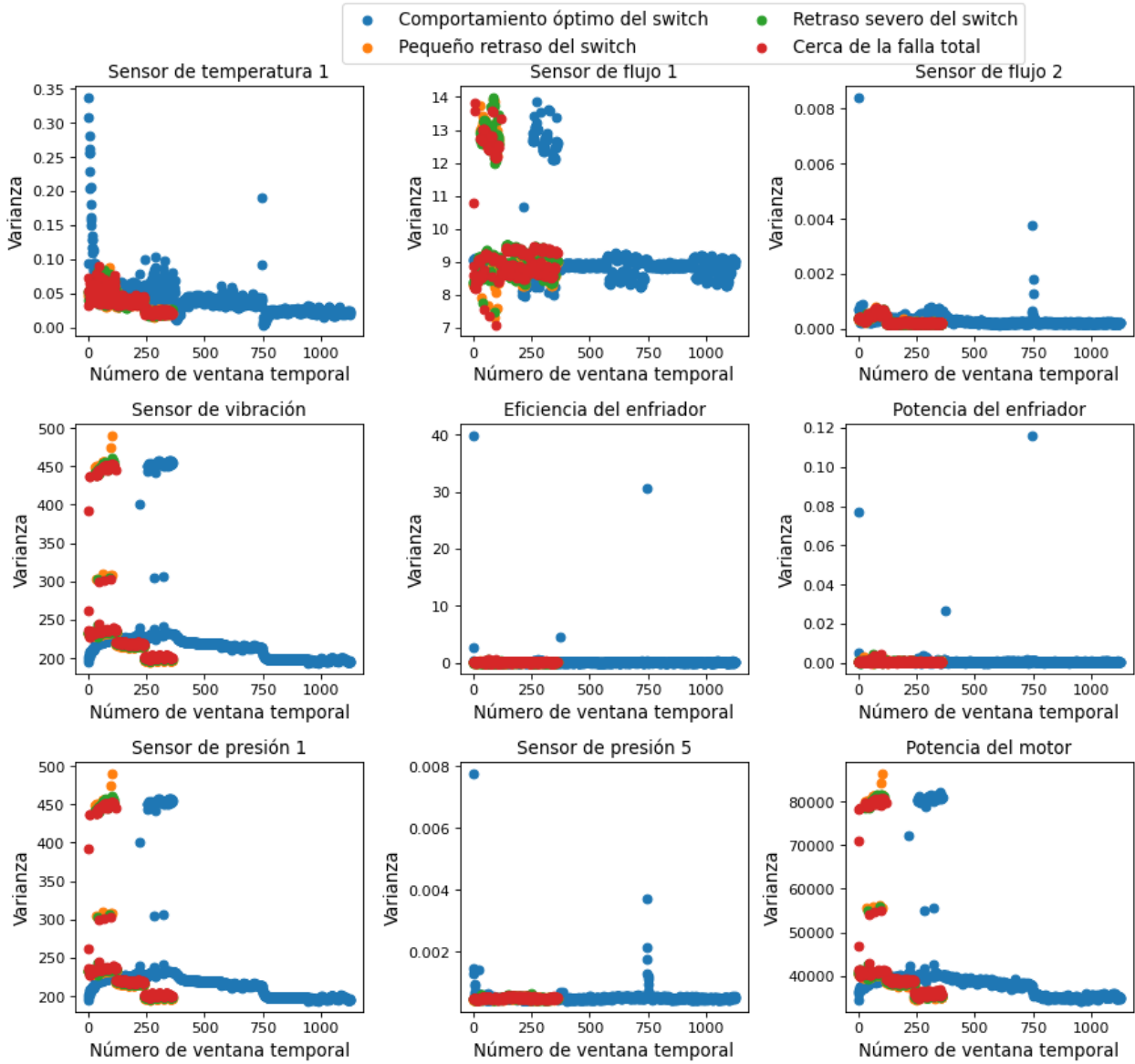


Figura A.114: Gráficos de la varianza obtenida según las clases del estado de la válvula.

Media para la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo

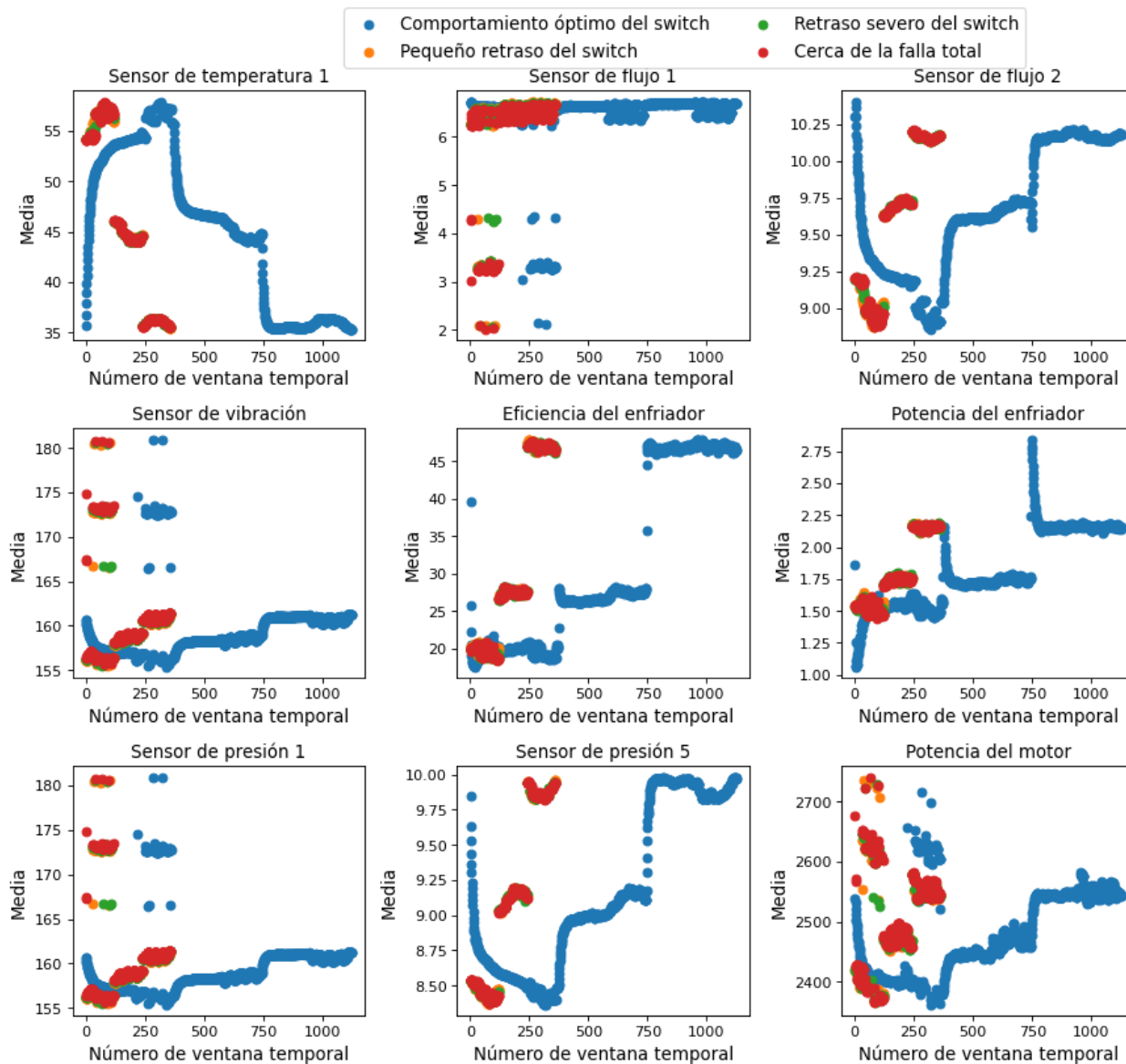


Figura A.115: Gráficos de la media obtenida según las clases del estado de la válvula.

Valor peak to peak para la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo

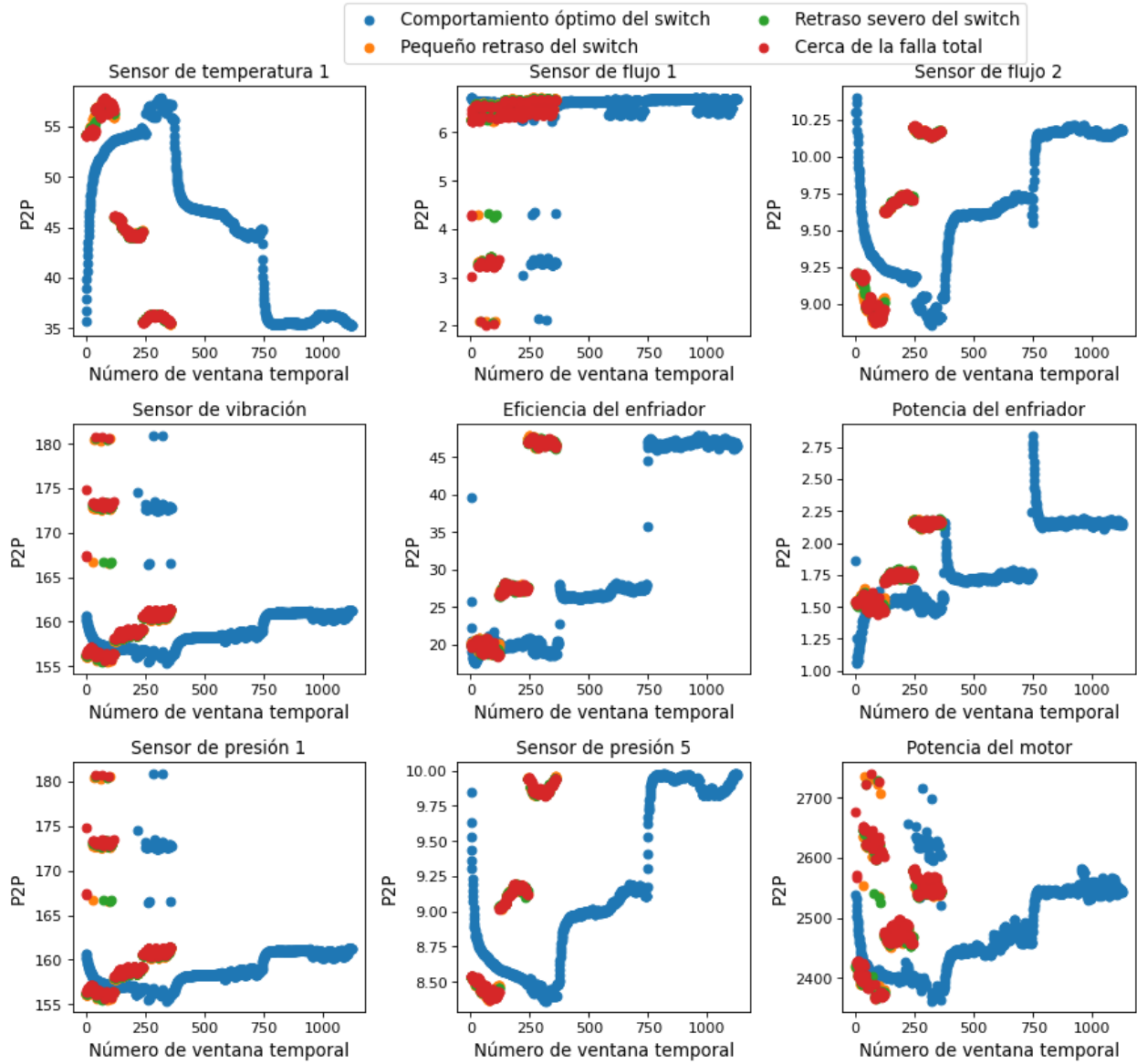


Figura A.116: Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.

A.2.2.2. Gráficos usando 2 datos por ciclo

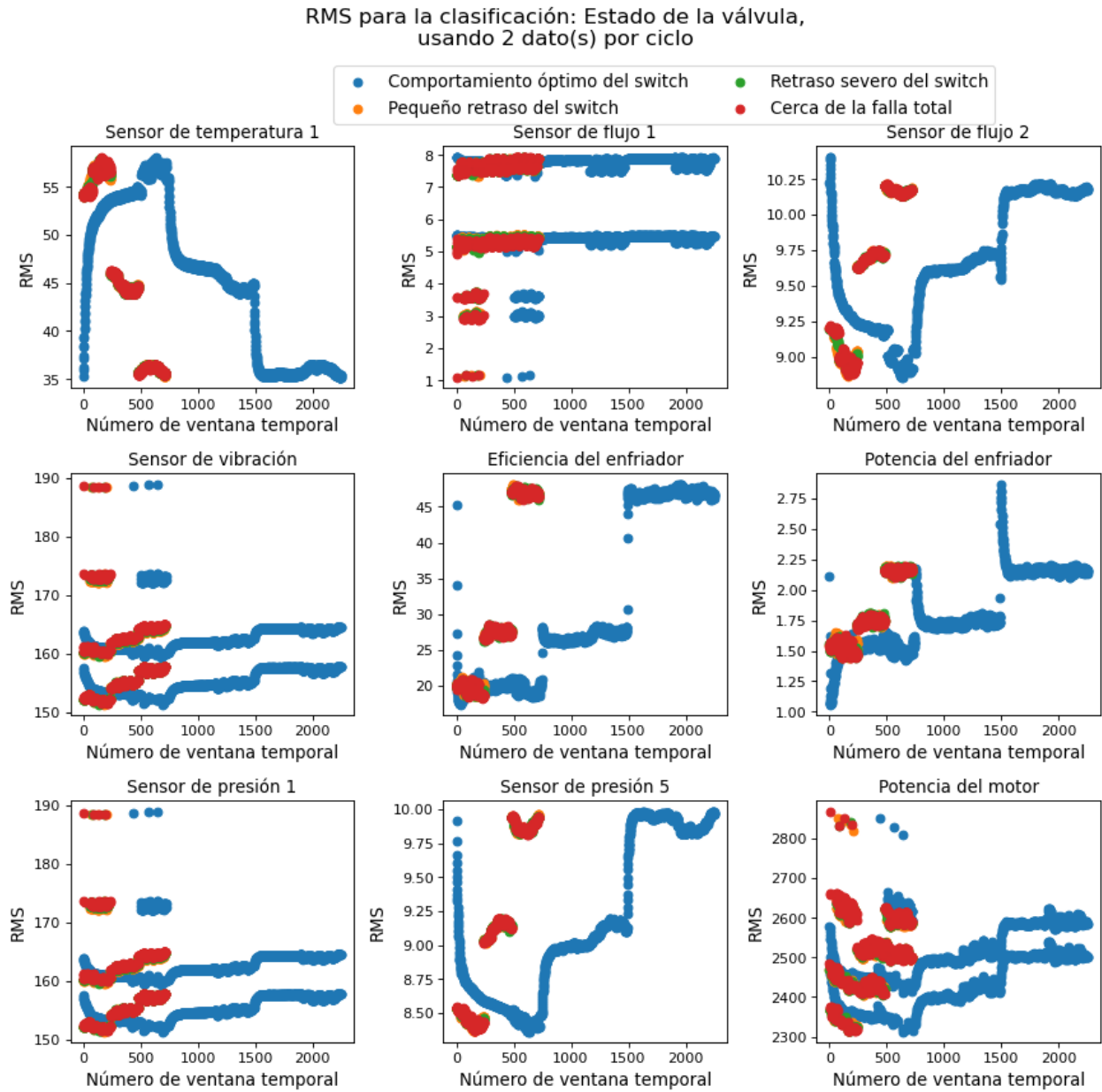


Figura A.117: Gráficos del RMS obtenido según las clases del estado de la válvula.

Varianza para la clasificación: Estado de la válvula,
usando 2 dato(s) por ciclo

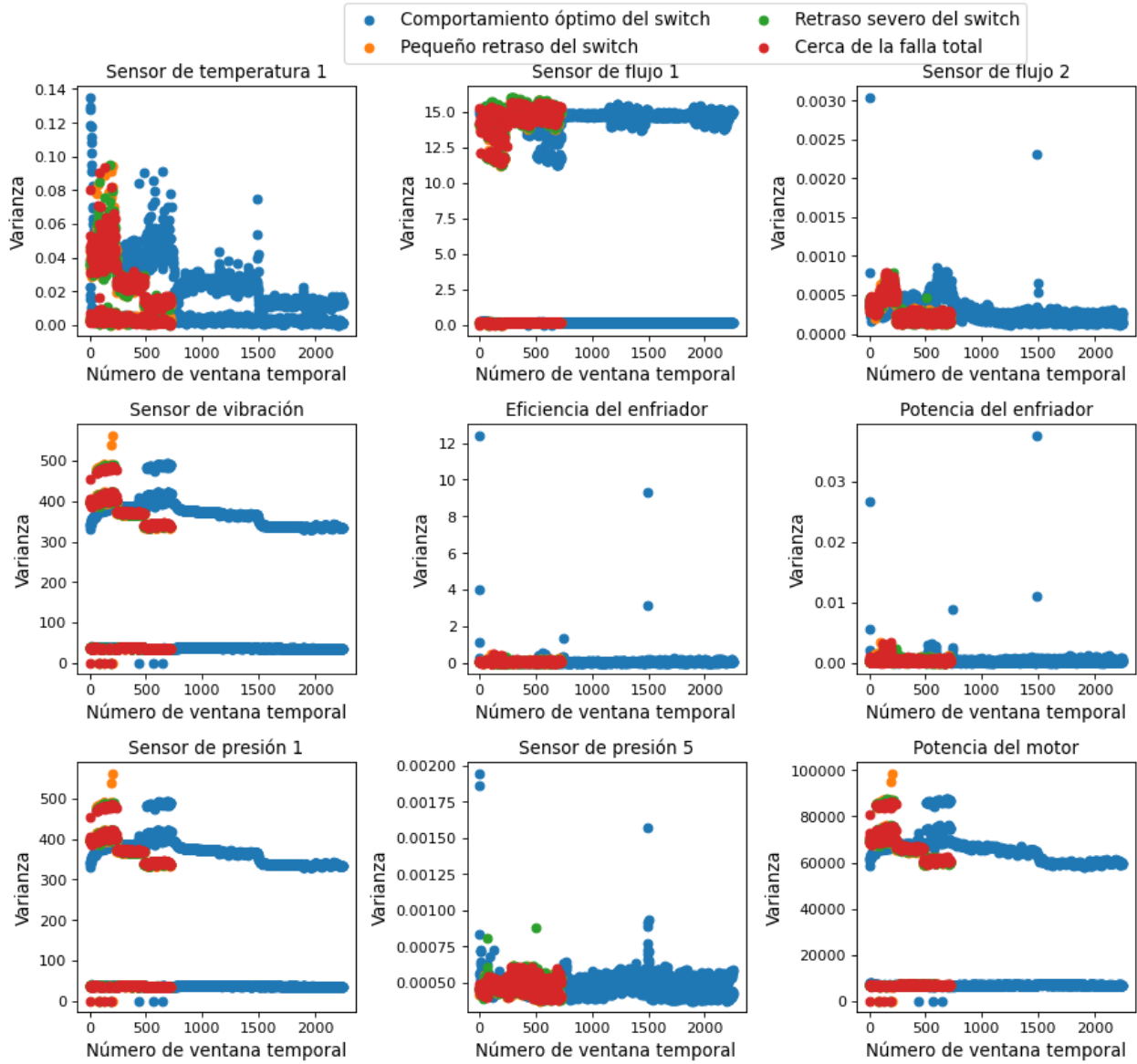


Figura A.118: Gráficos de la varianza obtenida según las clases del estado de la válvula.

Media para la clasificación: Estado de la válvula,
usando 2 dato(s) por ciclo

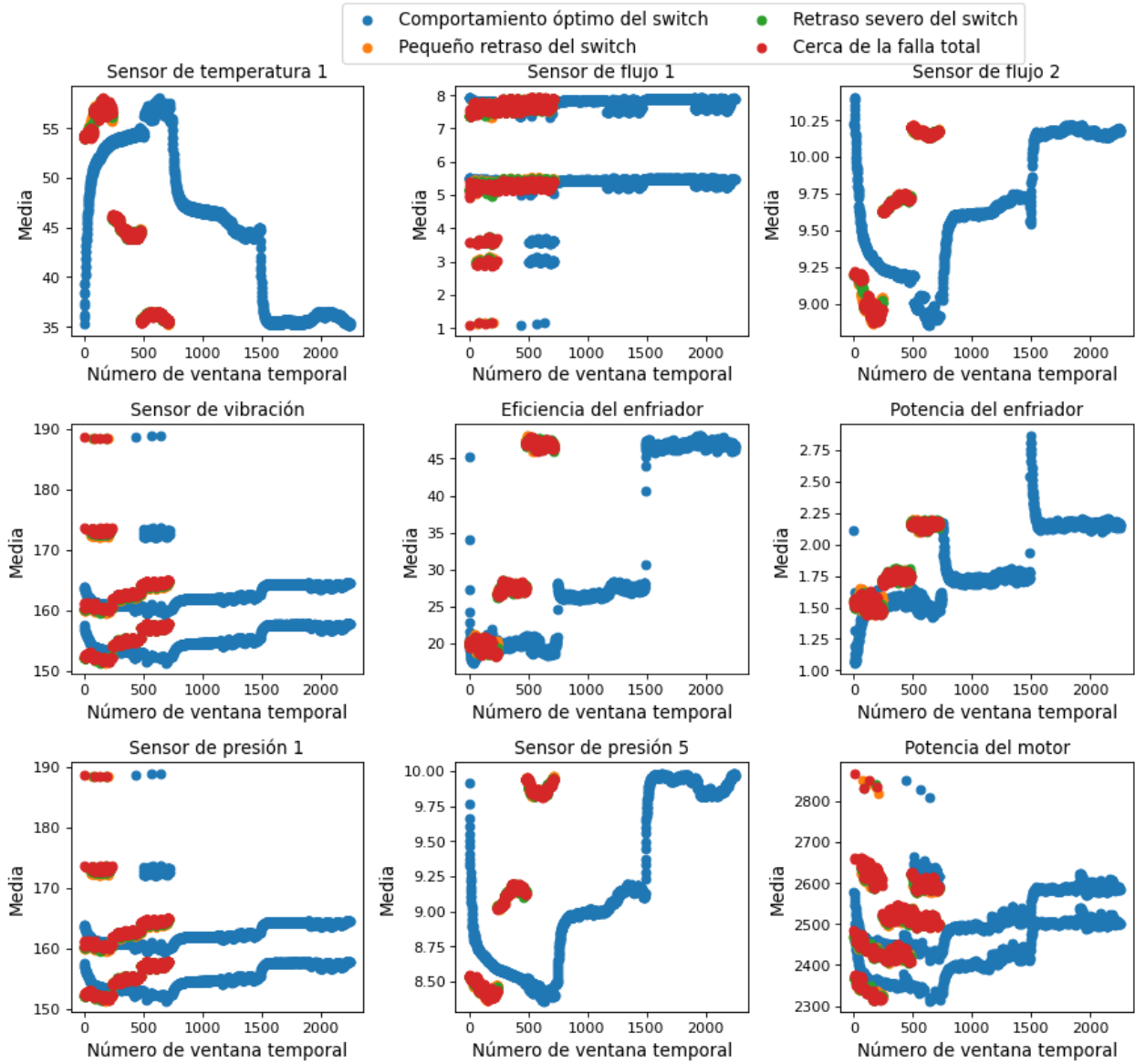


Figura A.119: Gráficos de la media obtenida según las clases del estado de la válvula.

Valor peak to peak para la clasificación: Estado de la válvula,
usando 2 dato(s) por ciclo

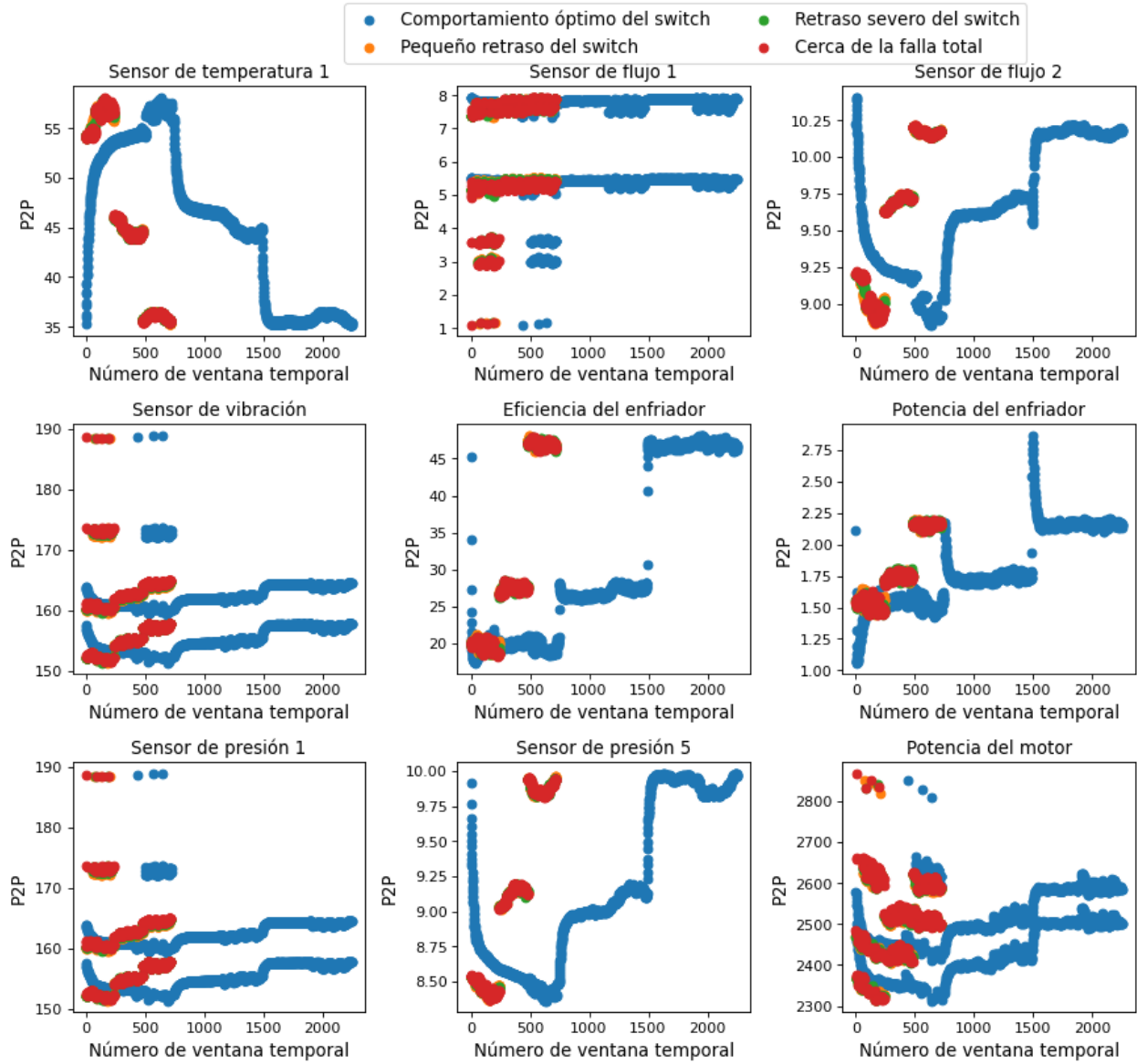


Figura A.120: Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.

A.2.2.3. Gráficos usando 6 datos por ciclo

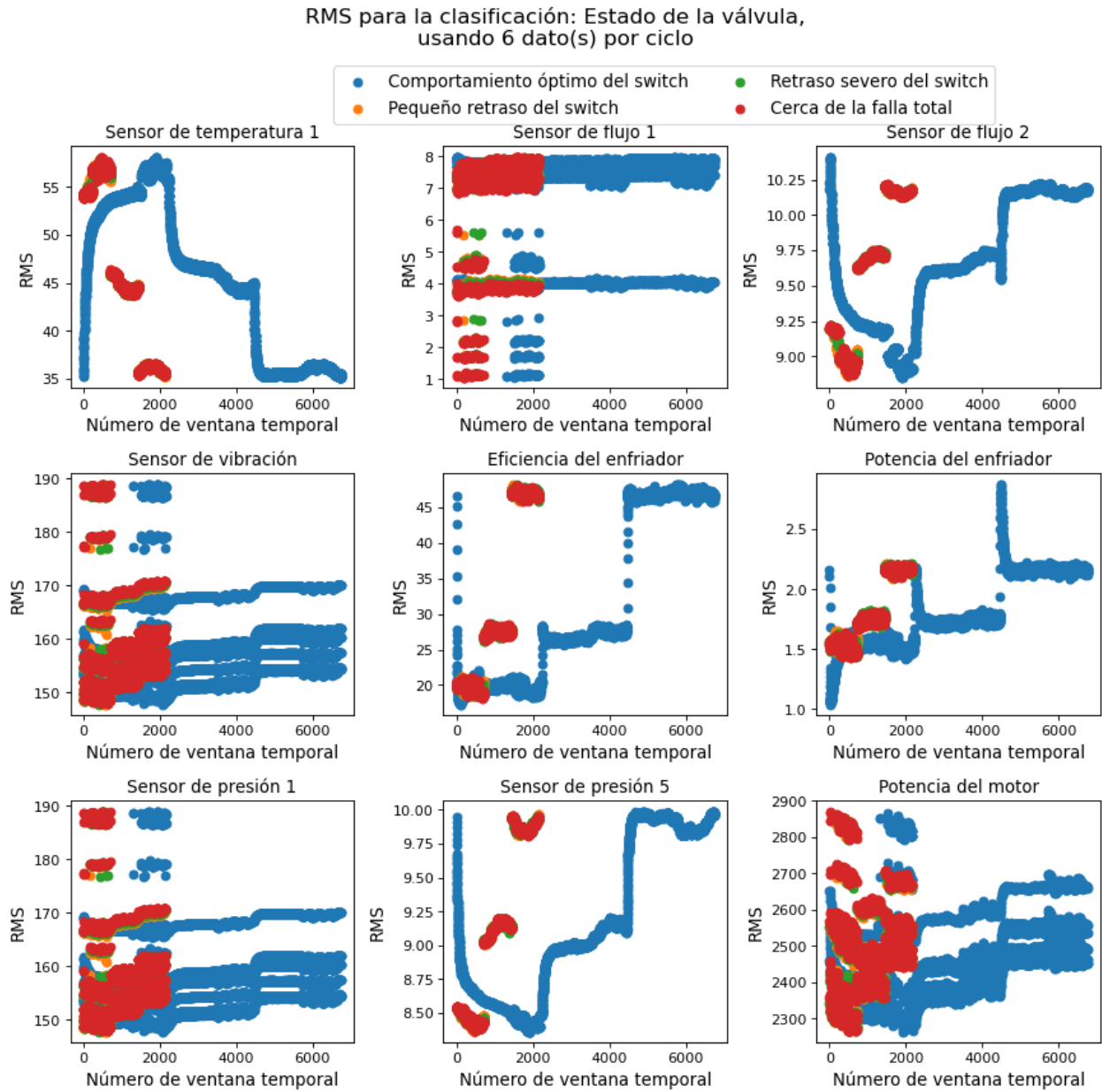


Figura A.121: Gráficos del RMS obtenido según las clases del estado de la válvula.

Varianza para la clasificación: Estado de la válvula,
usando 6 dato(s) por ciclo

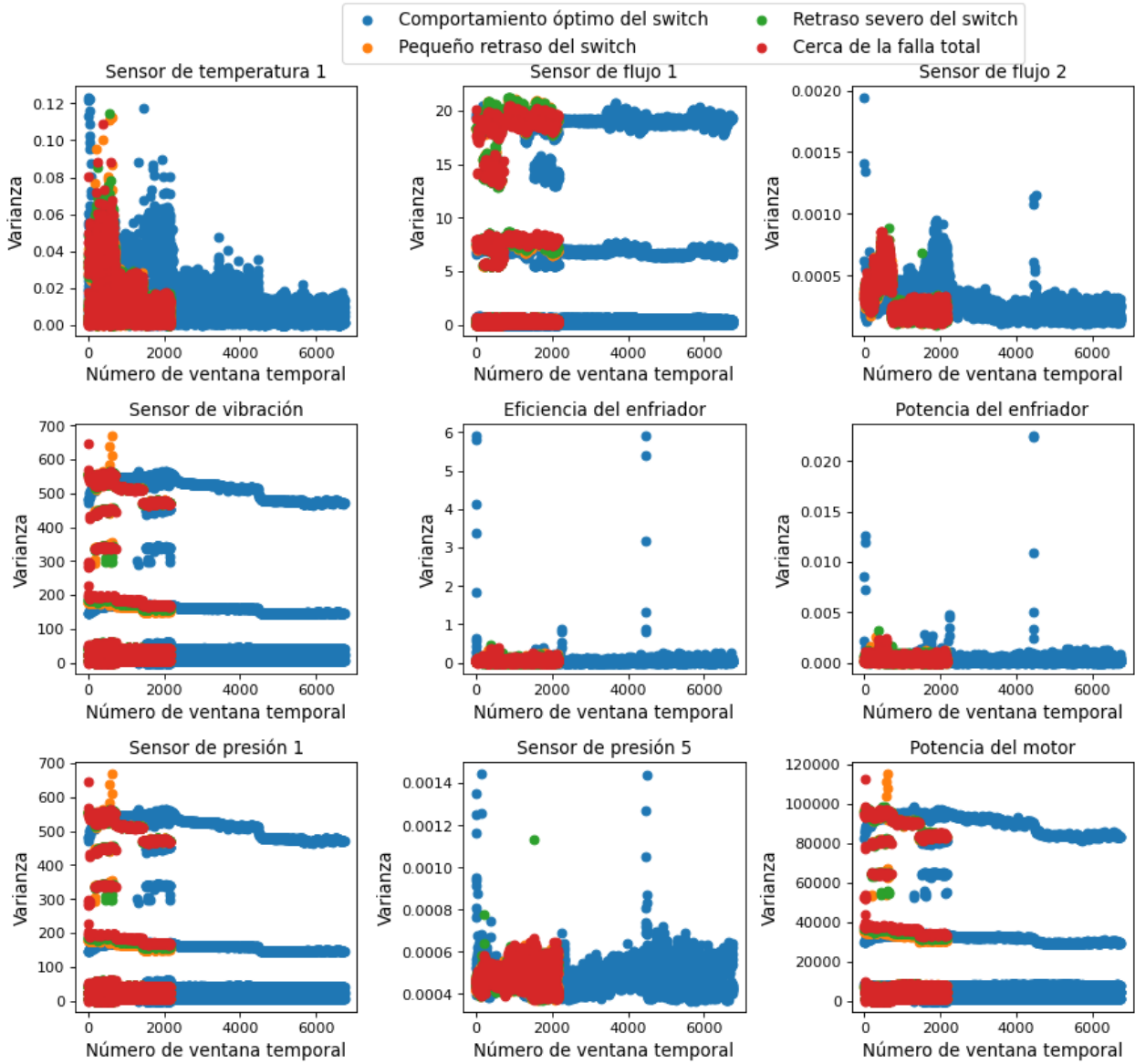


Figura A.122: Gráficos de la varianza obtenida según las clases del estado de la válvula.

Media para la clasificación: Estado de la válvula,
usando 6 dato(s) por ciclo

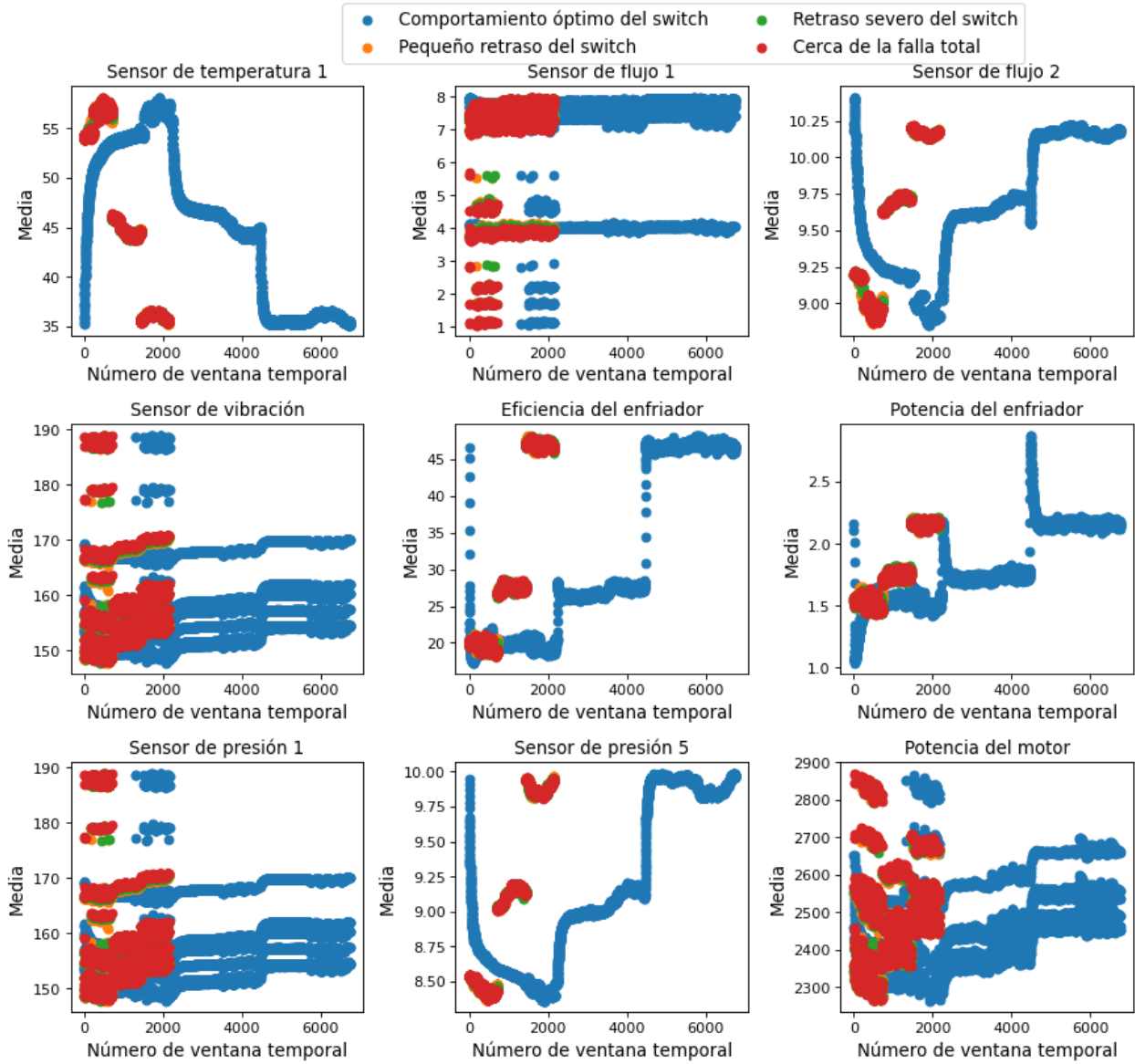


Figura A.123: Gráficos de la media obtenida según las clases del estado de la válvula.

Valor peak to peak para la clasificación: Estado de la válvula, usando 6 dato(s) por ciclo

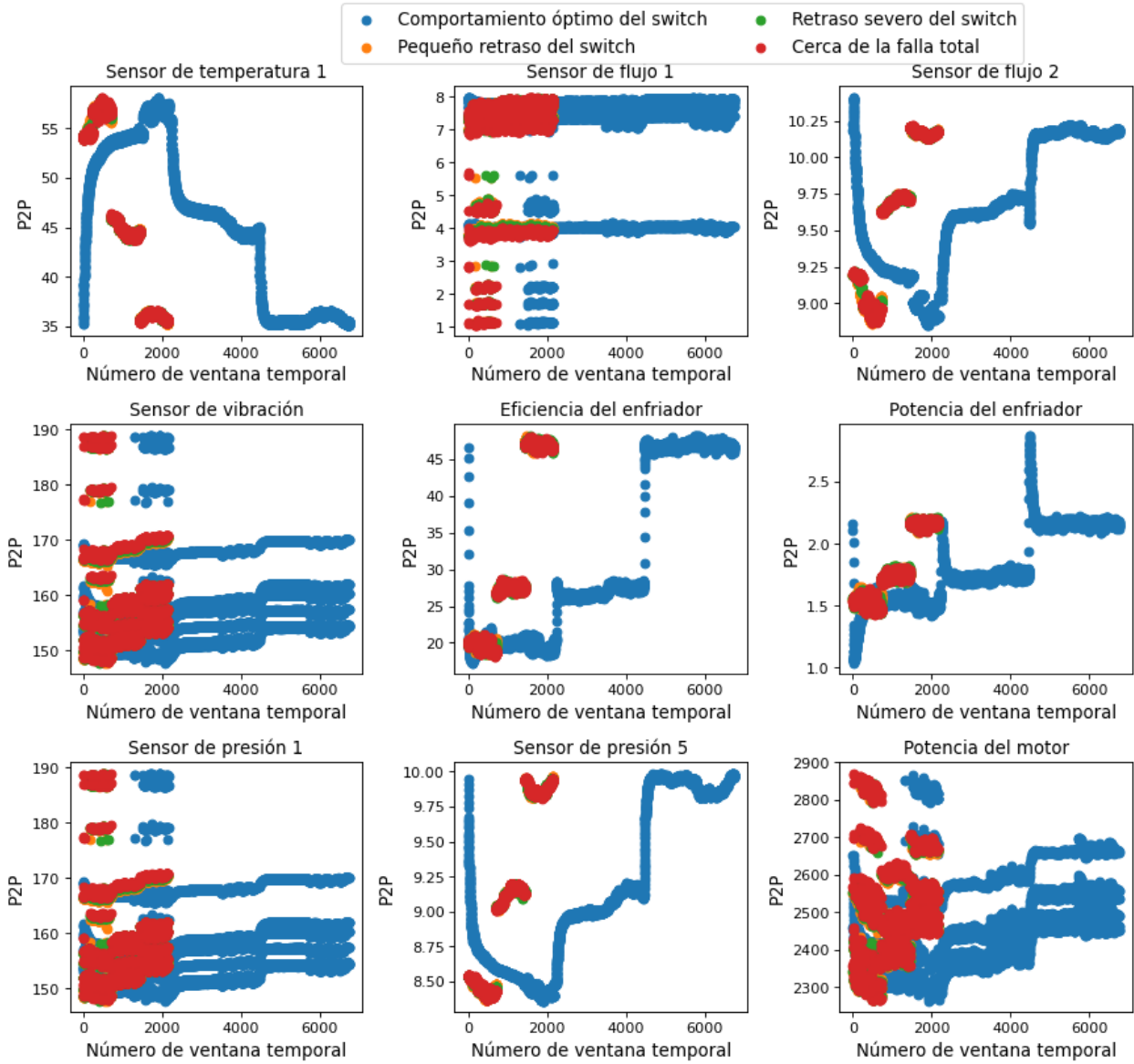


Figura A.124: Gráficos del valor peak to peak obtenido según las clases del estado de la válvula.

A.2.3. Gráficos de la bomba

A.2.3.1. Gráficos usando 1 dato por ciclo

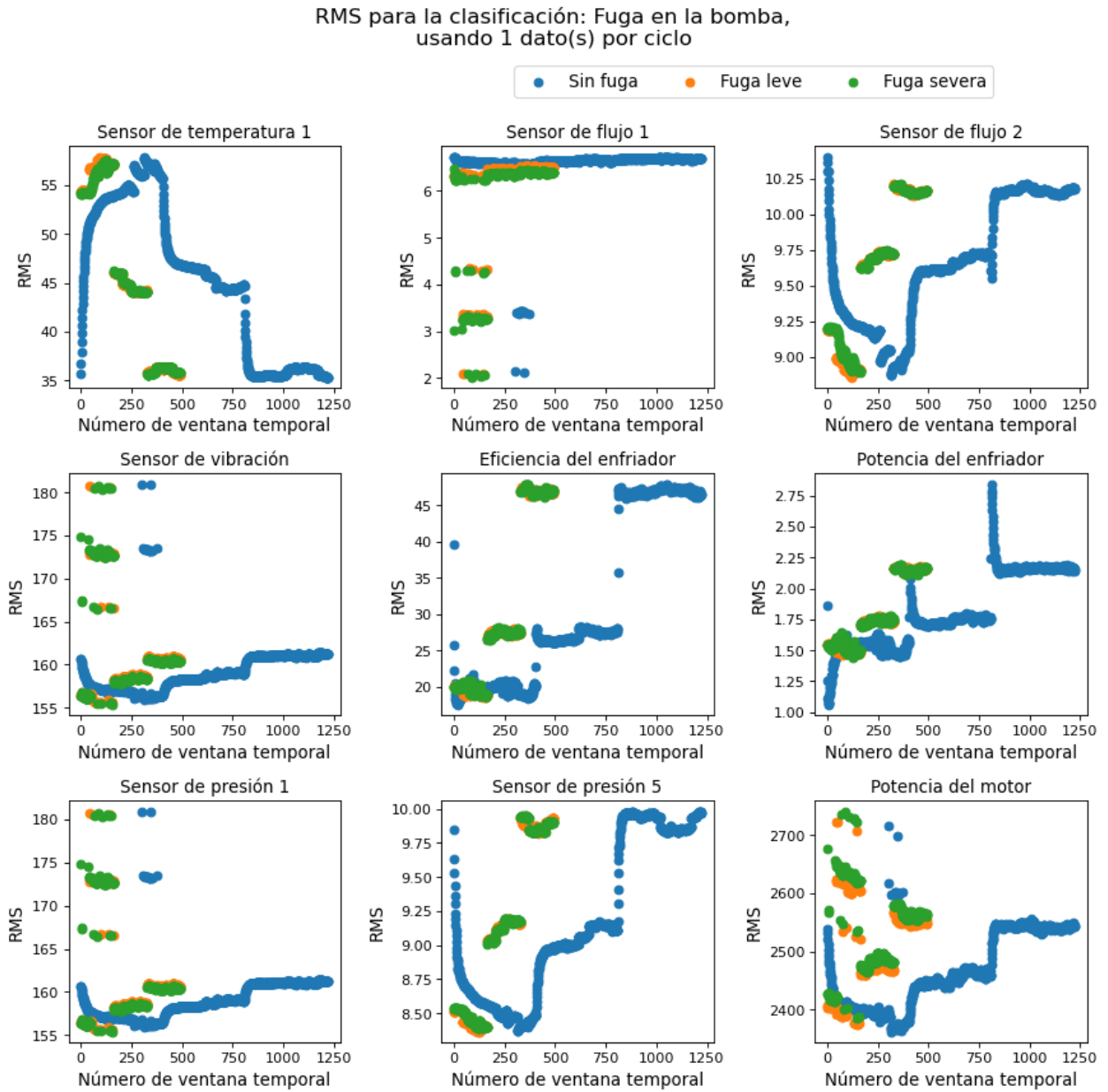


Figura A.125: Gráficos del RMS obtenido según las clases del estado de la bomba.

Varianza para la clasificación: Fuga en la bomba,
usando 1 dato(s) por ciclo

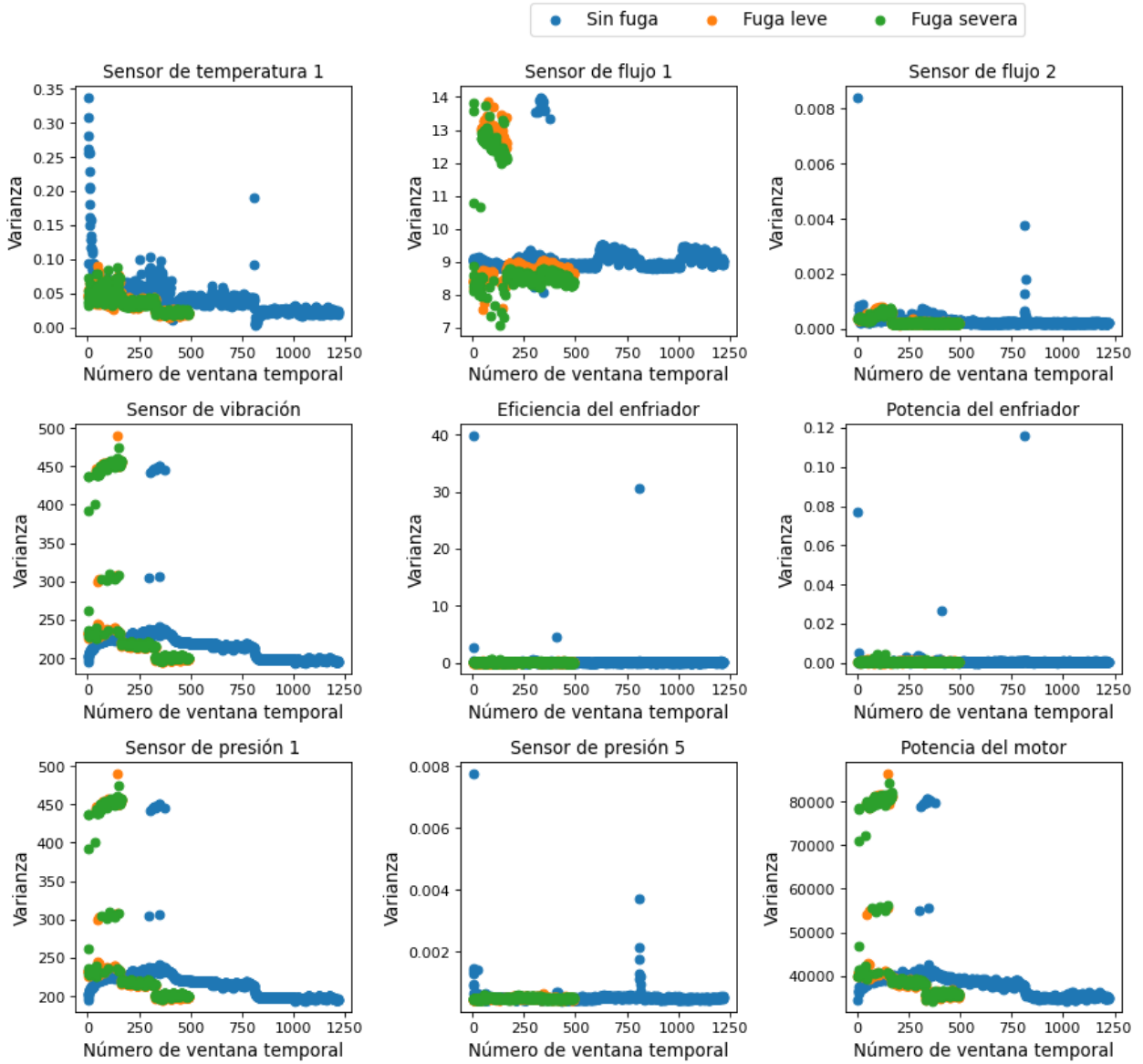


Figura A.126: Gráficos de la varianza obtenida según las clases del estado de la bomba.

Media para la clasificación: Fuga en la bomba,
usando 1 dato(s) por ciclo

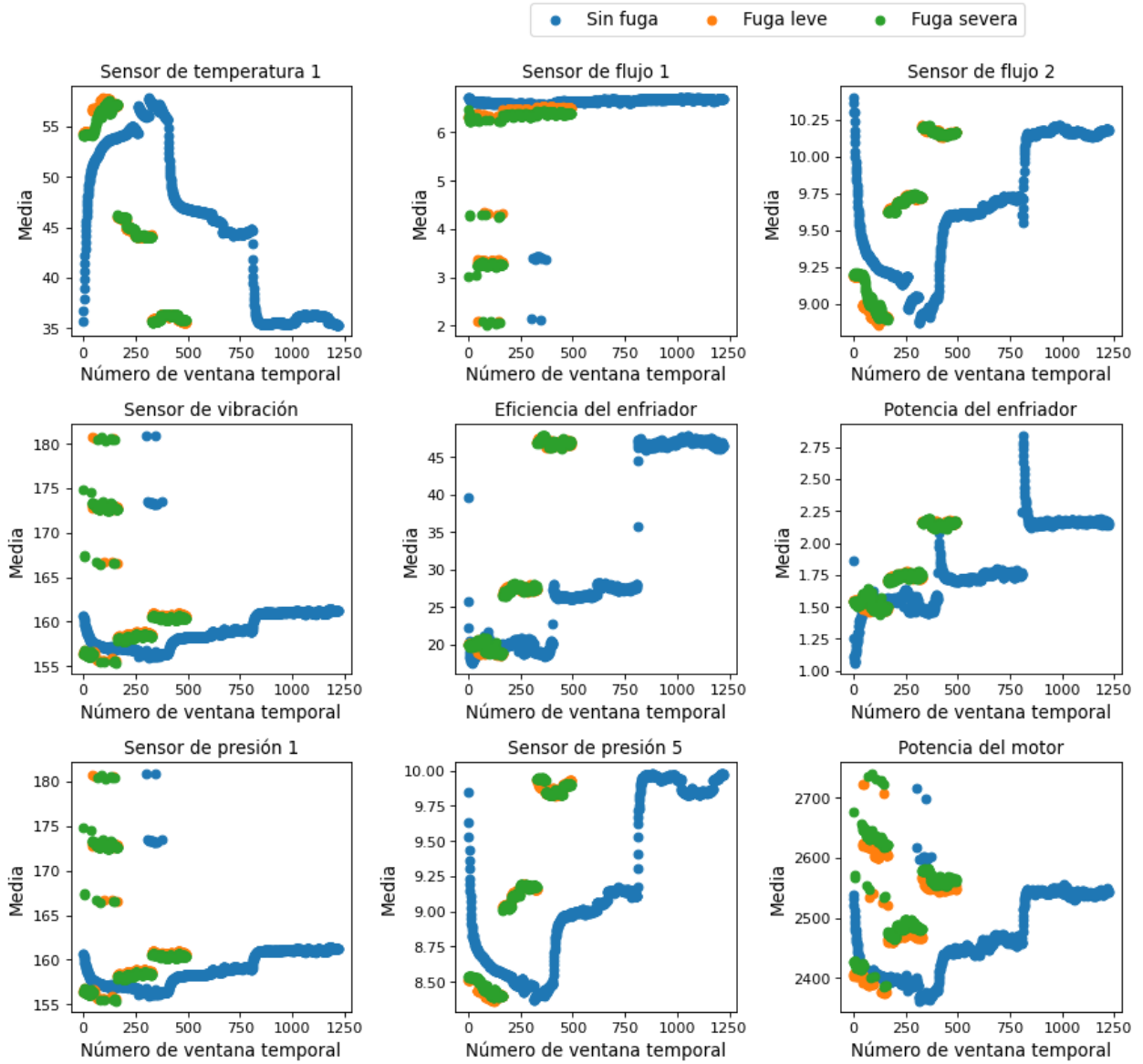


Figura A.127: Gráficos de la media obtenida según las clases del estado de la bomba.

Valor peak to peak para la clasificación: Fuga en la bomba,
usando 1 dato(s) por ciclo

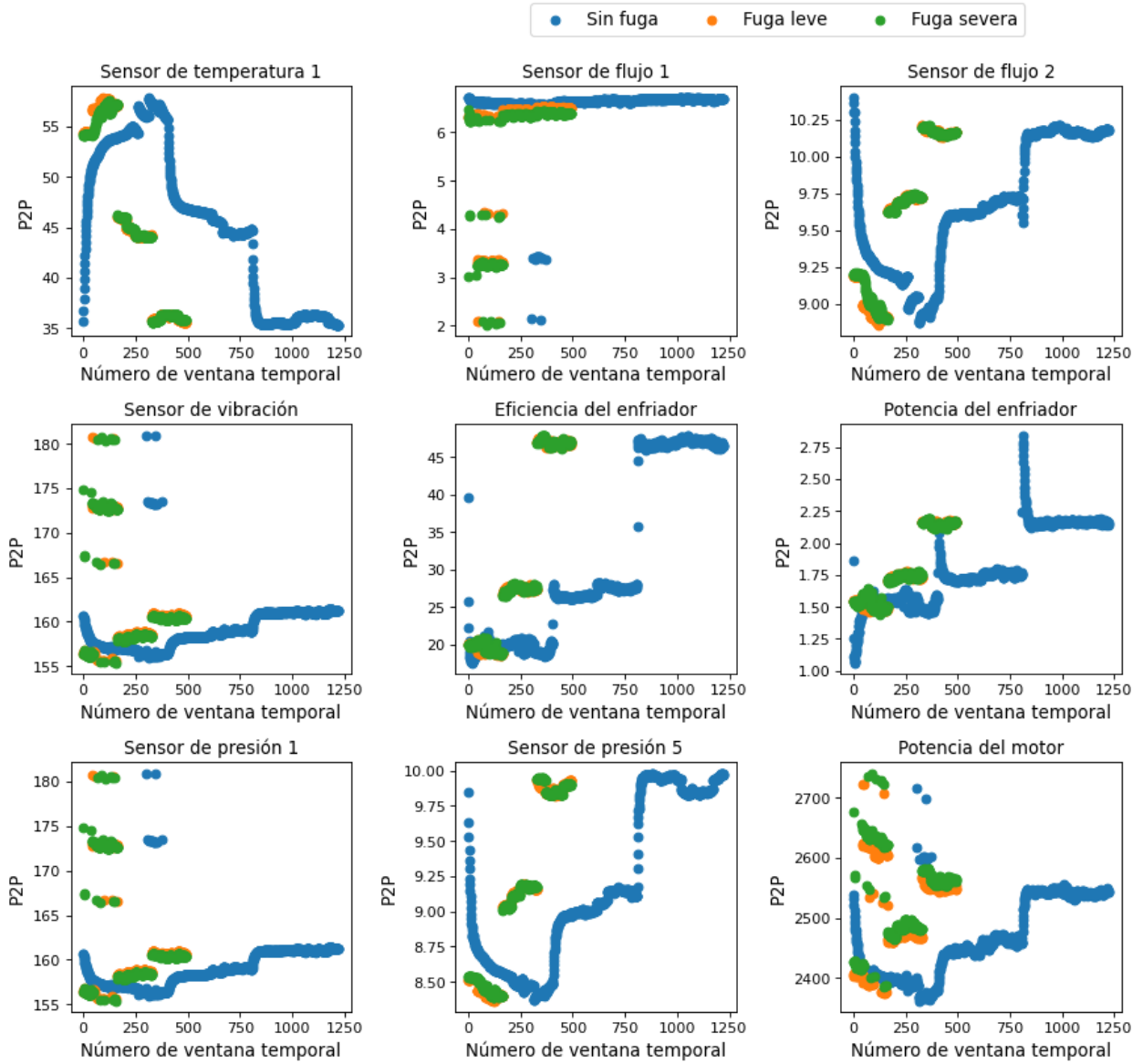


Figura A.128: Gráficos del valor peak to peak obtenido según las clases del estado de la bomba.

A.2.3.2. Gráficos usando 2 datos por ciclo

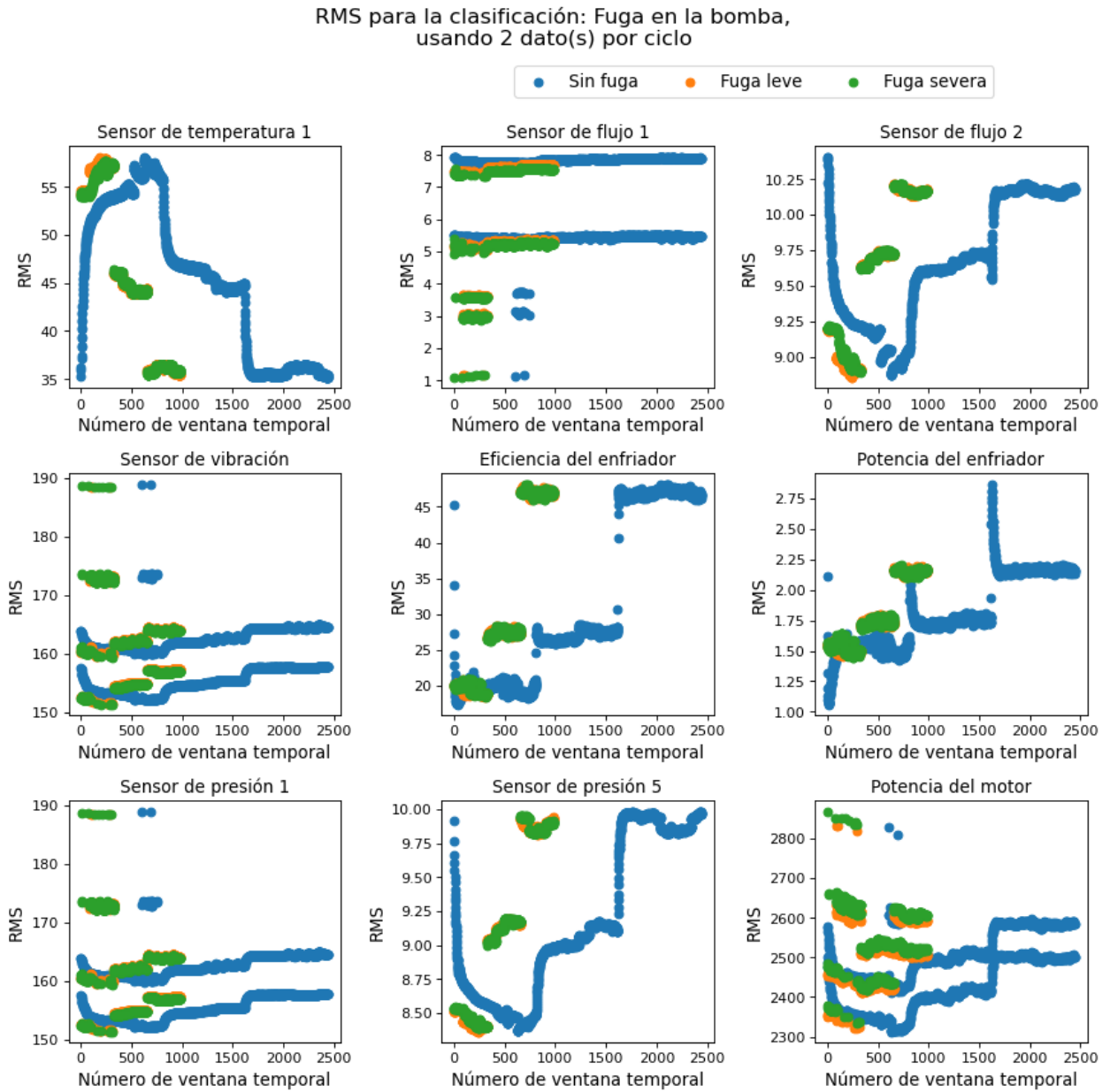


Figura A.129: Gráficos del RMS obtenido según las clases del estado de la bomba.

Varianza para la clasificación: Fuga en la bomba,
usando 2 dato(s) por ciclo

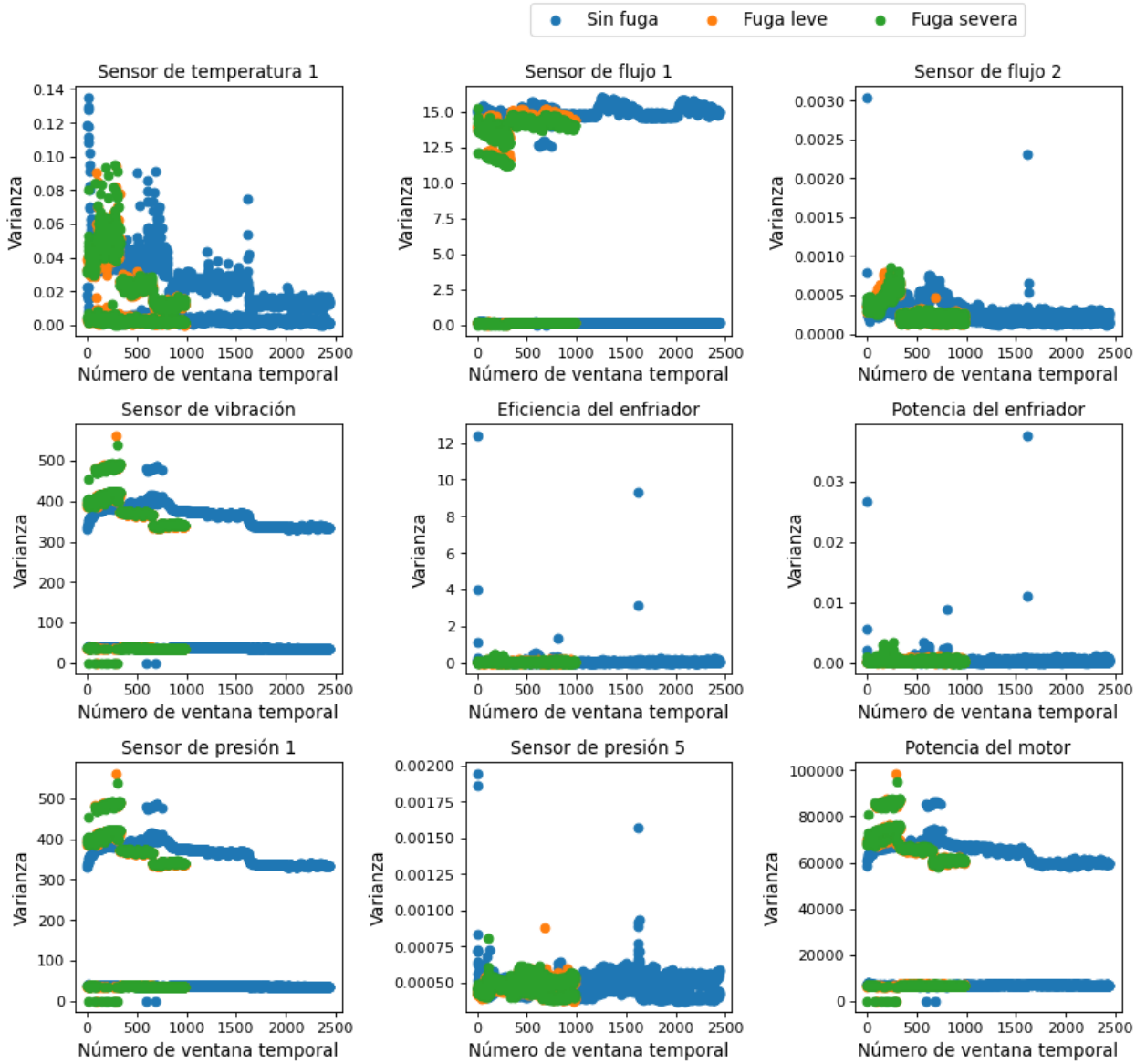


Figura A.130: Gráficos de la varianza obtenida según las clases del estado de la bomba.

Media para la clasificación: Fuga en la bomba,
usando 2 dato(s) por ciclo

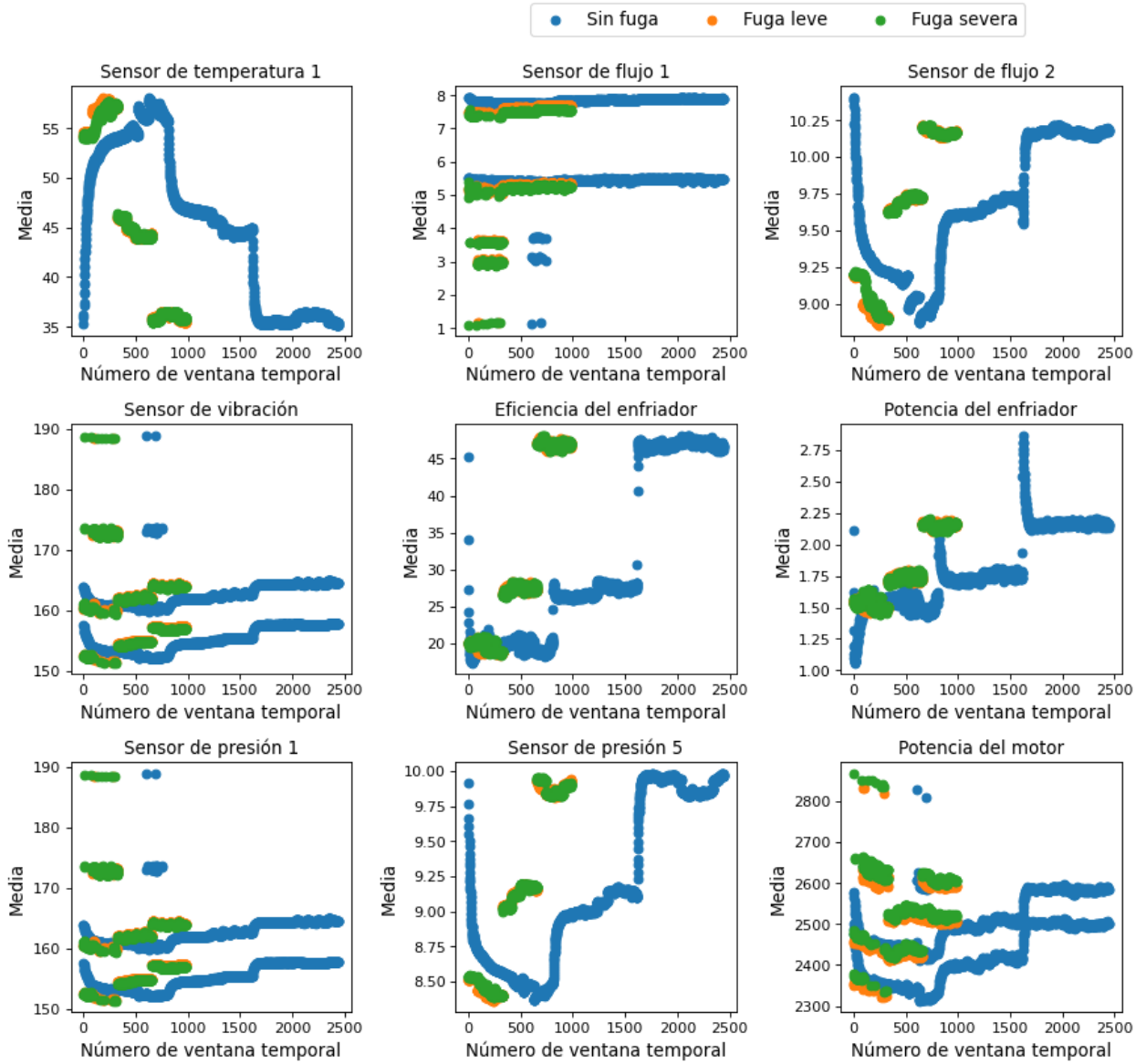


Figura A.131: Gráficos de la media obtenida según las clases del estado de la bomba.

Valor peak to peak para la clasificación: Fuga en la bomba,
usando 2 dato(s) por ciclo

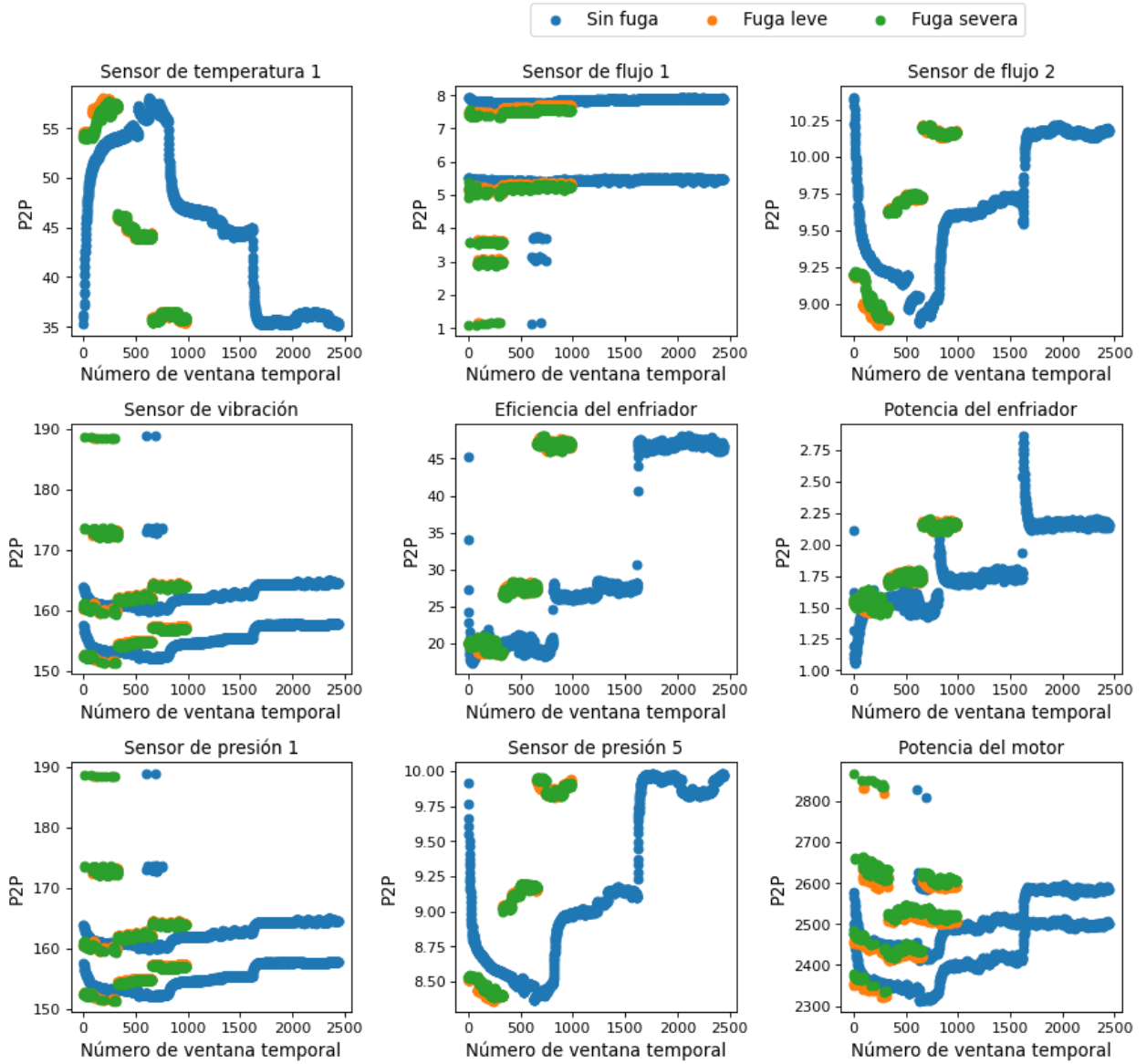


Figura A.132: Gráficos del valor peak to peak obtenido según las clases del estado de la bomba.

A.2.3.3. Gráficos usando 6 datos por ciclo

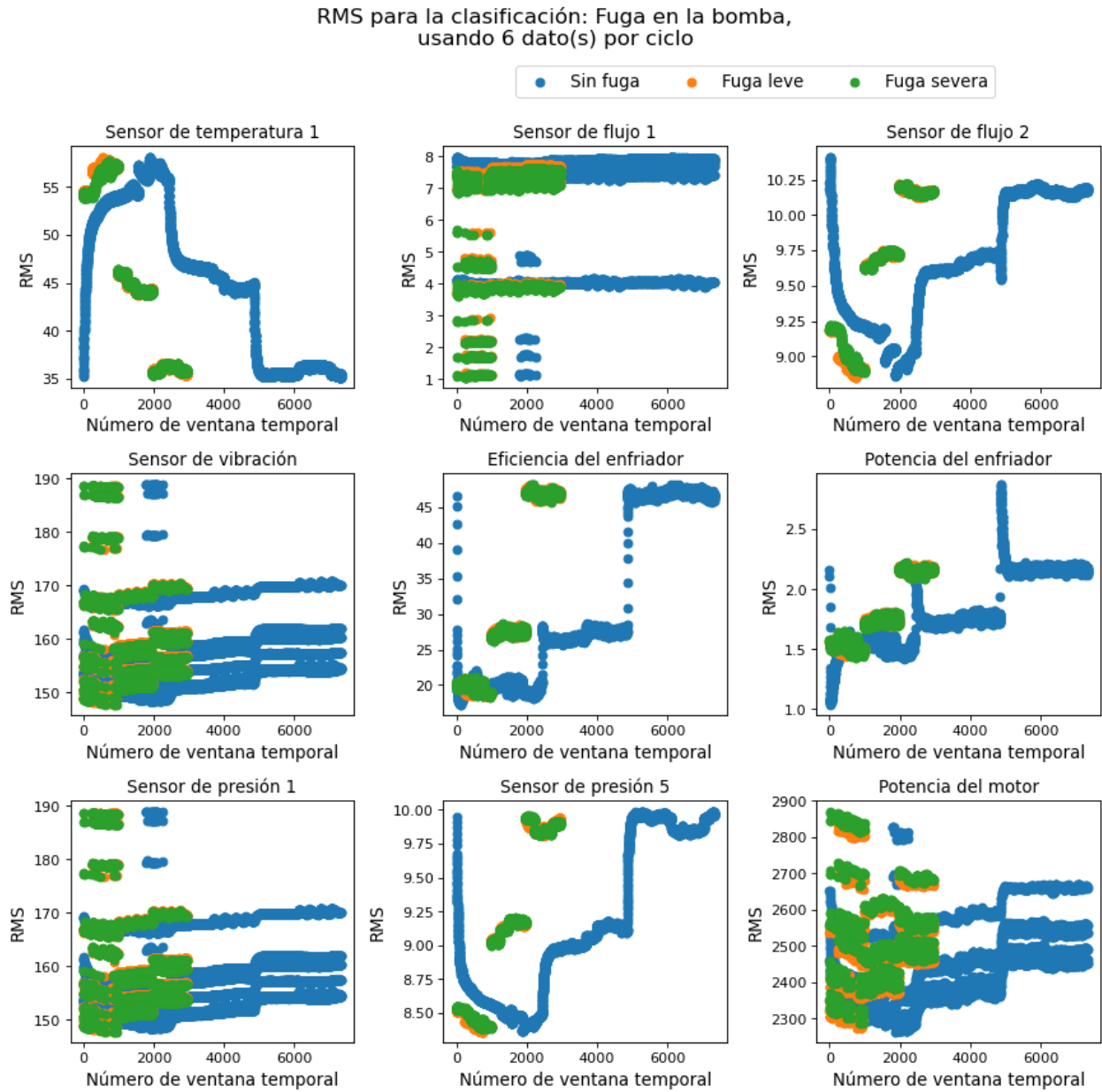


Figura A.133: Gráficos del RMS obtenido según las clases del estado de la bomba.

Varianza para la clasificación: Fuga en la bomba,
usando 6 dato(s) por ciclo

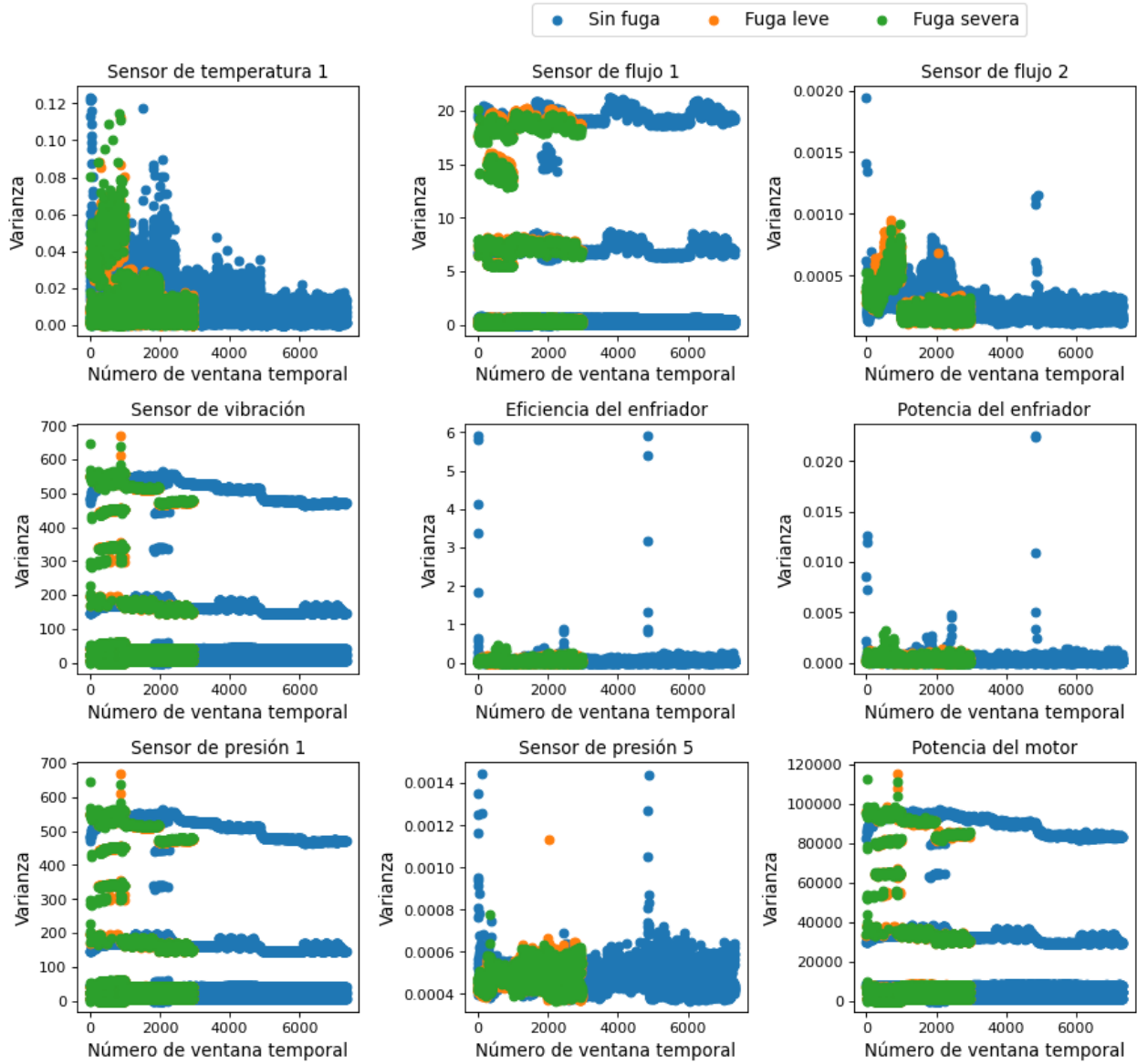


Figura A.134: Gráficos de la varianza obtenida según las clases del estado de la bomba.

Media para la clasificación: Fuga en la bomba,
usando 6 dato(s) por ciclo

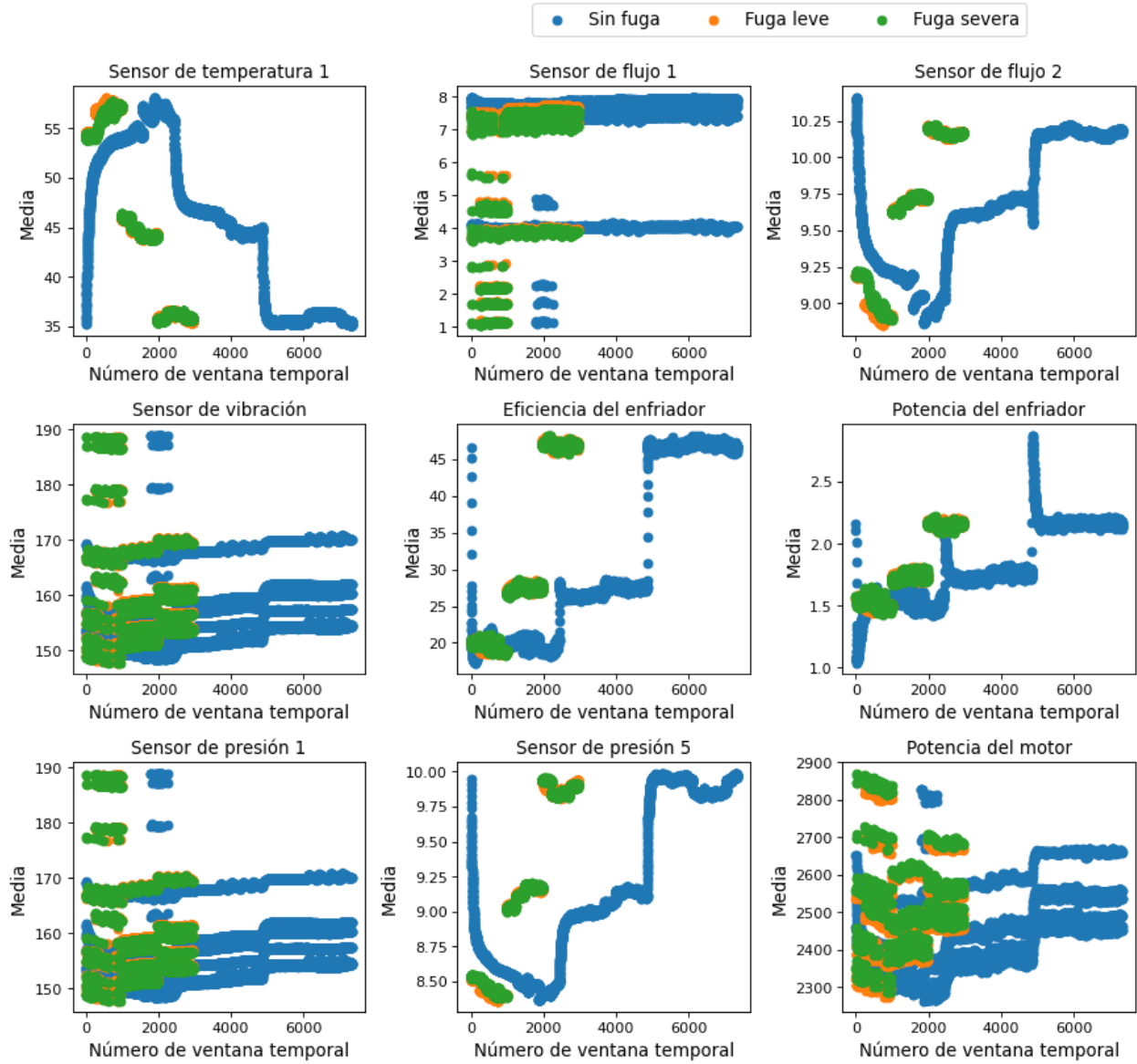


Figura A.135: Gráficos de la media obtenida según las clases del estado de la bomba.

Valor peak to peak para la clasificación: Fuga en la bomba, usando 6 dato(s) por ciclo

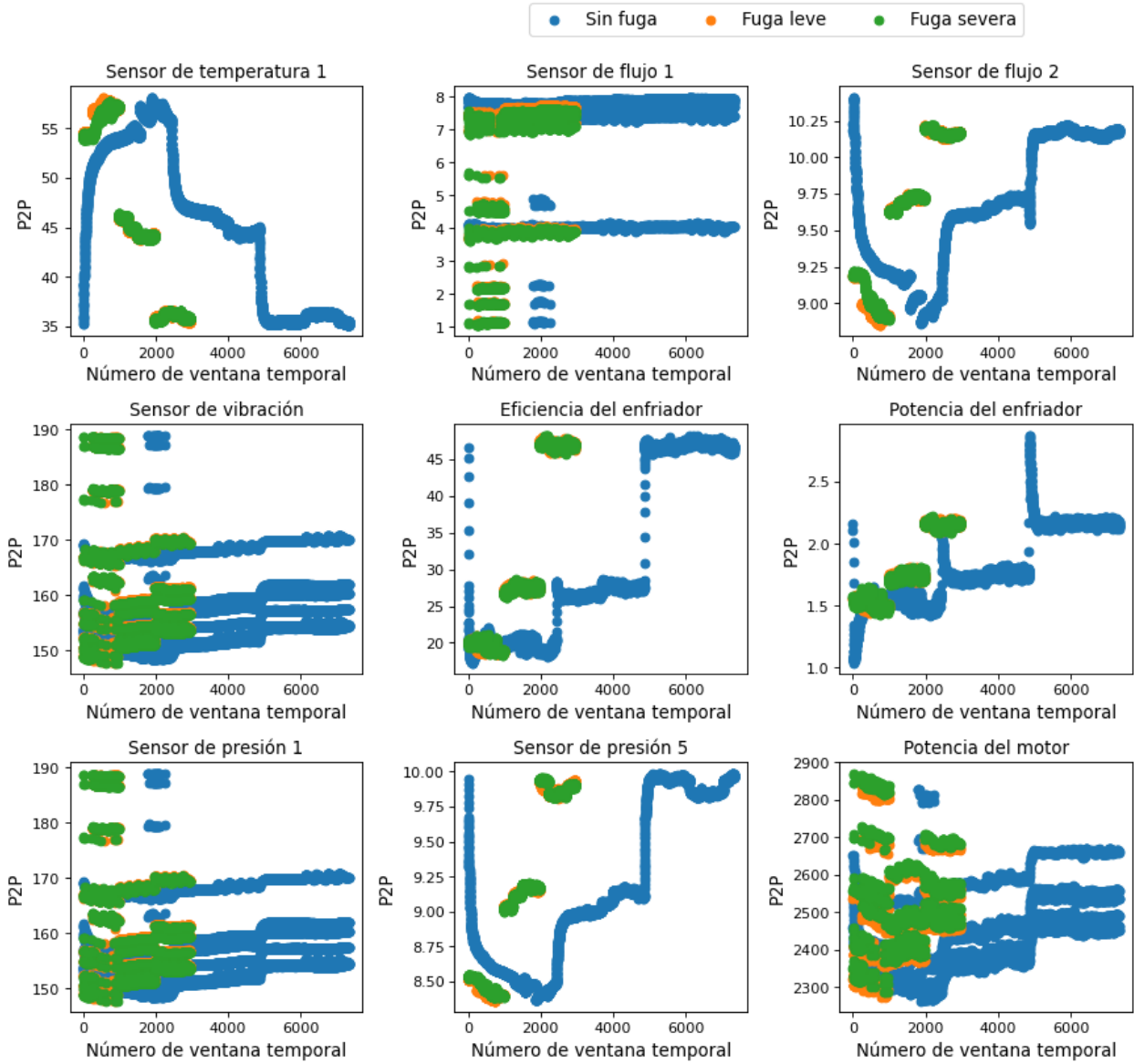


Figura A.136: Gráficos del valor peak to peak obtenido según las clases del estado de la bomba.

A.2.4. Gráficos del acumulador

A.2.4.1. Gráficos usando 1 dato por ciclo

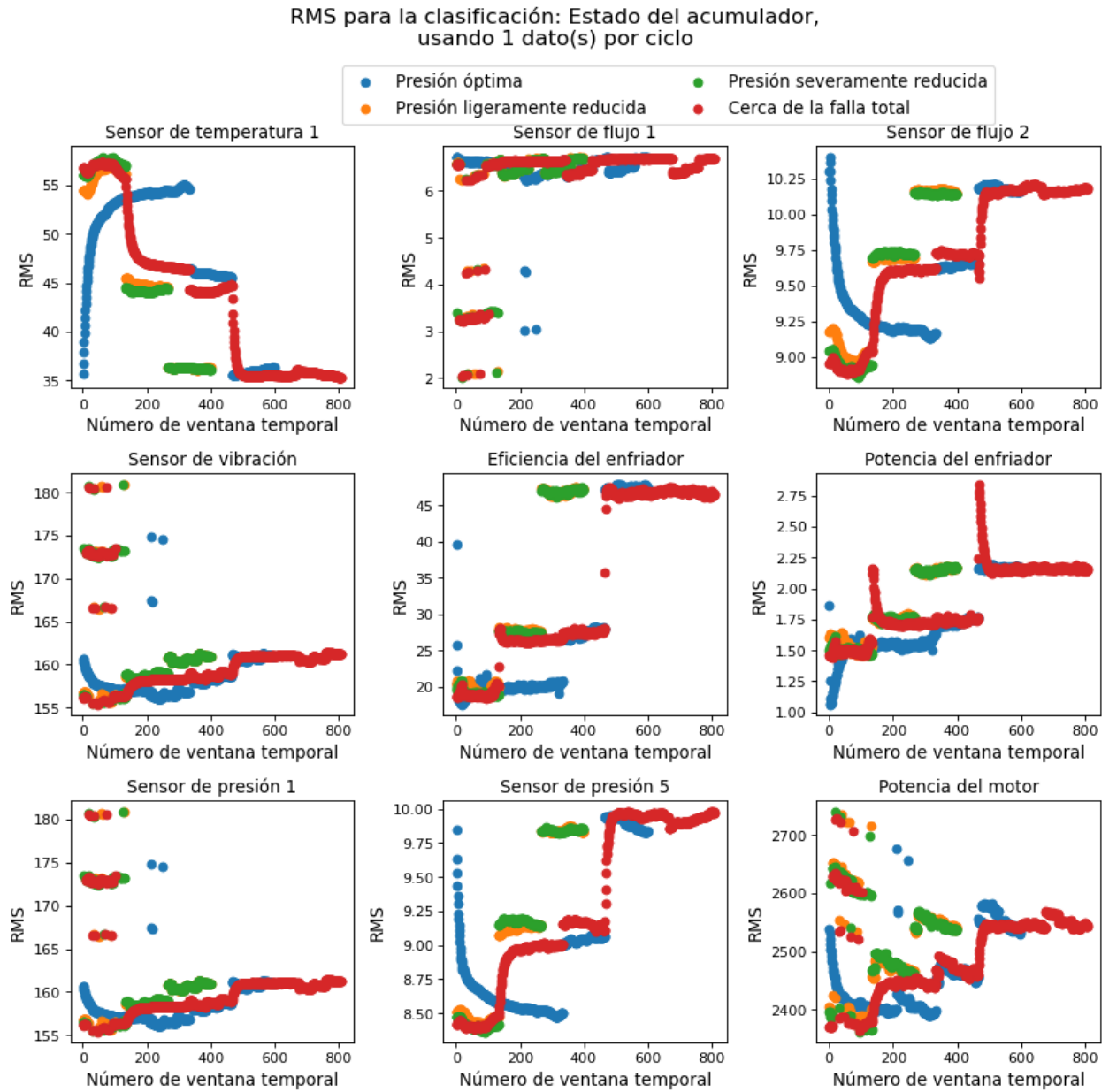


Figura A.137: Gráficos del RMS obtenido según las clases del estado del acumulador.

Varianza para la clasificación: Estado del acumulador, usando 1 dato(s) por ciclo

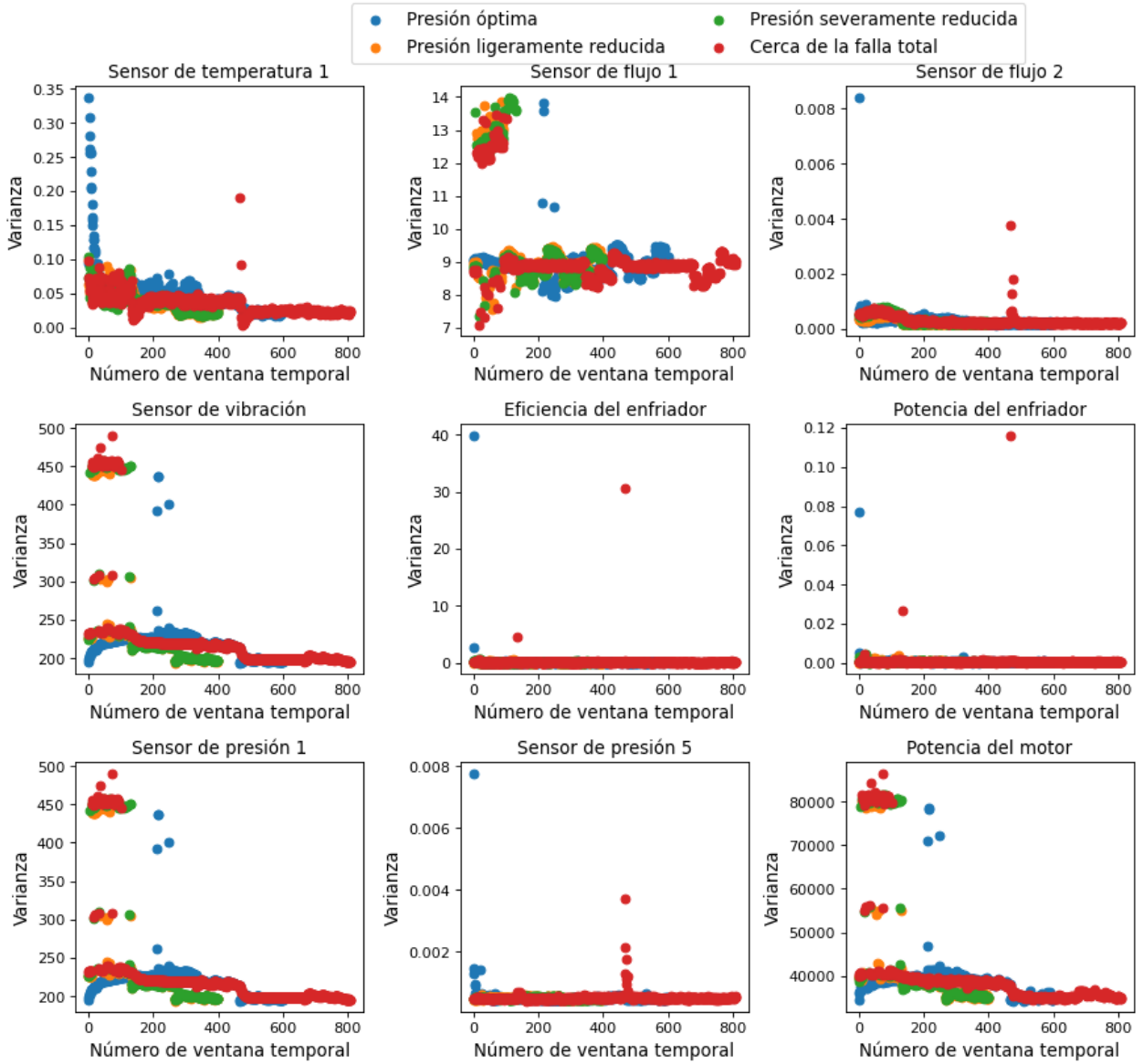


Figura A.138: Gráficos de la varianza obtenida según las clases del estado del acumulador.

Media para la clasificación: Estado del acumulador,
usando 1 dato(s) por ciclo

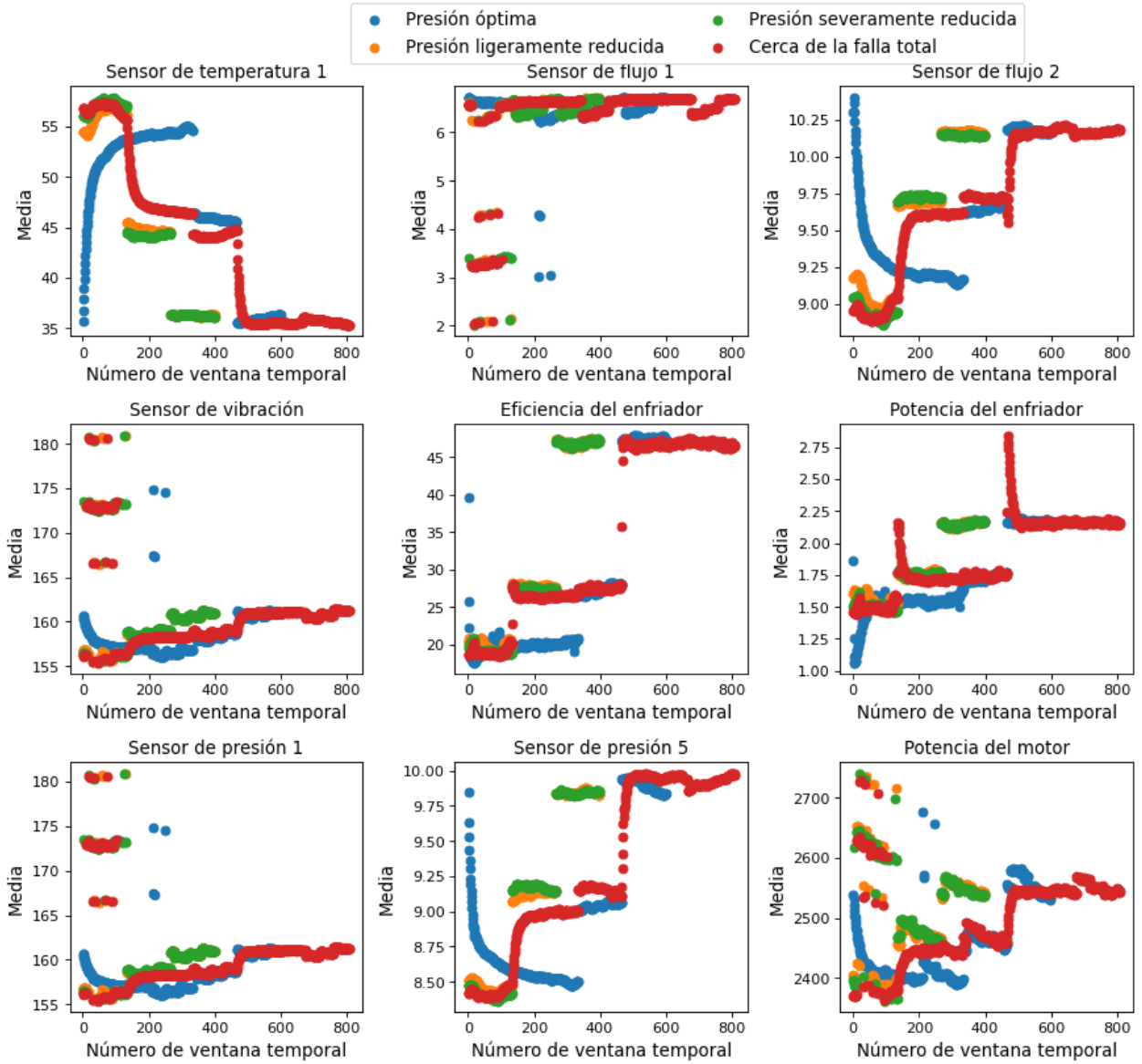


Figura A.139: Gráficos de la media obtenida según las clases del estado del acumulador.

Valor peak to peak para la clasificación: Estado del acumulador,
usando 1 dato(s) por ciclo

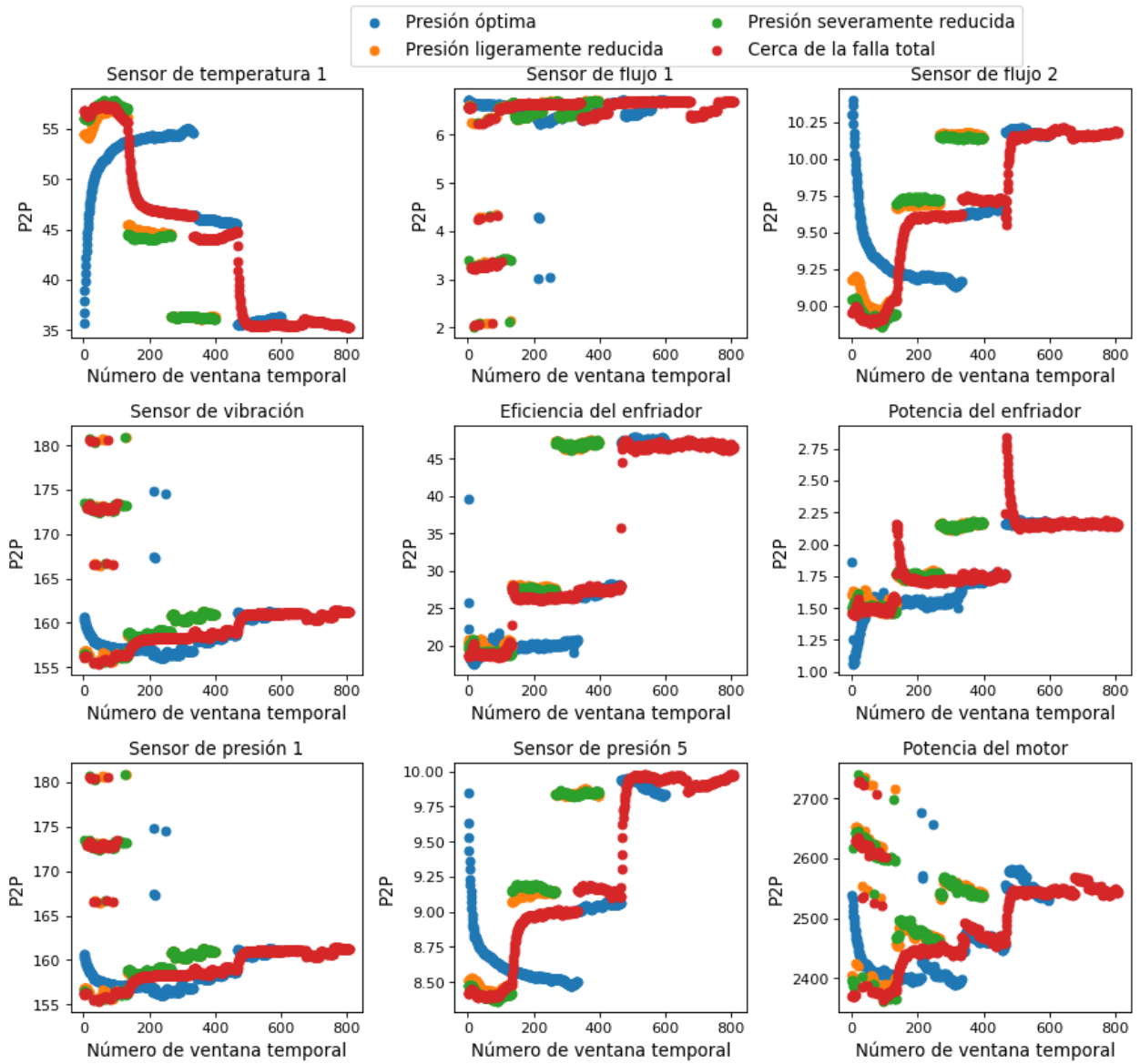


Figura A.140: Gráficos del valor peak to peak obtenido según las clases del estado del acumulador.

A.2.4.2. Gráficos usando 2 datos por ciclo

RMS para la clasificación: Estado del acumulador, usando 2 dato(s) por ciclo

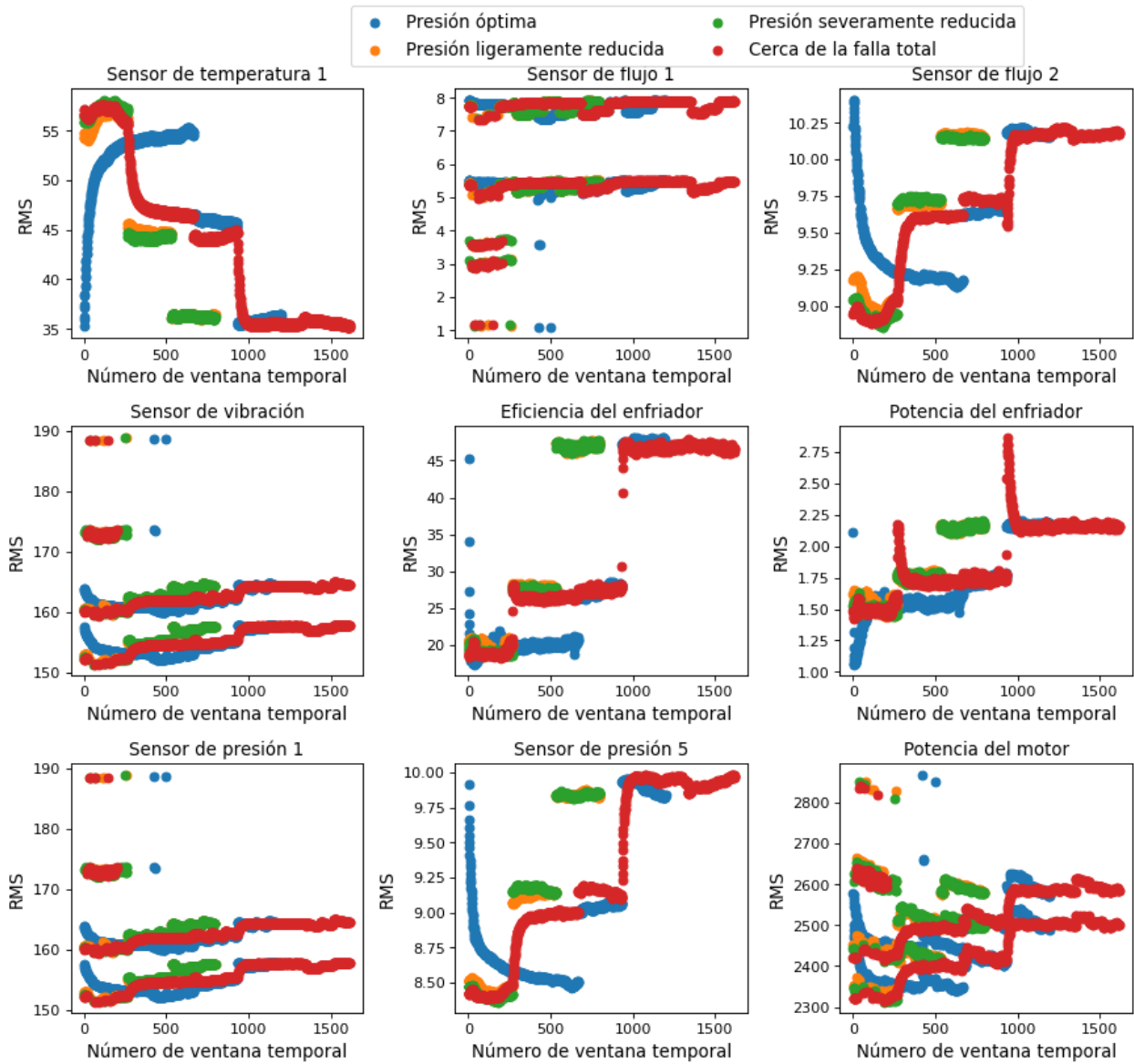


Figura A.141: Gráficos del RMS obtenido según las clases del estado del acumulador.

Varianza para la clasificación: Estado del acumulador, usando 2 dato(s) por ciclo

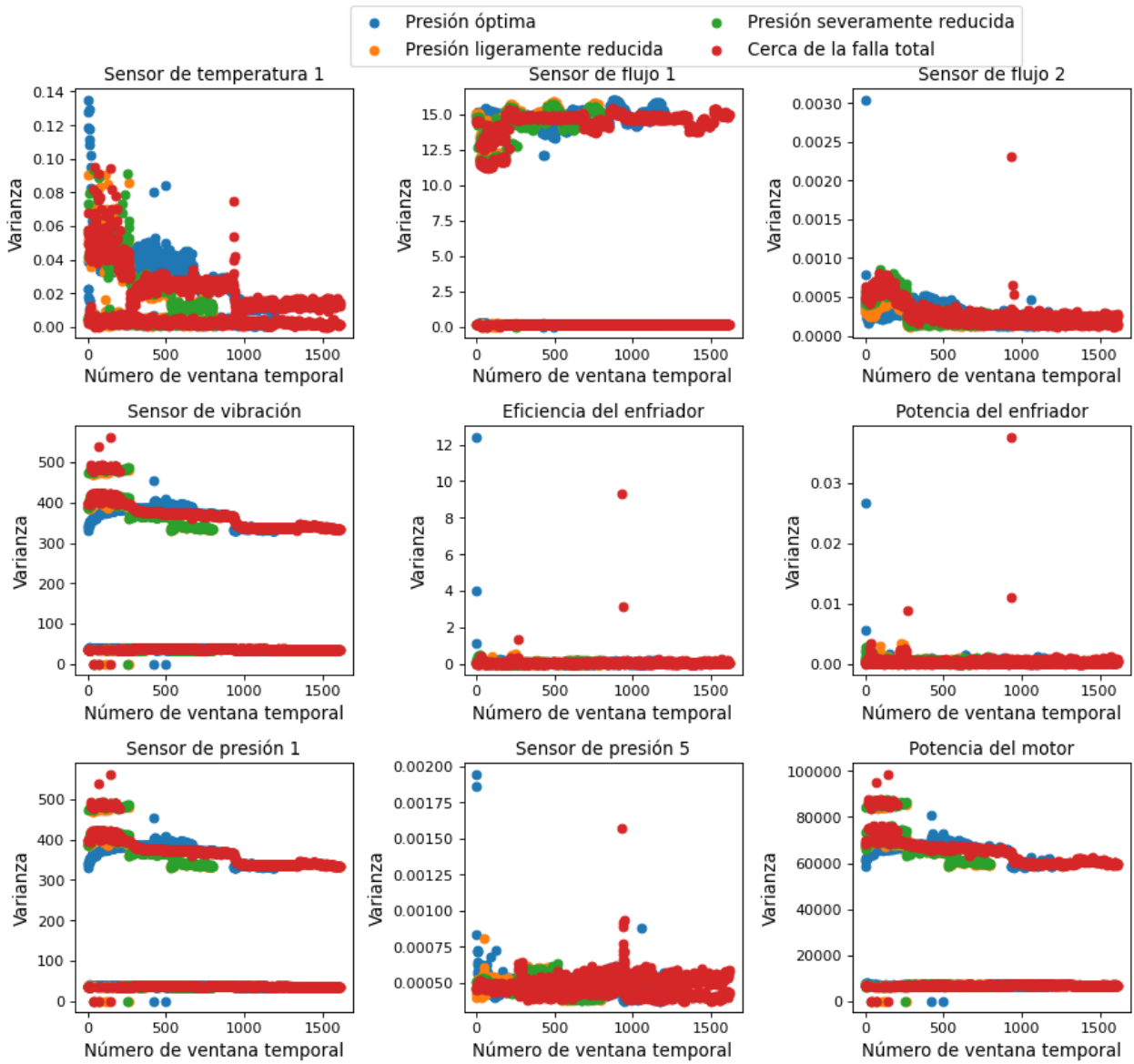


Figura A.142: Gráficos de la varianza obtenida según las clases del estado del acumulador.

Media para la clasificación: Estado del acumulador,
usando 2 dato(s) por ciclo

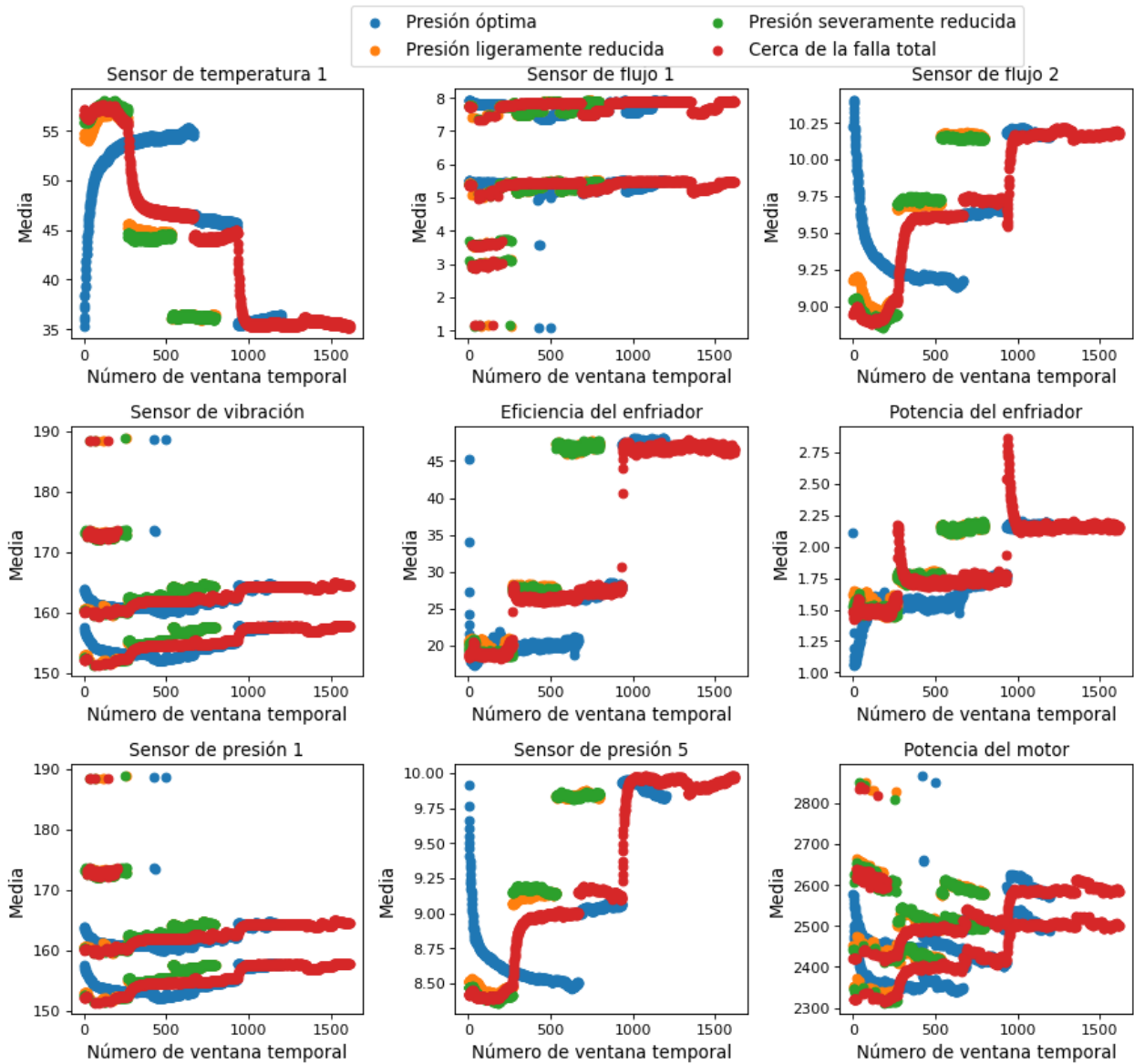


Figura A.143: Gráficos de la media obtenida según las clases del estado del acumulador.

Valor peak to peak para la clasificación: Estado del acumulador, usando 2 dato(s) por ciclo

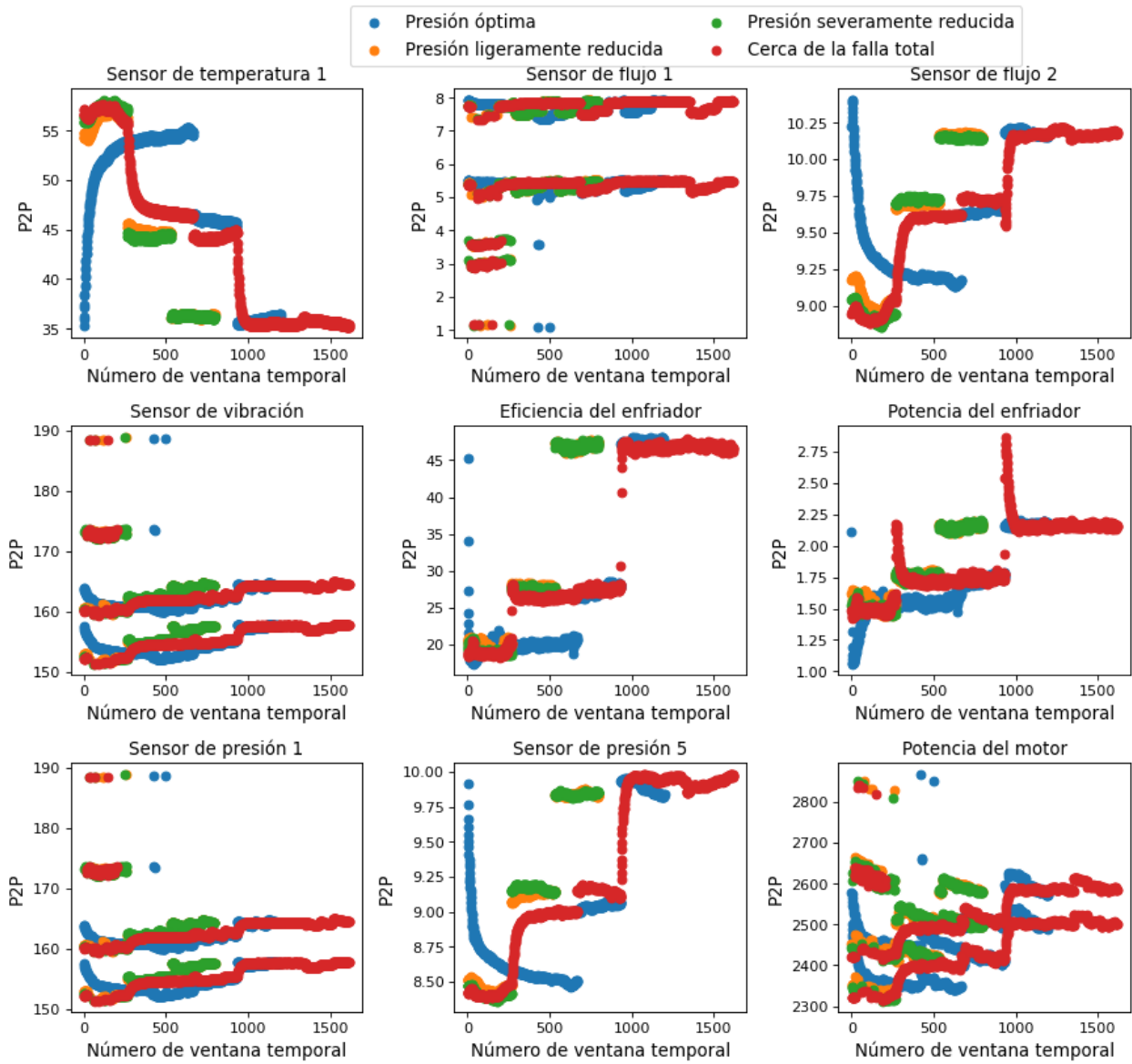


Figura A.144: Gráficos del valor peak to peak obtenido según las clases del estado del acumulador.

A.2.4.3. Gráficos usando 6 datos por ciclo

RMS para la clasificación: Estado del acumulador, usando 6 dato(s) por ciclo

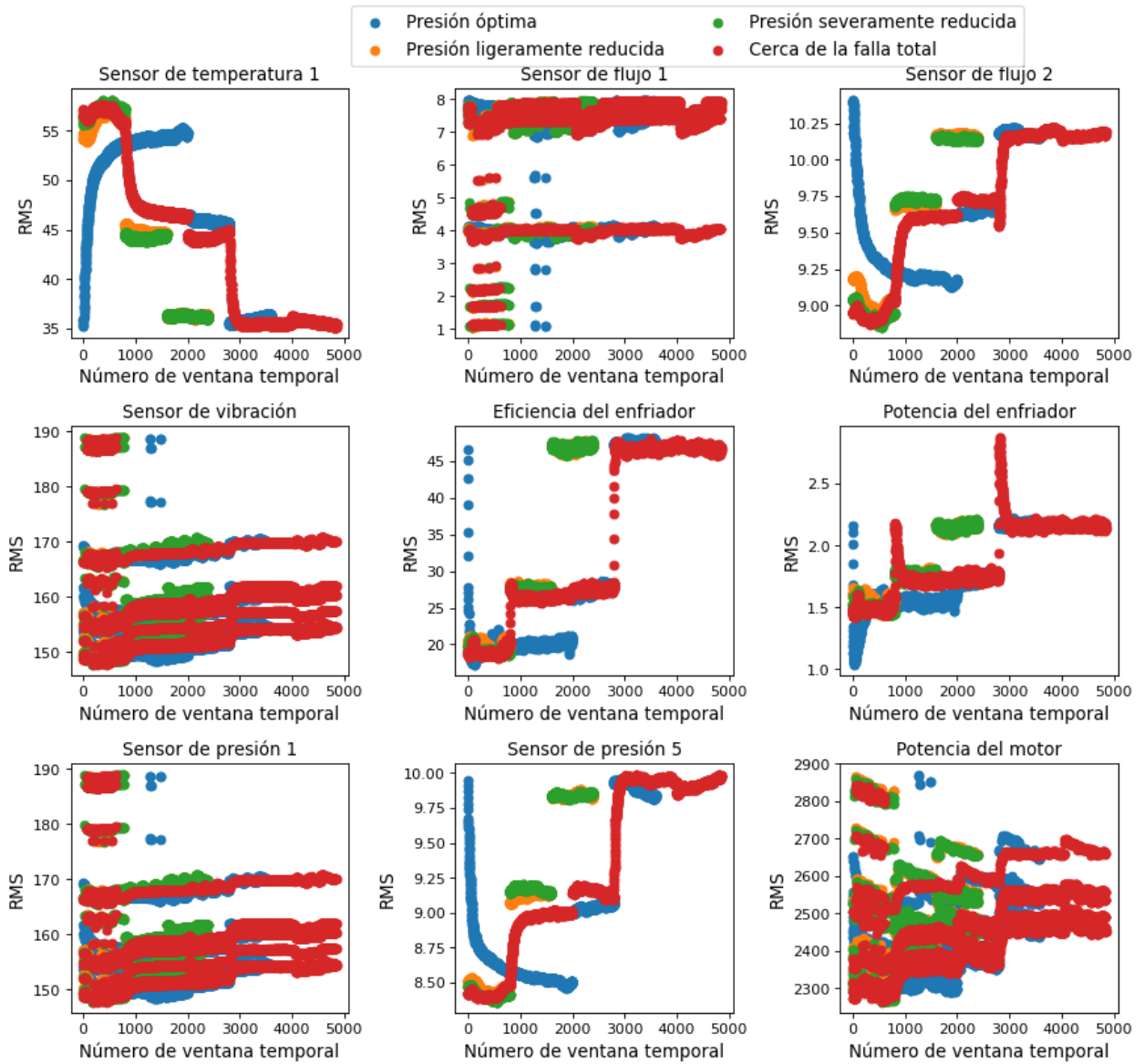


Figura A.145: Gráficos del RMS obtenido según las clases del estado del acumulador.

Varianza para la clasificación: Estado del acumulador, usando 6 dato(s) por ciclo

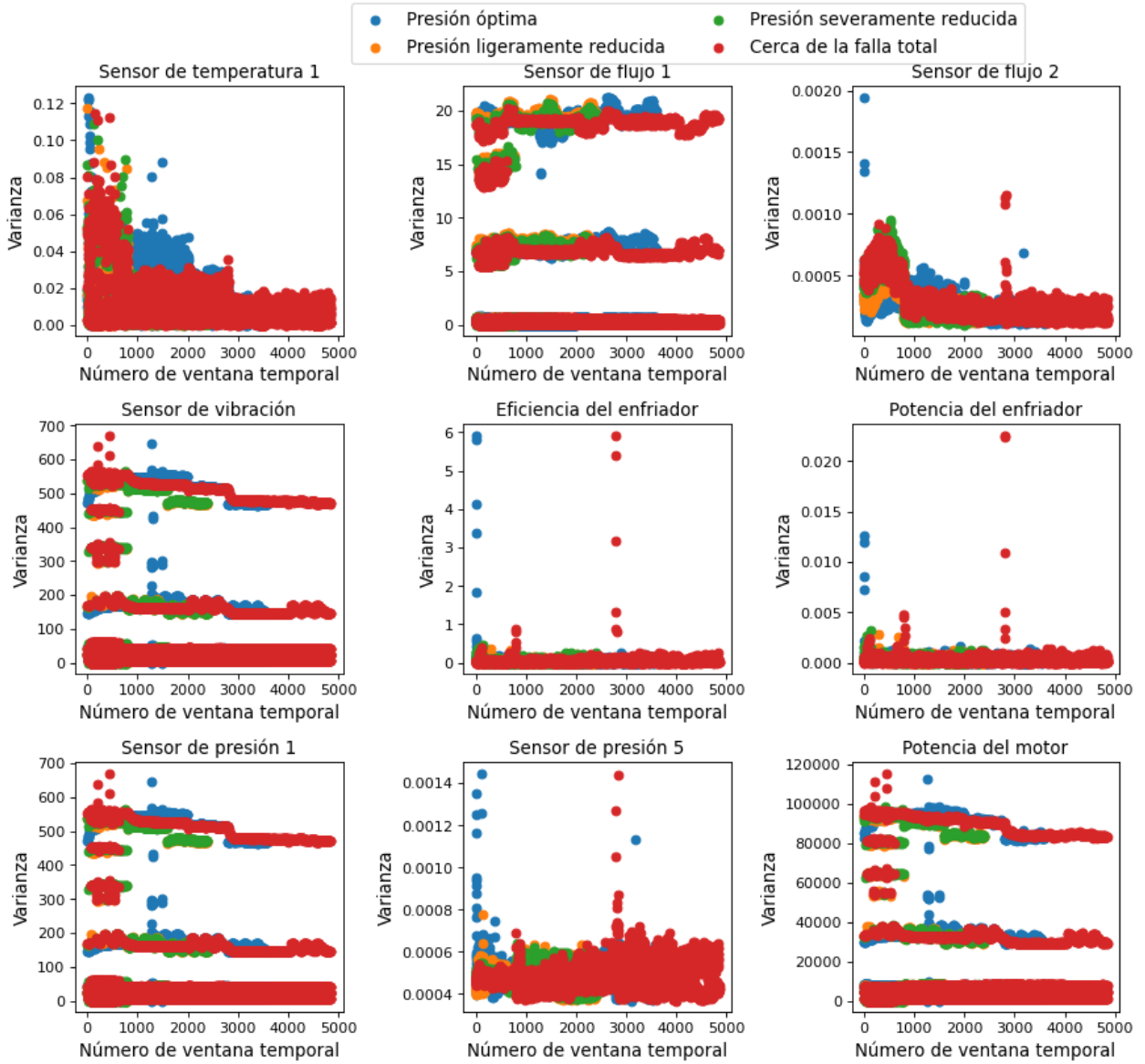


Figura A.146: Gráficos de la varianza obtenida según las clases del estado del acumulador.

Media para la clasificación: Estado del acumulador,
usando 6 dato(s) por ciclo

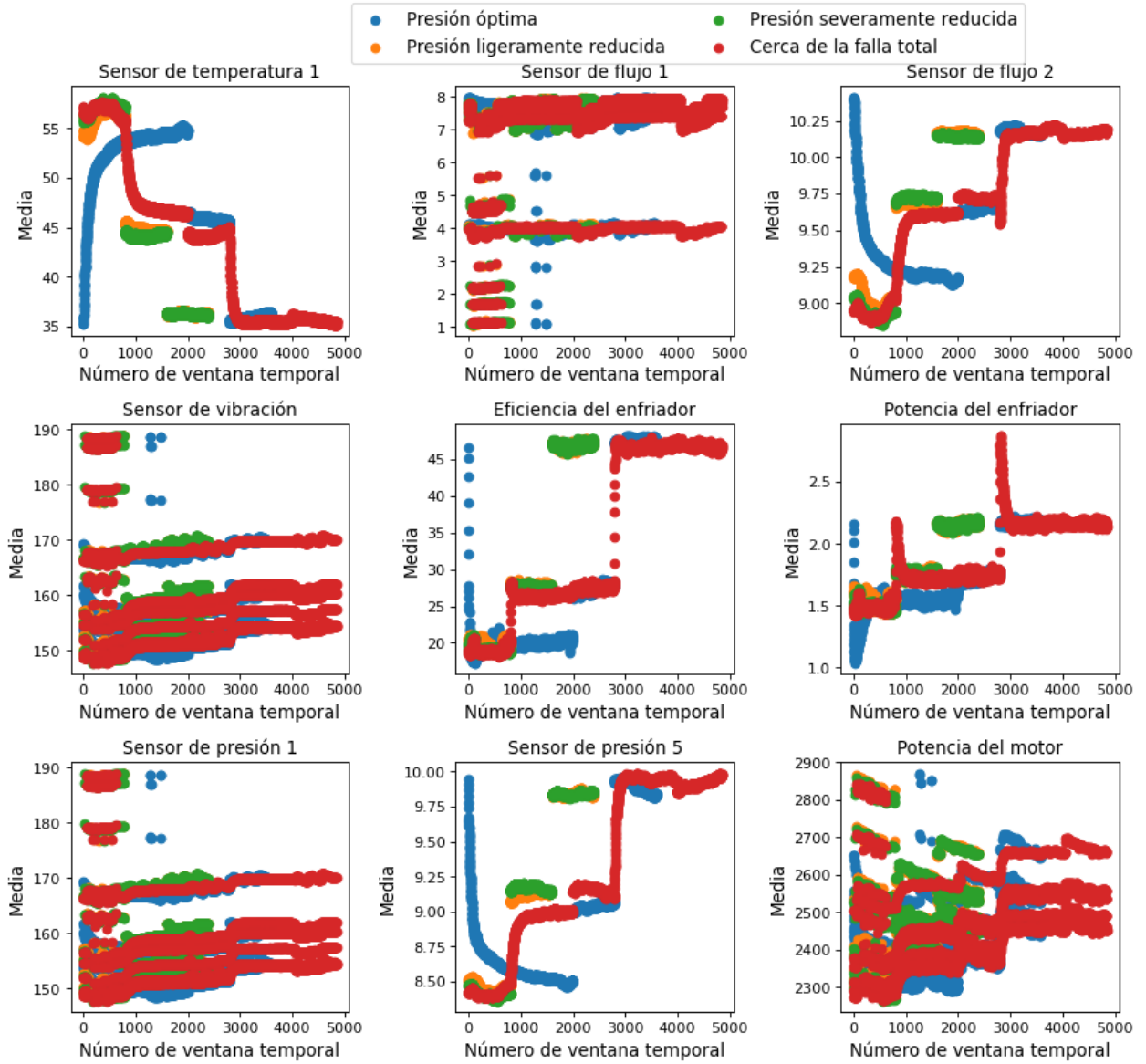


Figura A.147: Gráficos de la media obtenida según las clases del estado del acumulador.

Valor peak to peak para la clasificación: Estado del acumulador, usando 6 dato(s) por ciclo

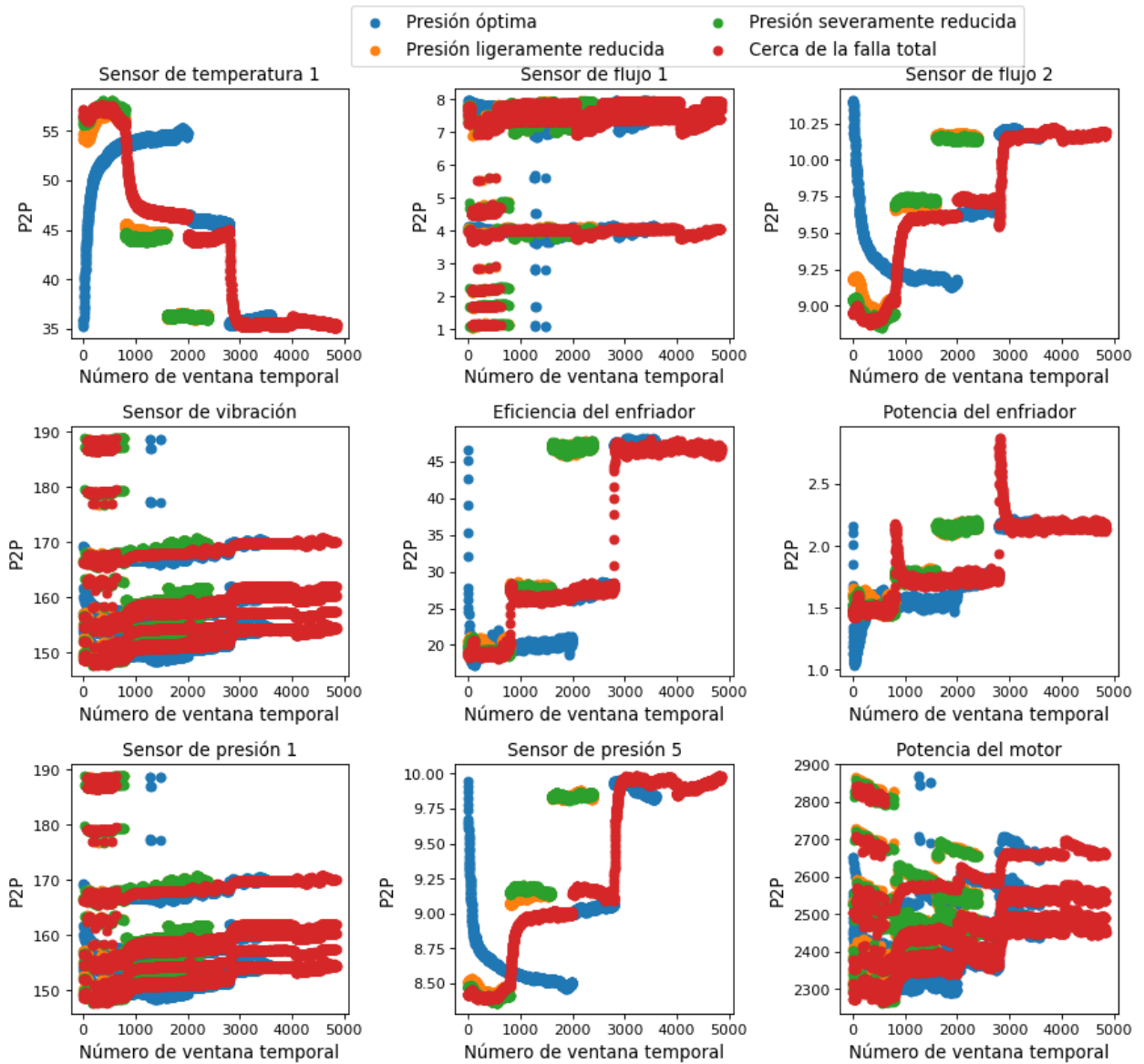


Figura A.148: Gráficos del valor peak to peak obtenido según las clases del estado del acumulador.

A.2.5. Gráficos de la estabilidad del sistema

A.2.5.1. Gráficos usando 1 dato por ciclo

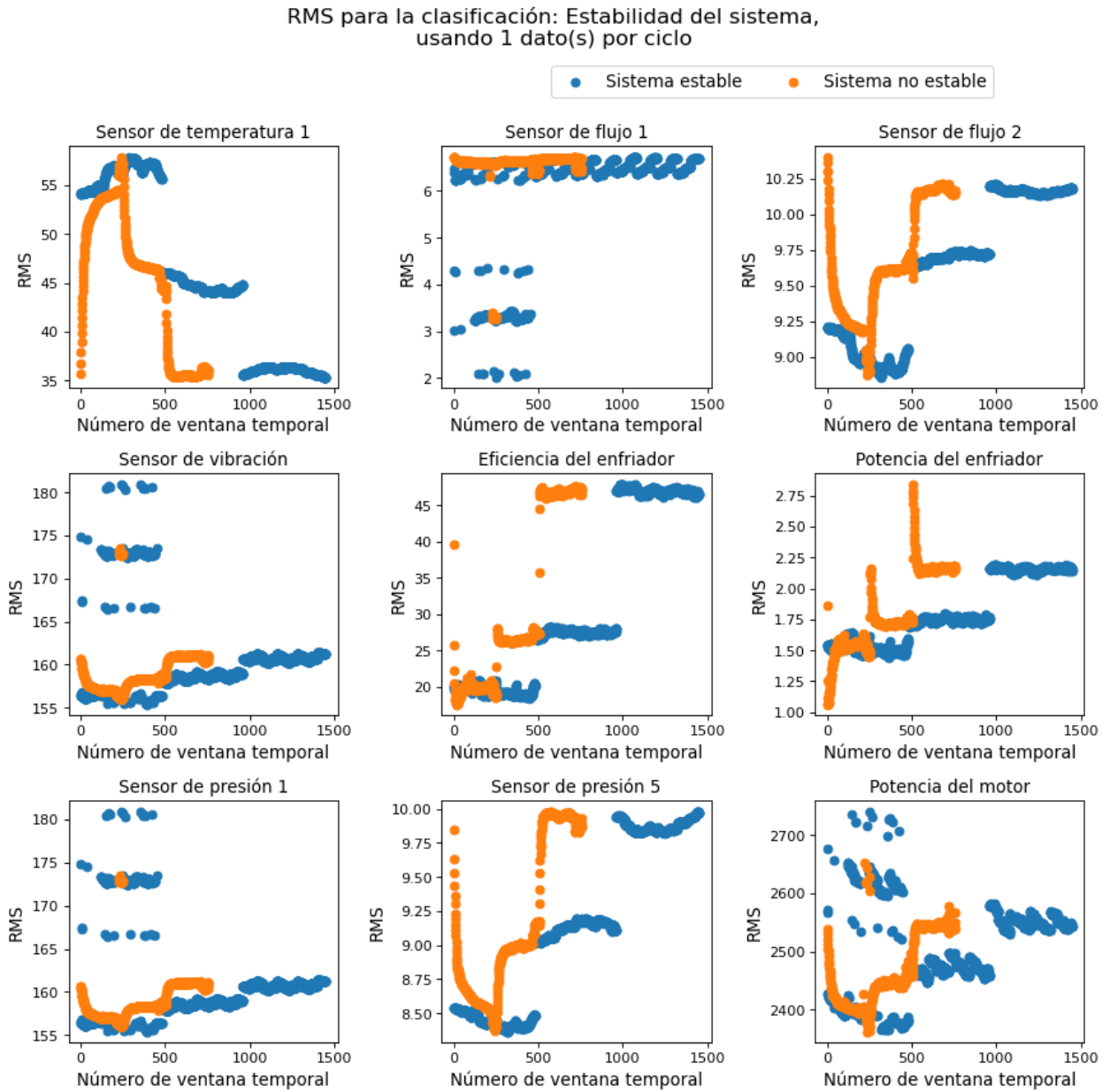


Figura A.149: Gráficos del RMS obtenido según las clases de la estabilidad del sistema.

Varianza para la clasificación: Estabilidad del sistema,
usando 1 dato(s) por ciclo

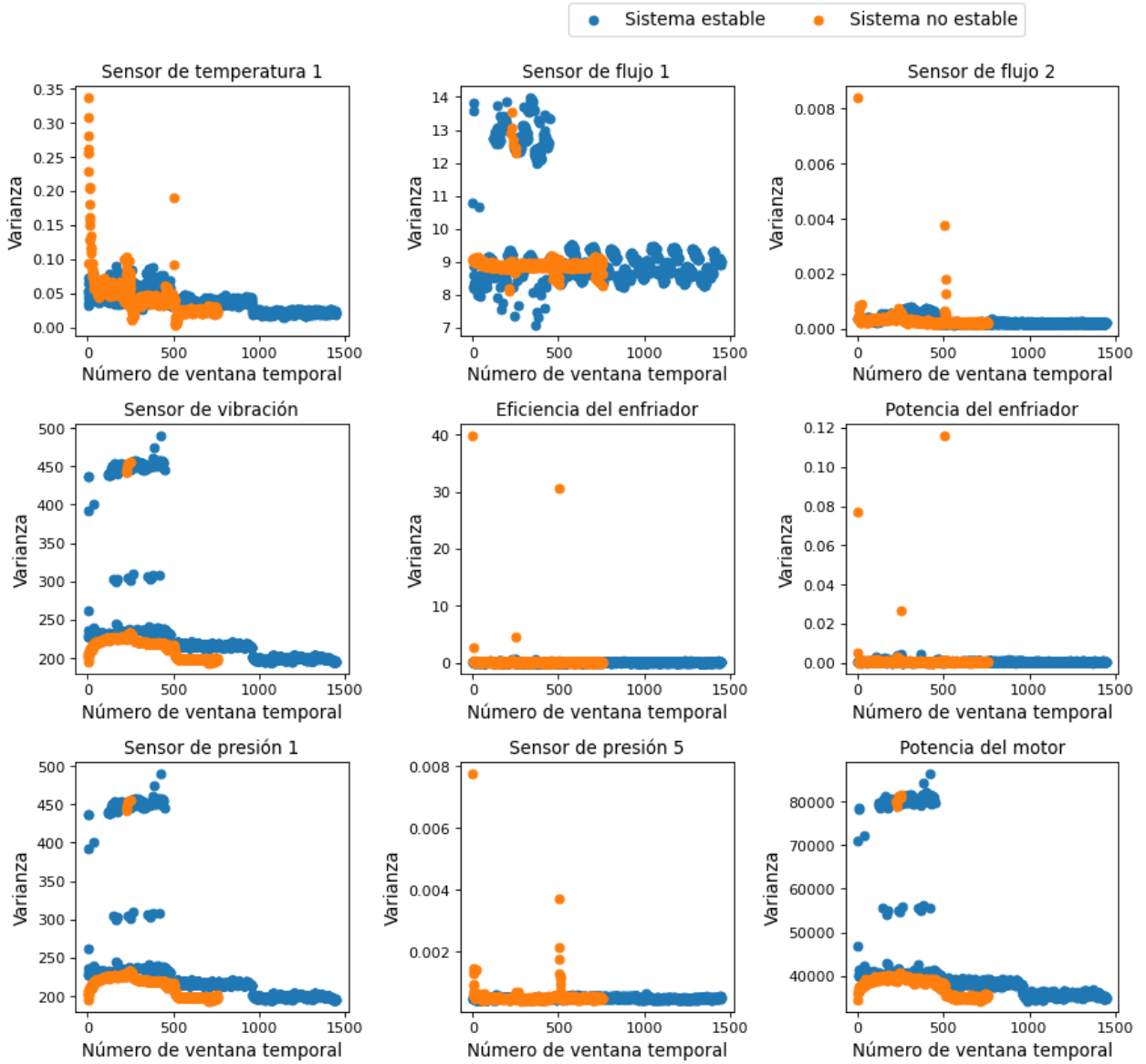


Figura A.150: Gráficos de la varianza obtenida según las clases de la estabilidad del sistema.

Media para la clasificación: Estabilidad del sistema,
usando 1 dato(s) por ciclo

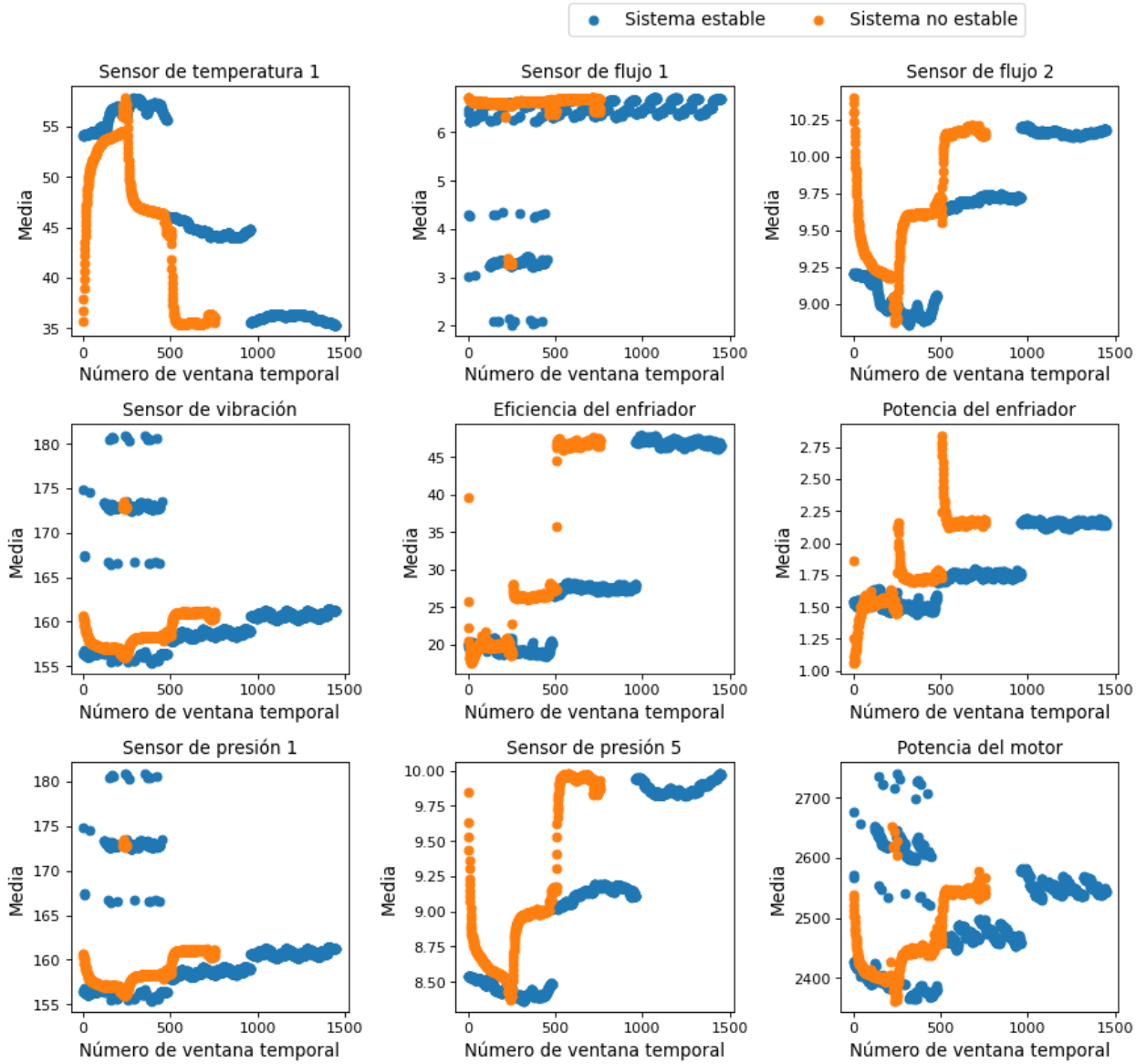


Figura A.151: Gráficos de la media obtenida según las clases de la estabilidad del sistema.

Valor peak to peak para la clasificación: Estabilidad del sistema, usando 1 dato(s) por ciclo

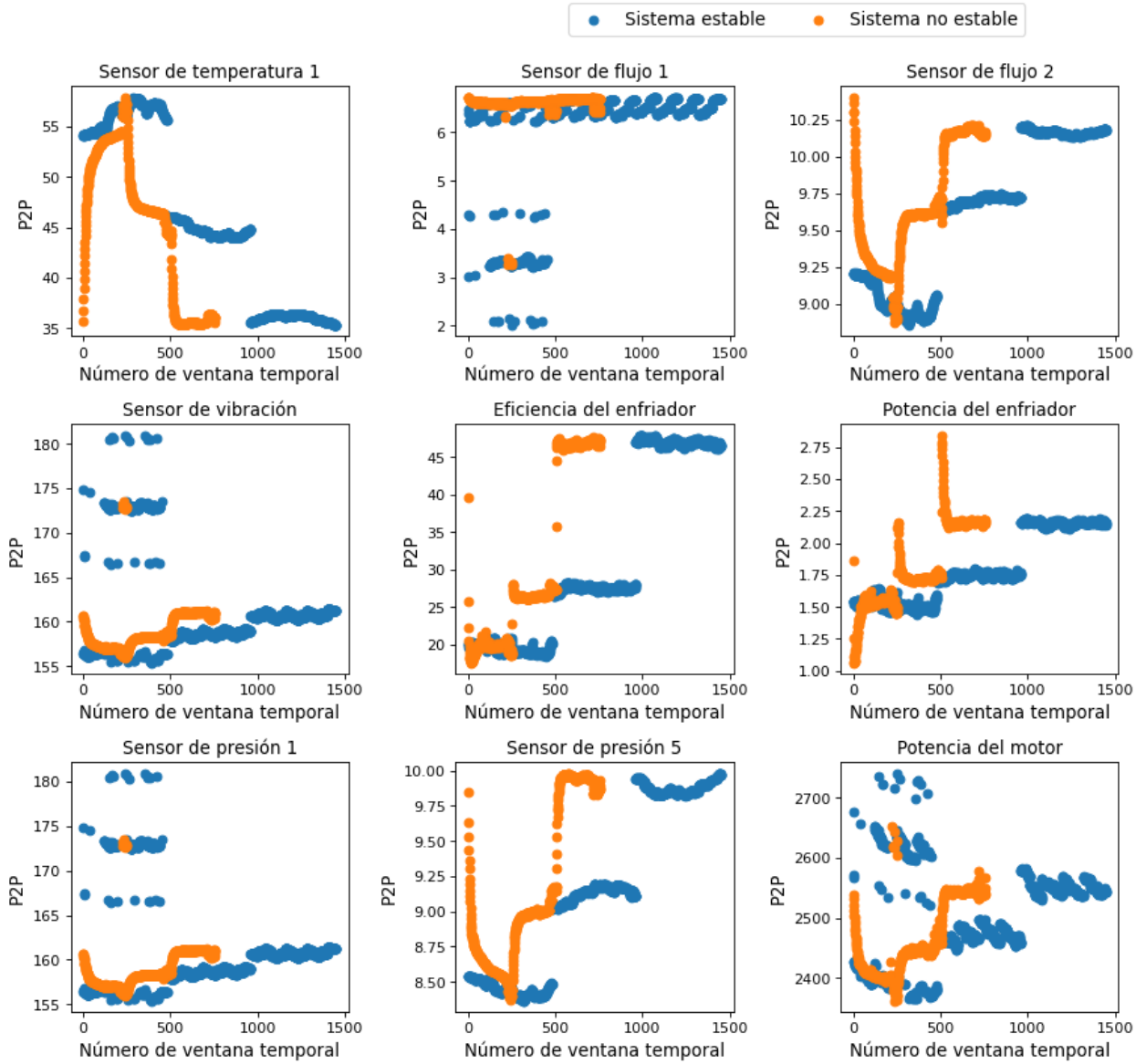


Figura A.152: Gráficos del valor peak to peak obtenido según las clases de la estabilidad del sistema.

A.2.5.2. Gráficos usando 2 datos por ciclo

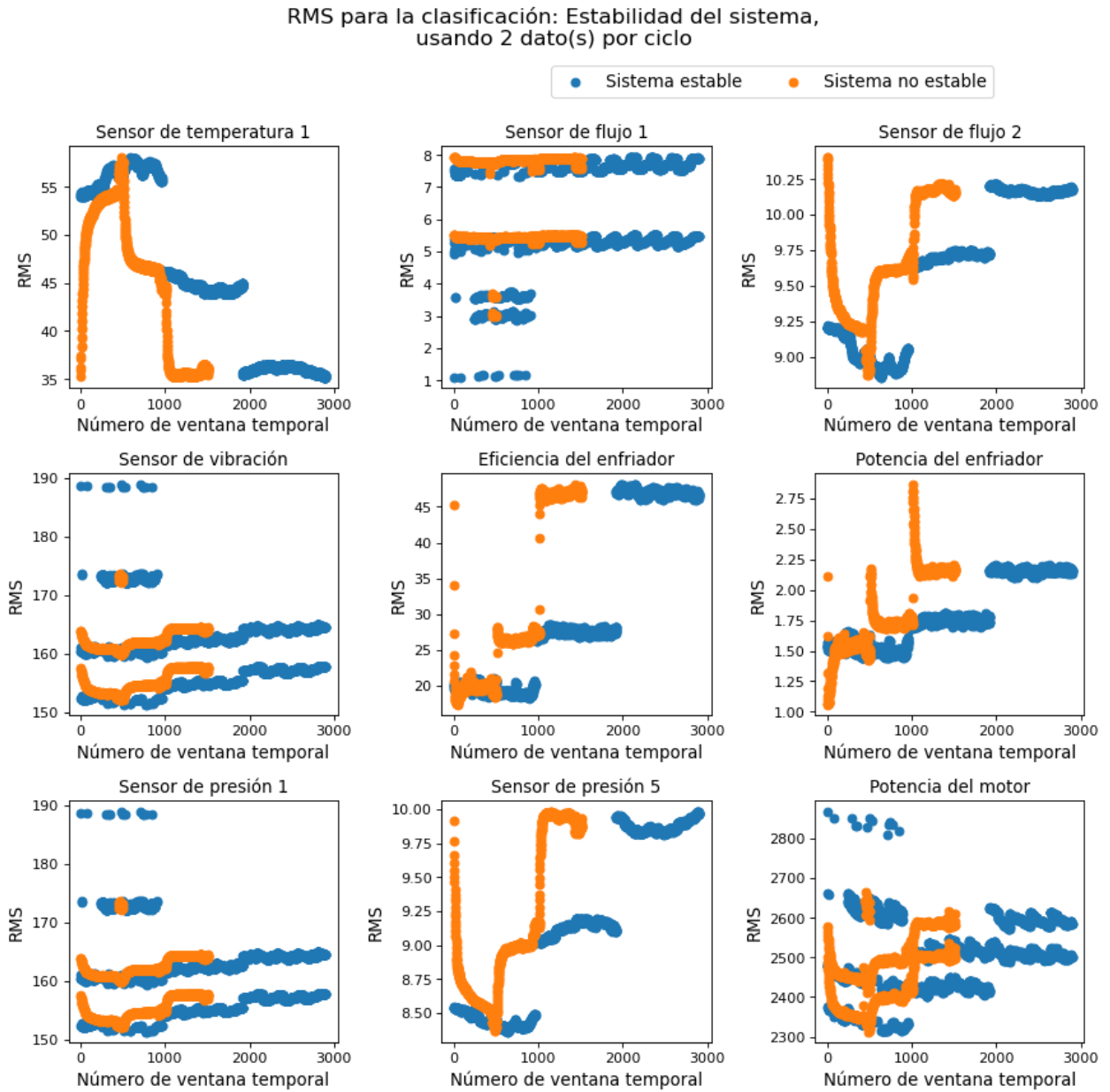


Figura A.153: Gráficos del RMS obtenido según las clases de la estabilidad del sistema.

Varianza para la clasificación: Estabilidad del sistema, usando 2 dato(s) por ciclo

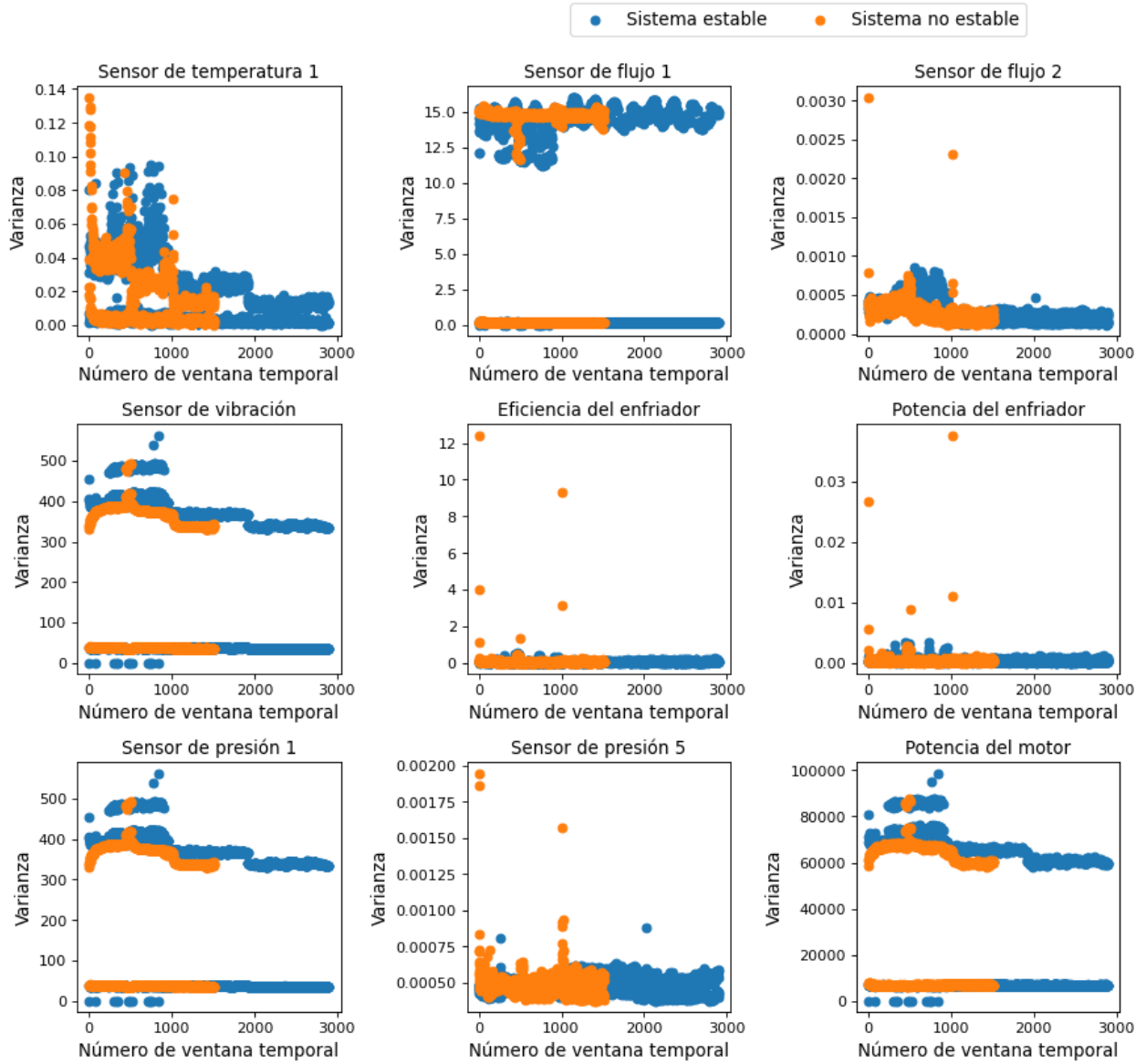


Figura A.154: Gráficos de la varianza obtenida según las clases de la estabilidad del sistema.

Media para la clasificación: Estabilidad del sistema,
usando 2 dato(s) por ciclo

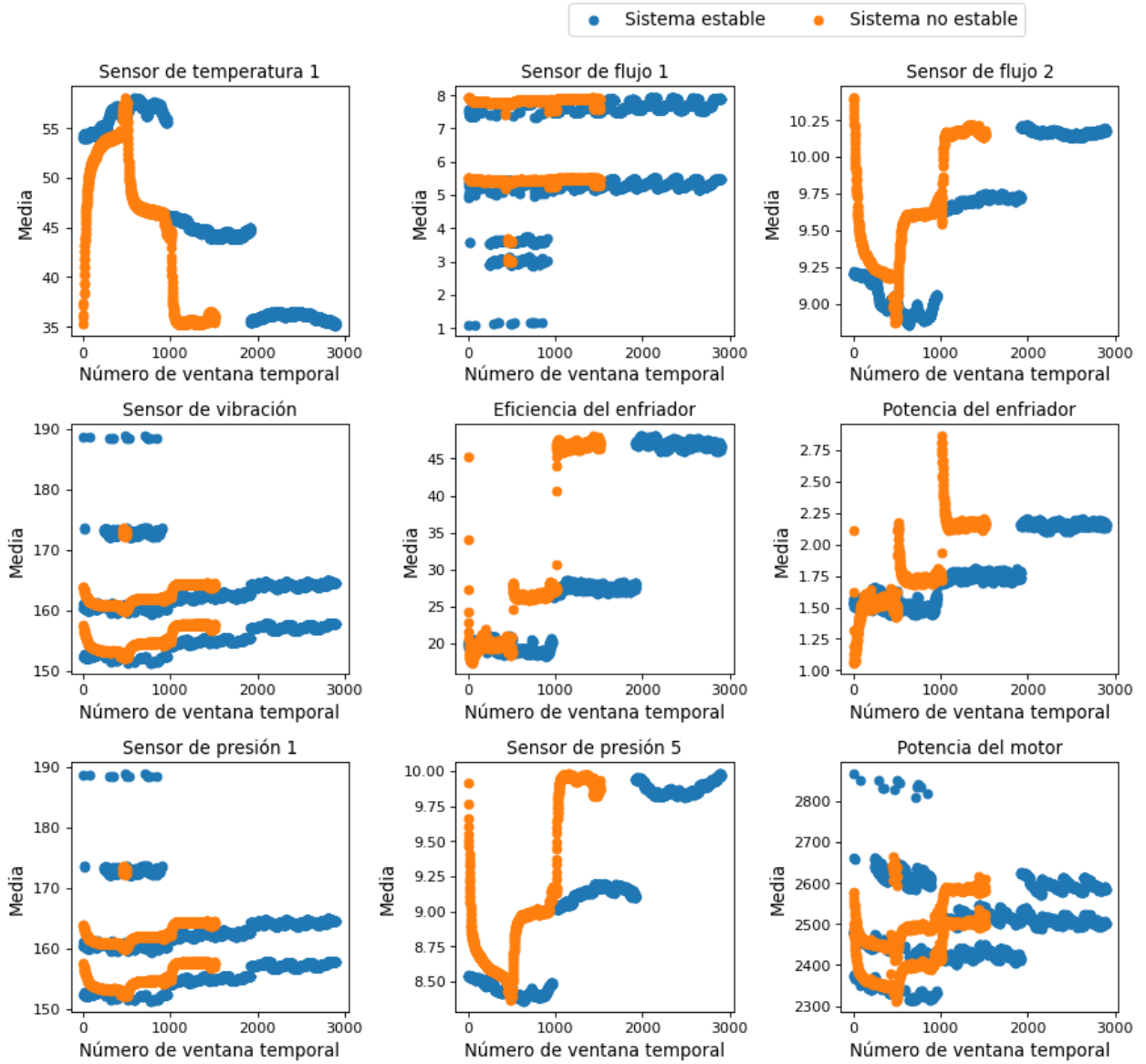


Figura A.155: Gráficos de la media obtenida según las clases de la estabilidad del sistema.

Valor peak to peak para la clasificación: Estabilidad del sistema, usando 2 dato(s) por ciclo

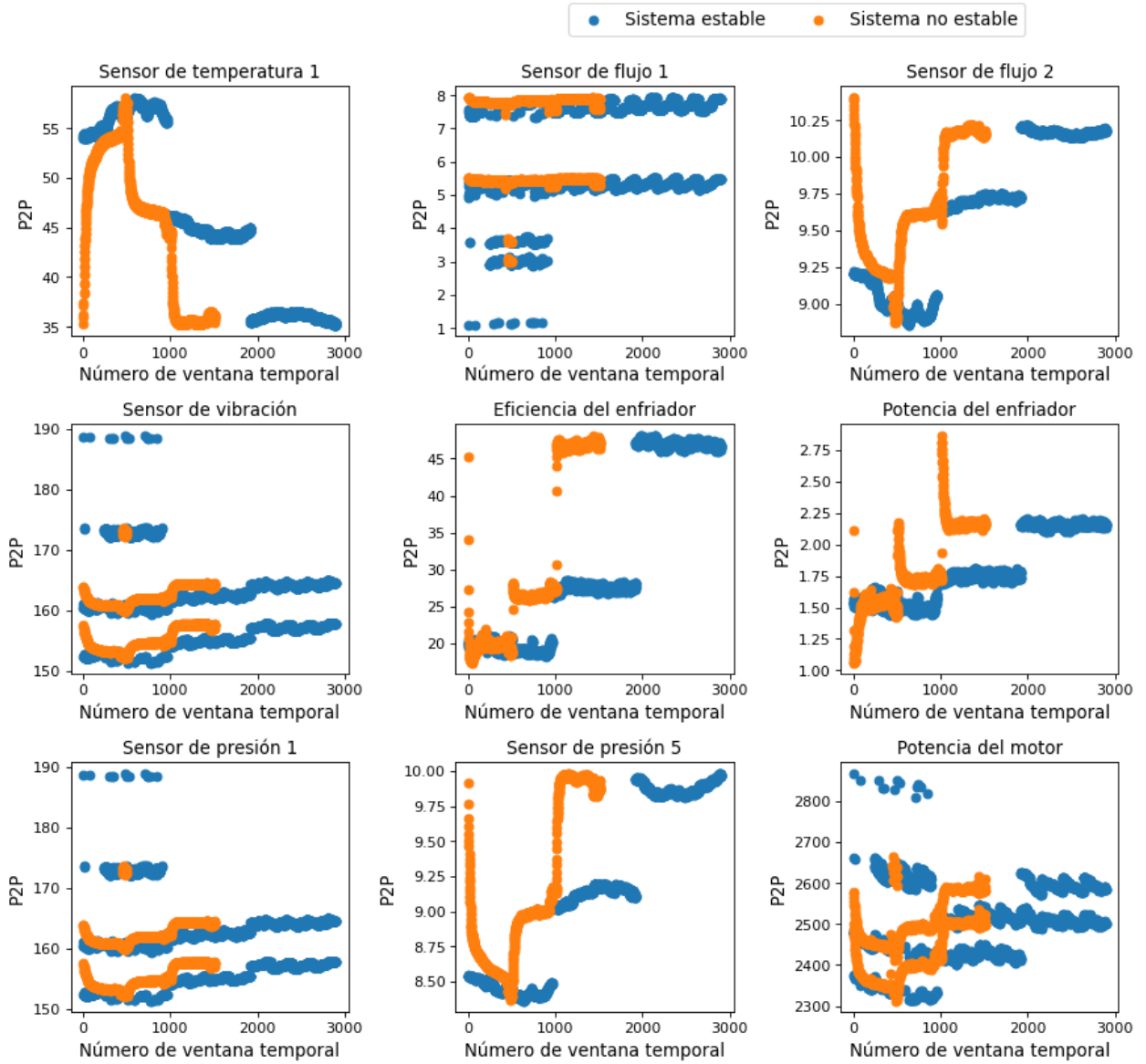


Figura A.156: Gráficos del valor peak to peak obtenido según las clases de la estabilidad del sistema.

A.2.5.3. Gráficos usando 6 datos por ciclo

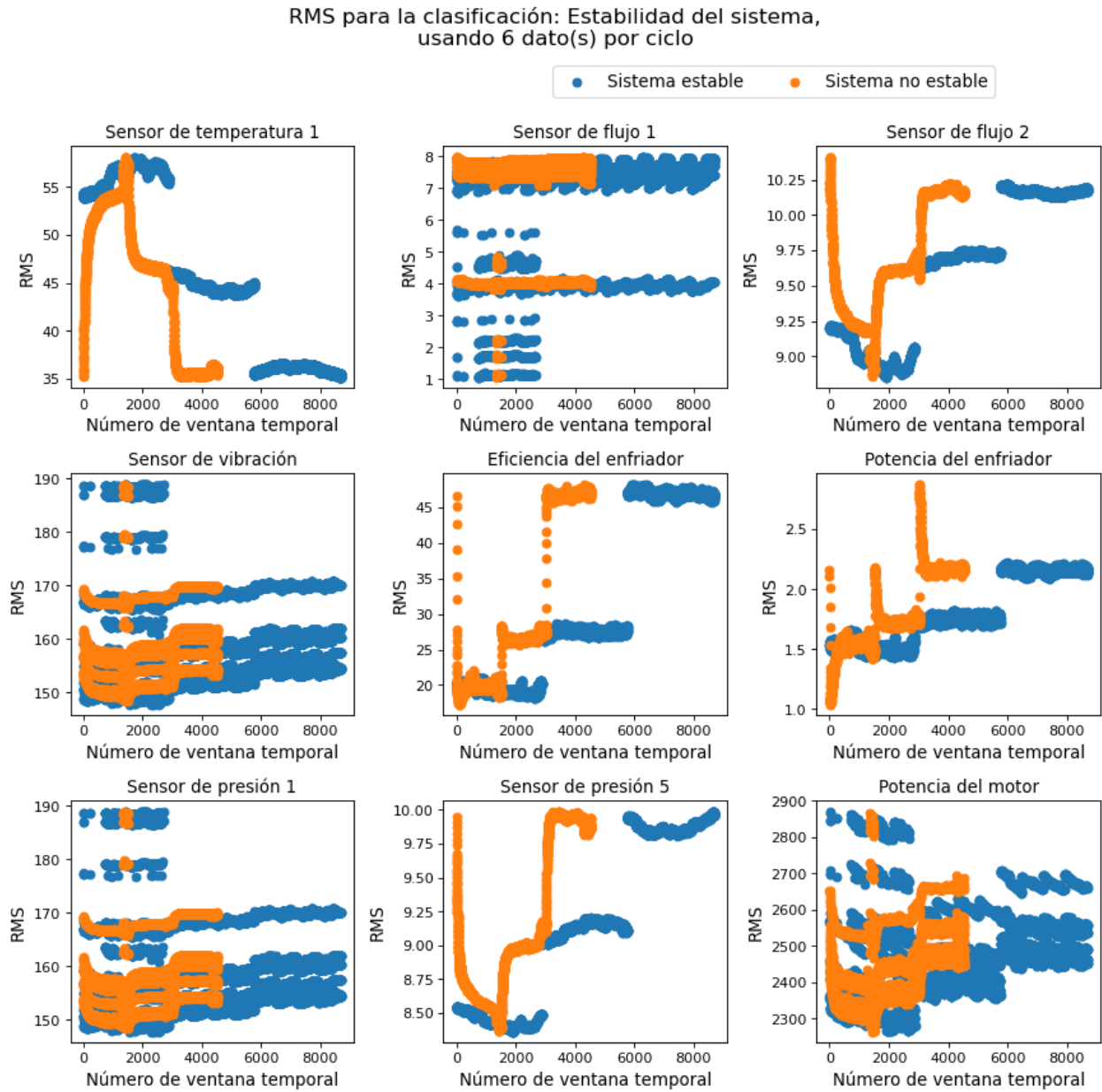


Figura A.157: Gráficos del RMS obtenido según las clases de la estabilidad del sistema.

Varianza para la clasificación: Estabilidad del sistema, usando 6 dato(s) por ciclo

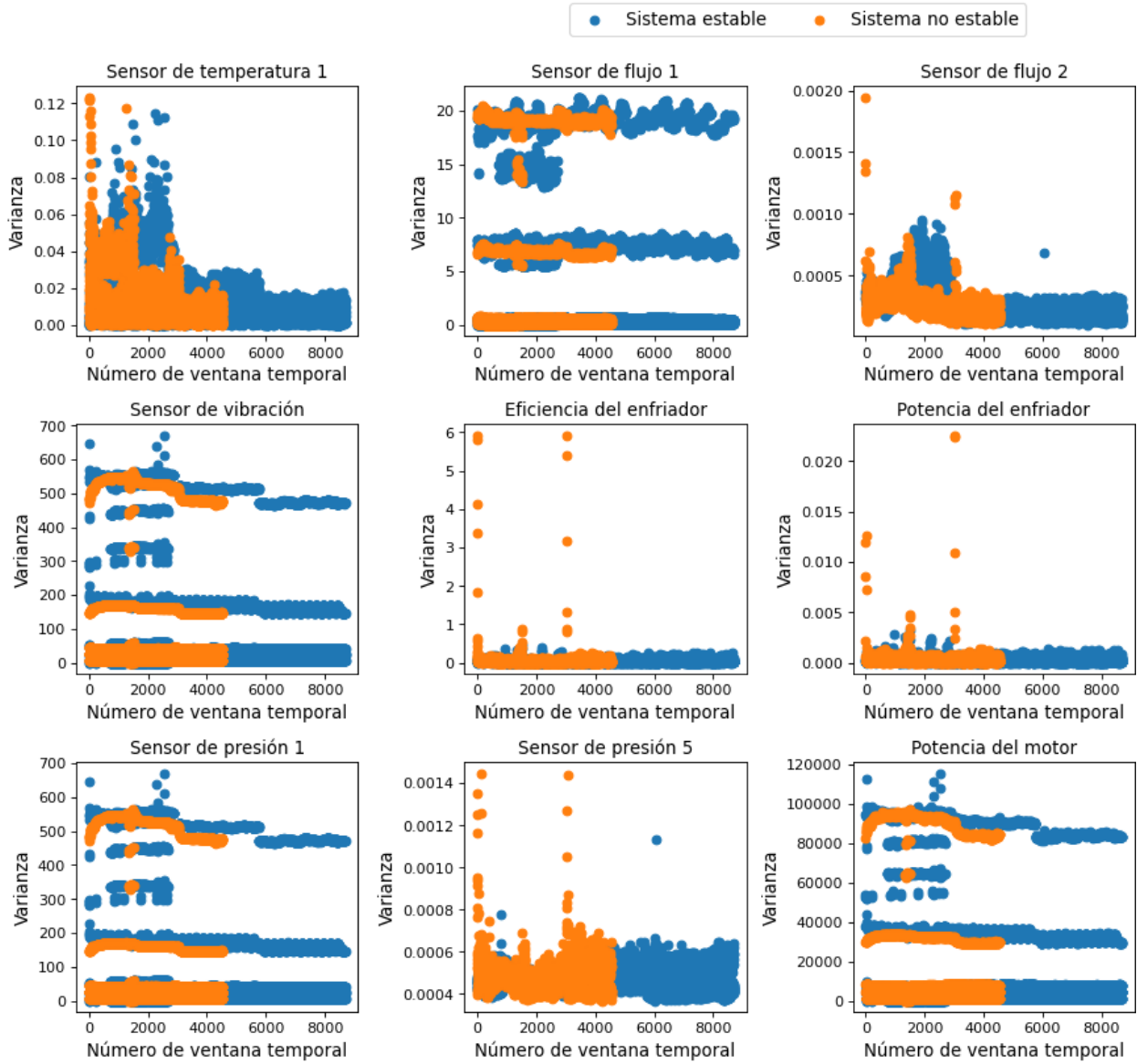


Figura A.158: Gráficos de la varianza obtenida según las clases de la estabilidad del sistema.

Media para la clasificación: Estabilidad del sistema,
usando 6 dato(s) por ciclo

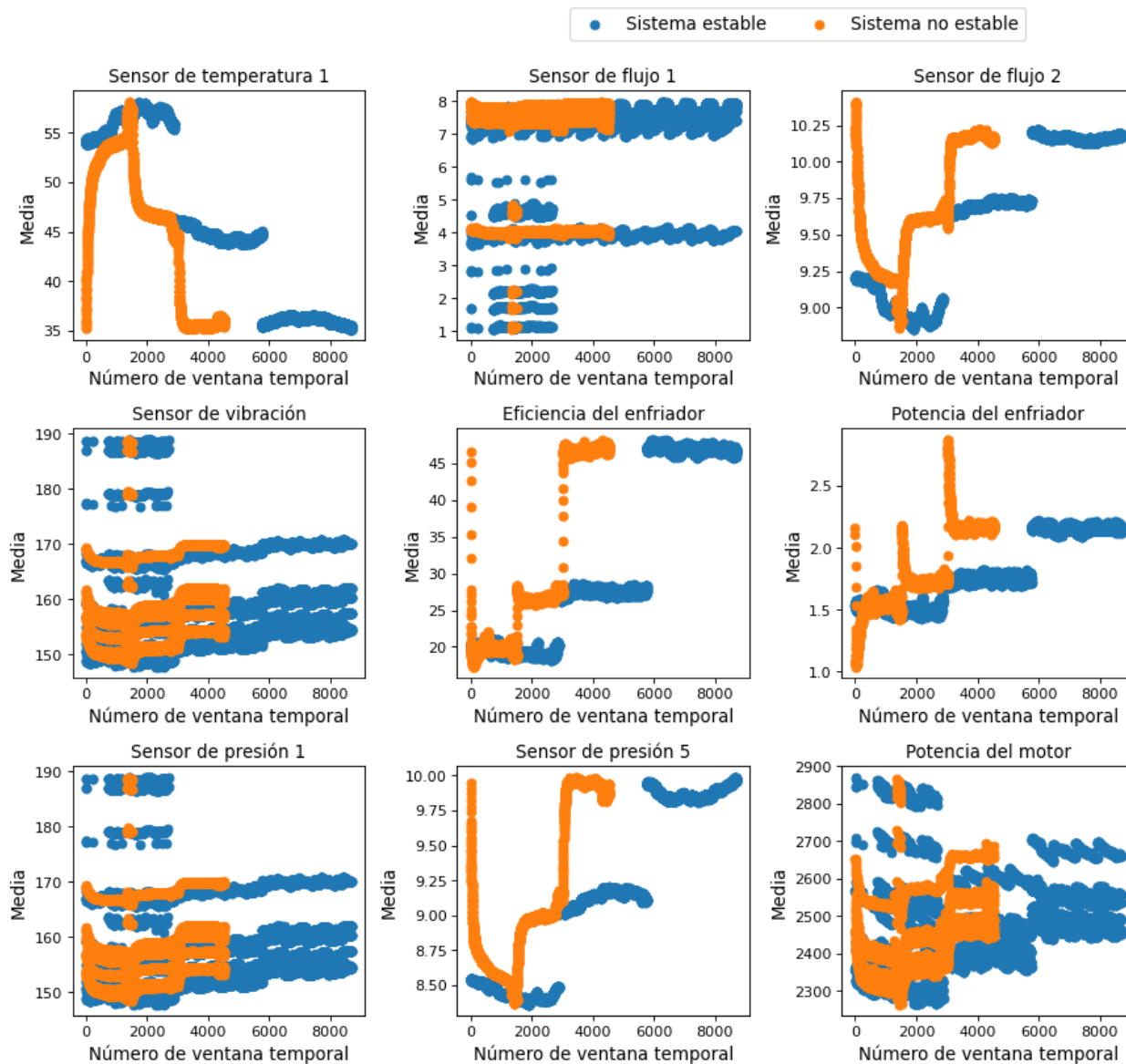


Figura A.159: Gráficos de la media obtenida según las clases de la estabilidad del sistema.

Valor peak to peak para la clasificación: Estabilidad del sistema, usando 6 dato(s) por ciclo

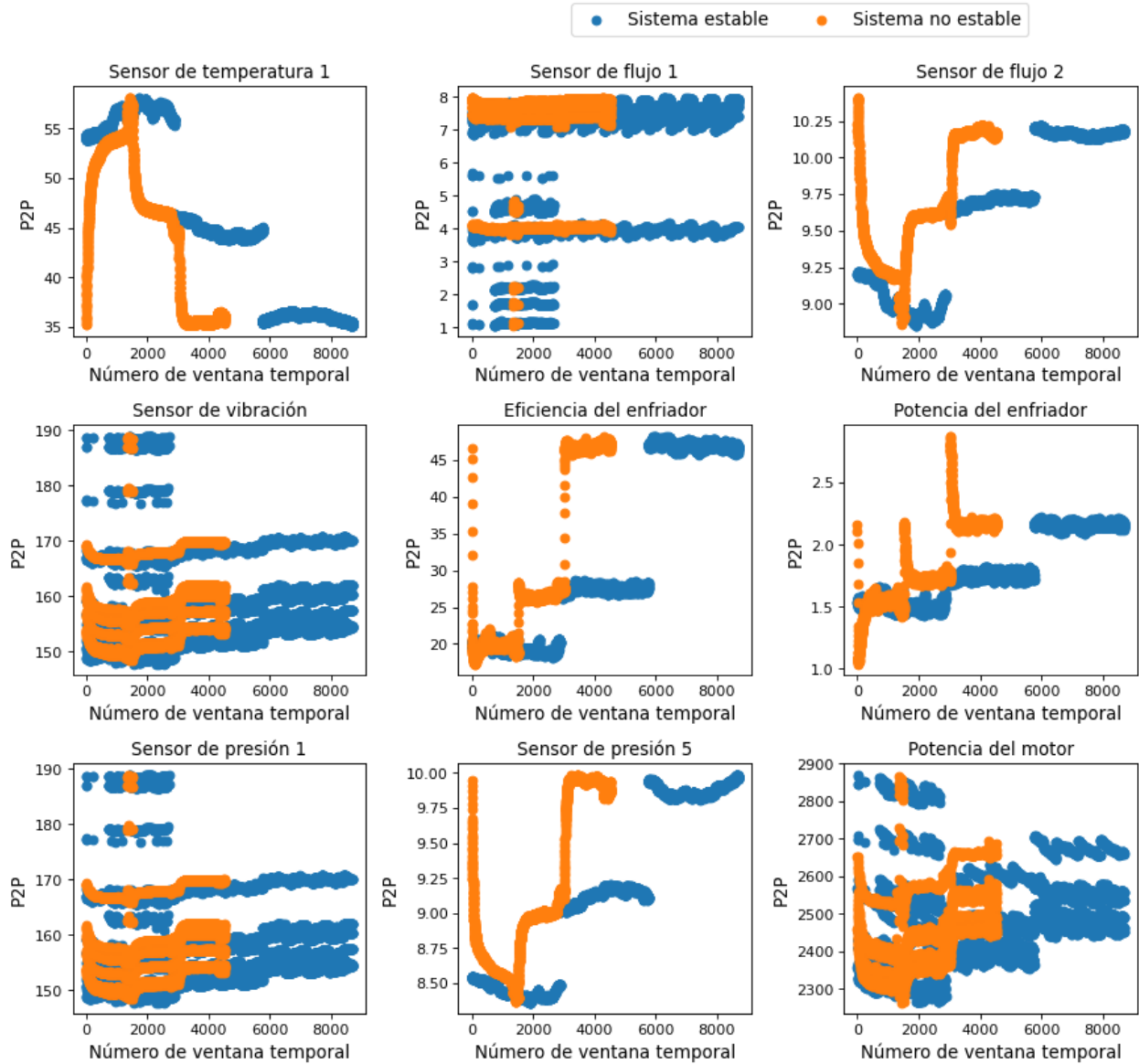


Figura A.160: Gráficos del valor peak to peak obtenido según las clases de la estabilidad del sistema.

A.3. Resultados a partir de los modelos RF

Accuracias a partir del modelo RF, clasificación: Estado del enfriador

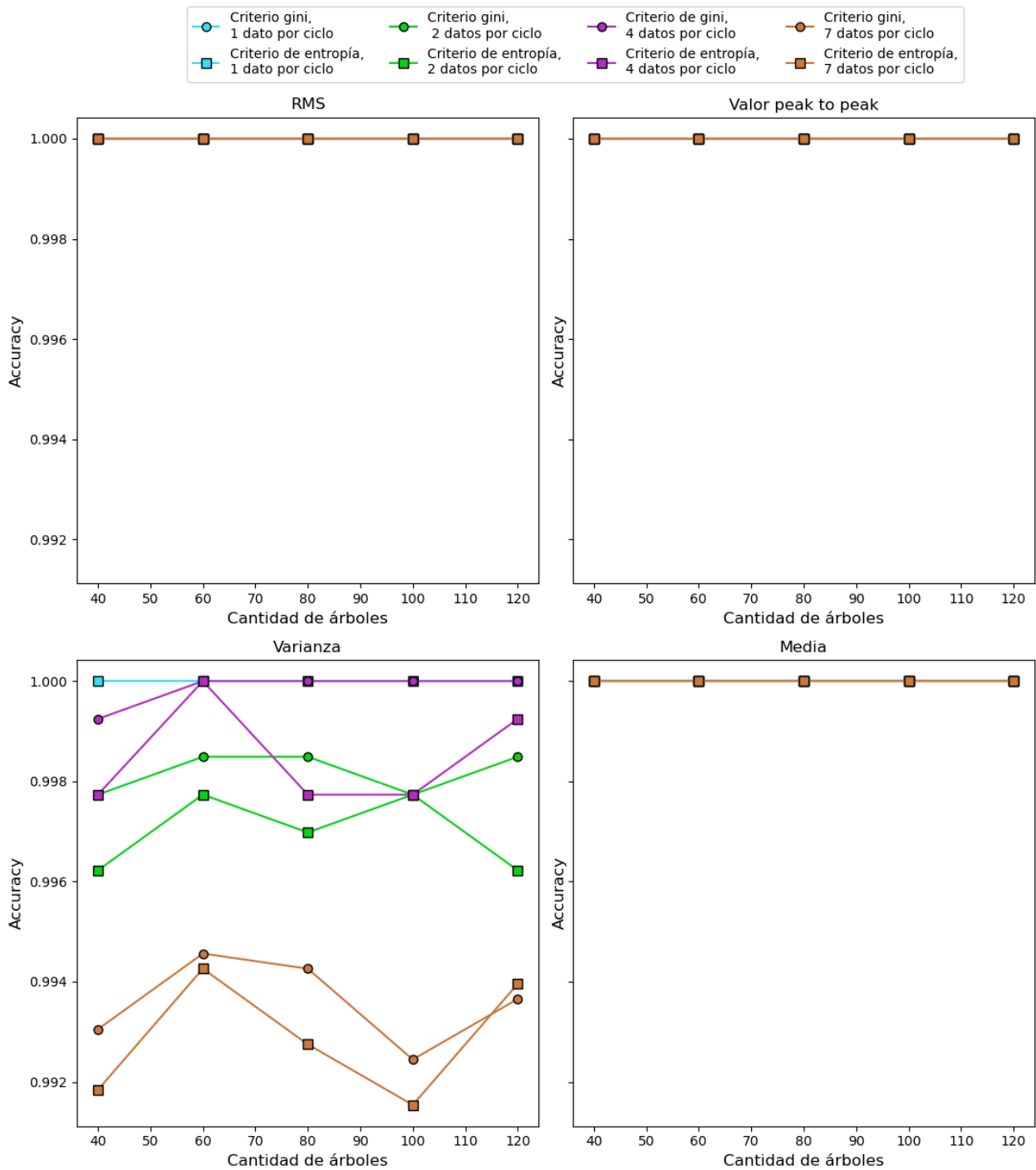


Figura A.161: Valores de accuracy obtenidos de modelos RF.

Accuracias a partir del modelo RF, clasificación: Estado de la válvula

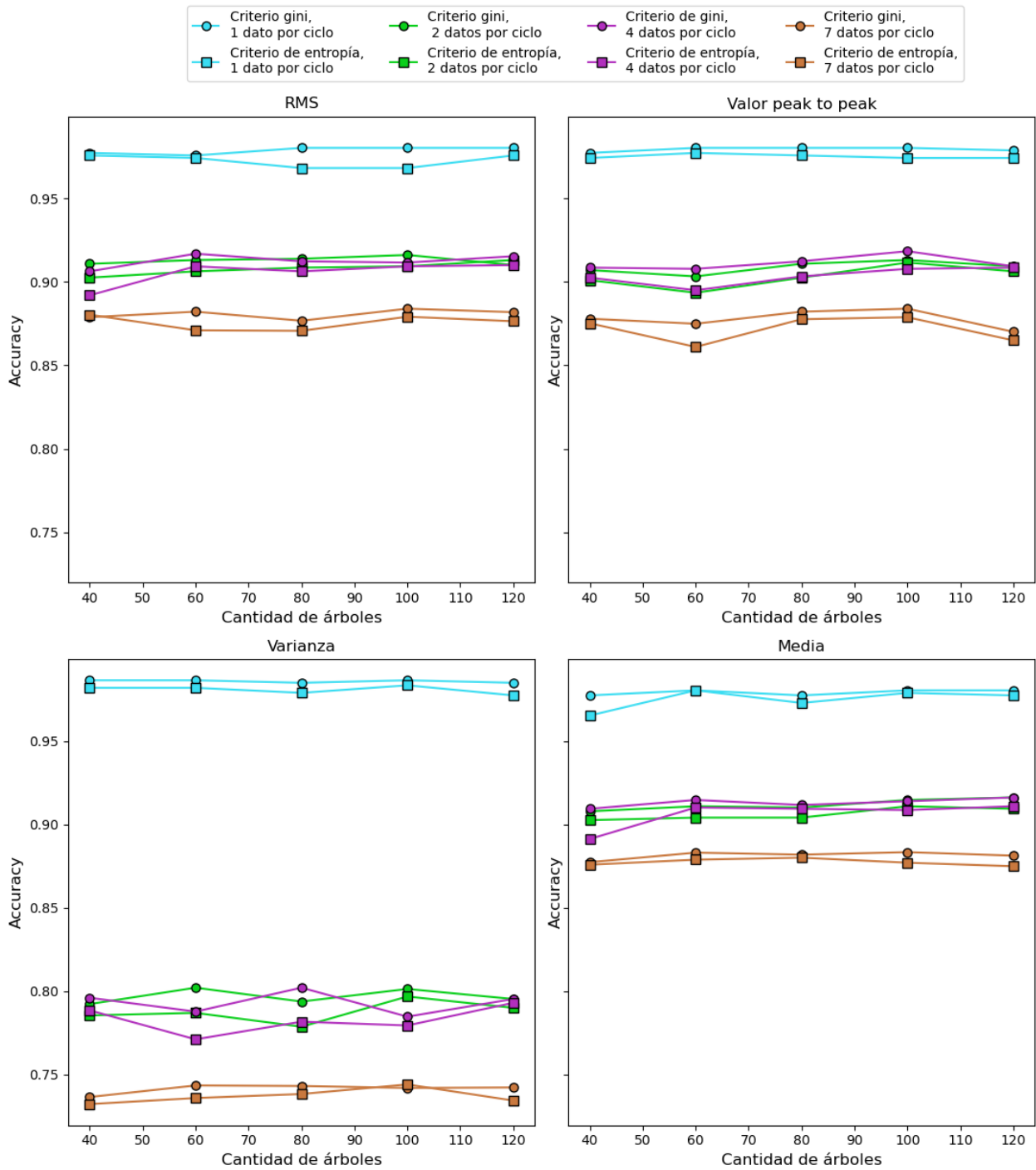


Figura A.162: Valores de accuracy obtenidos de modelos RF.

Accuracias a partir del modelo RF, clasificación: Fuga en la bomba

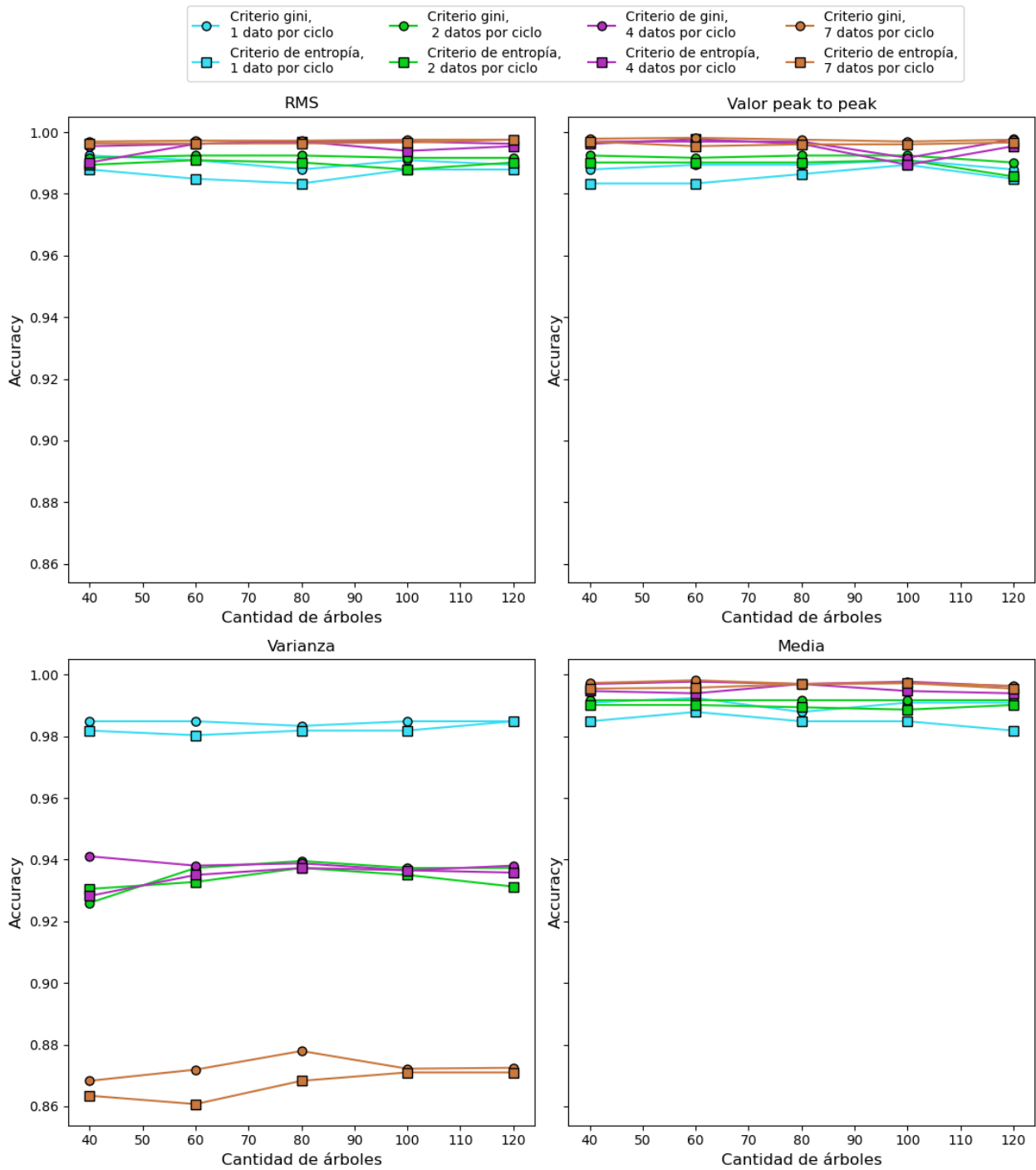


Figura A.163: Valores de accuracy obtenidos de modelos RF.

Accuracias a partir del modelo RF, clasificación: Estado del acumulador

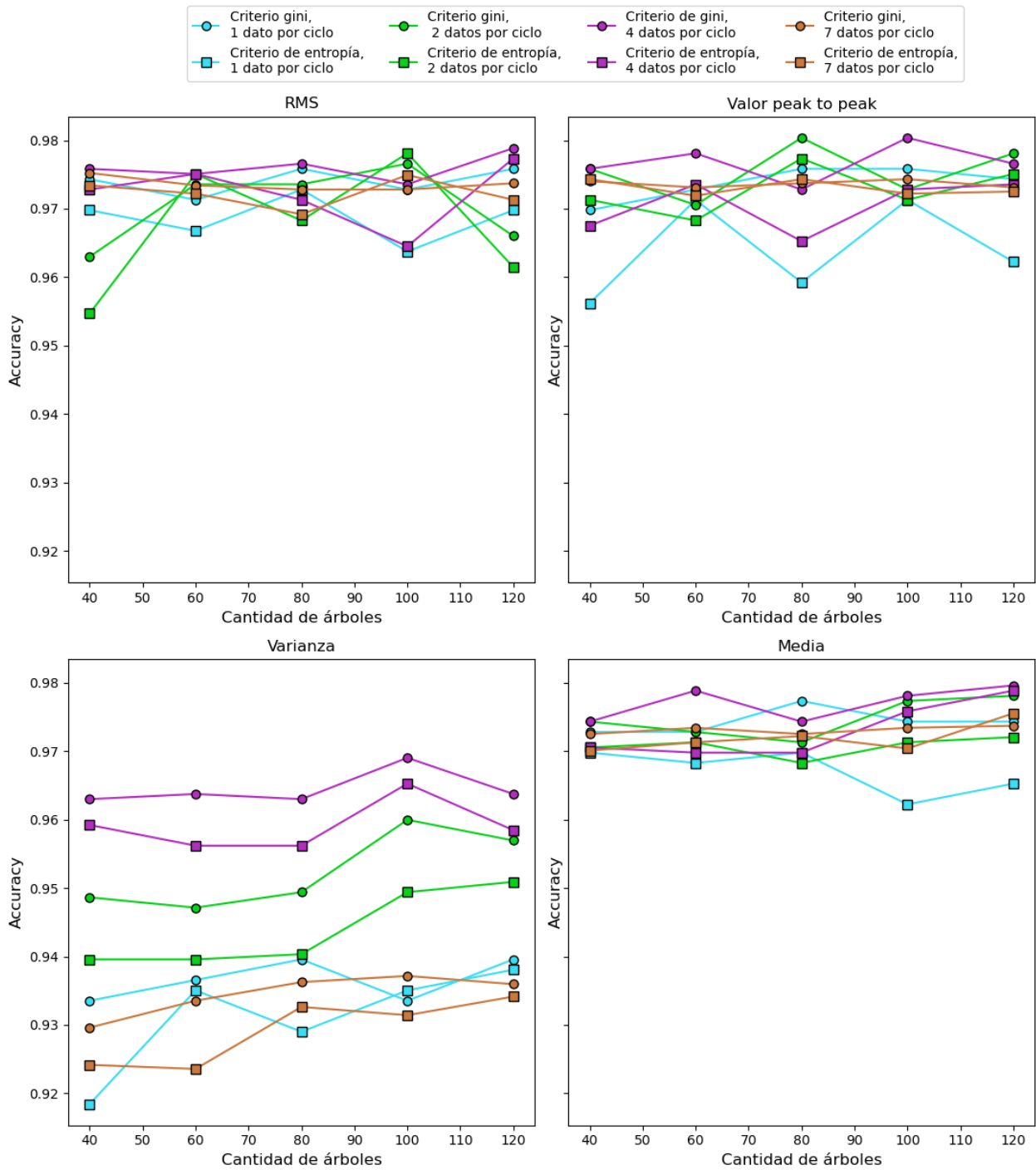


Figura A.164: Valores de accuracy obtenidos de modelos RF.

Accuracies a partir del modelo RF, clasificación: Estabilidad del sistema

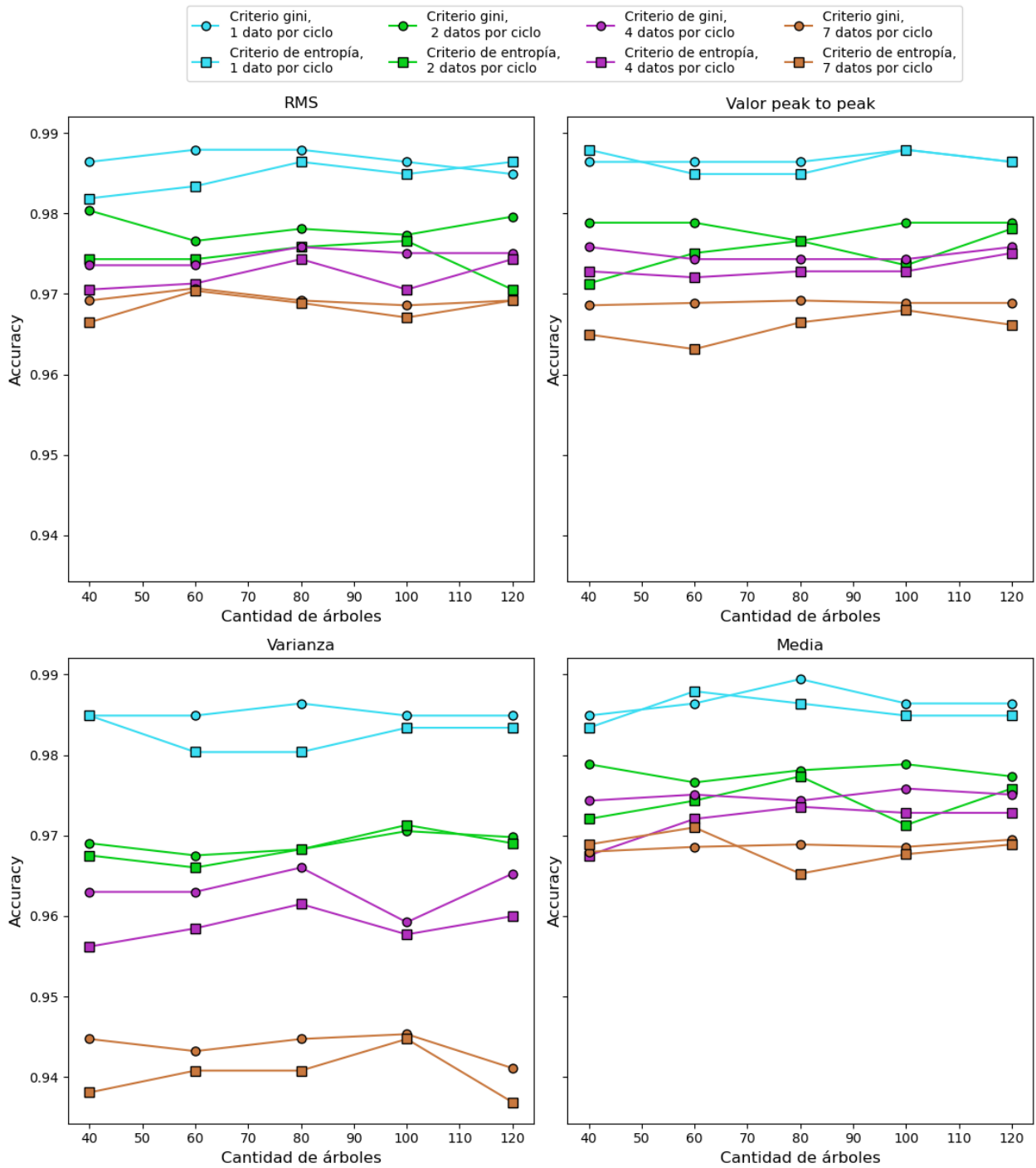


Figura A.165: Valores de accuracy obtenidos de modelos RF.

A.4. Resultados a partir de los modelos KNN

Accuracias a partir del modelo KNN, clasificación: Estado del enfriador

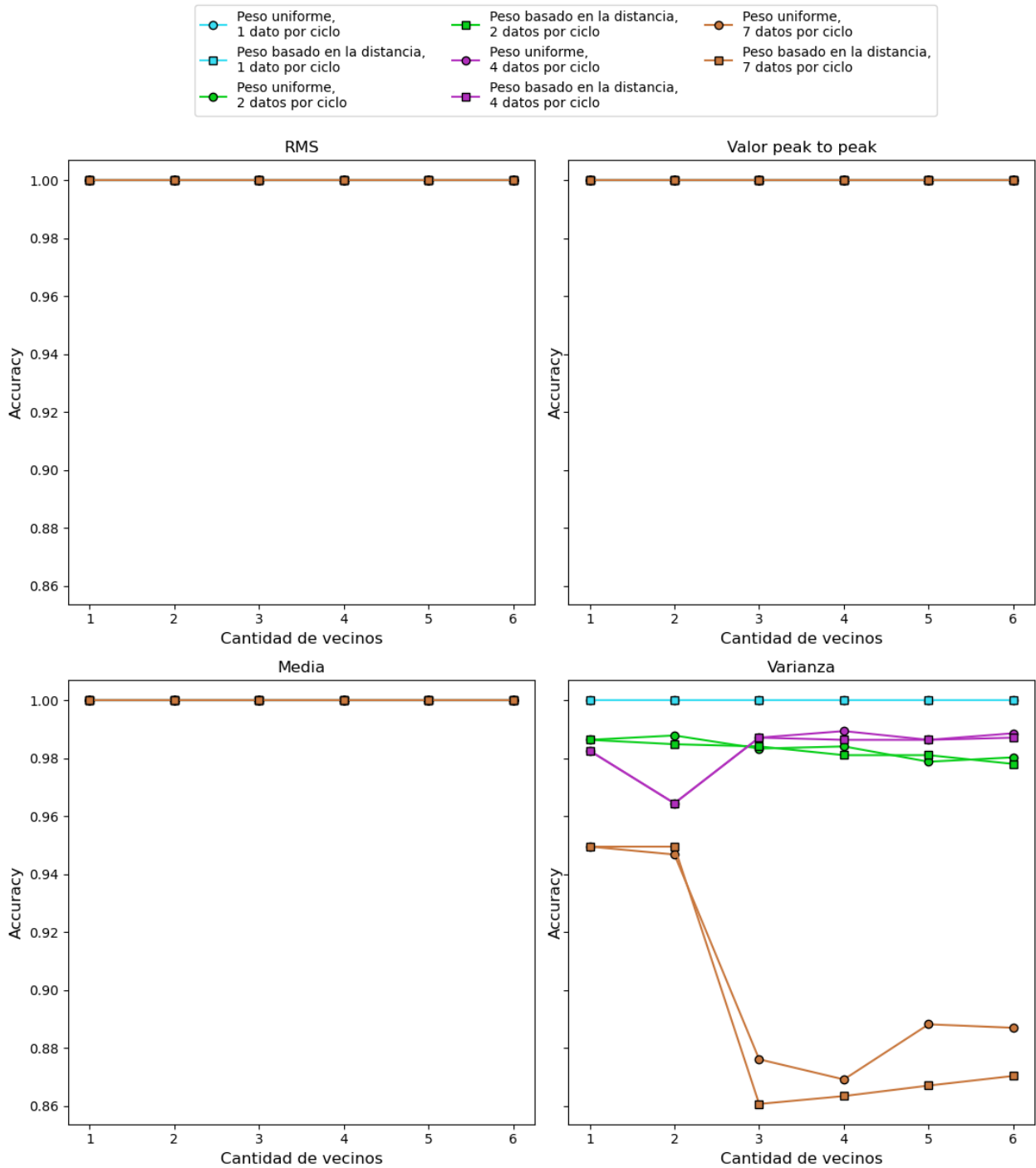


Figura A.166: Valores de accuracy obtenidos de modelos KNN.

Accuracias a partir del modelo KNN, clasificación: Estado de la válvula

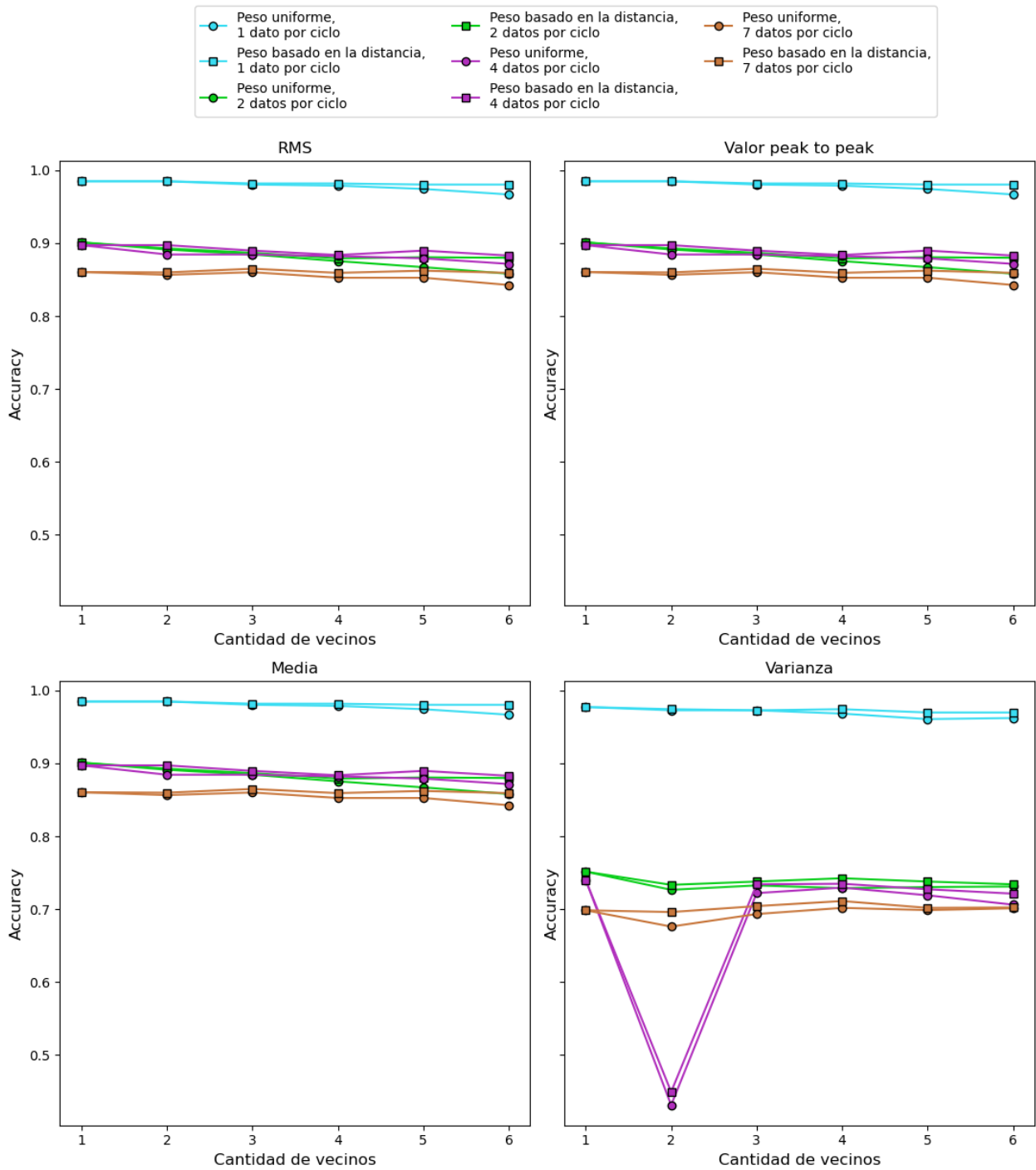


Figura A.167: Valores de accuracy obtenidos de modelos KNN.

Accuracias a partir del modelo KNN, clasificación: Fuga en la bomba

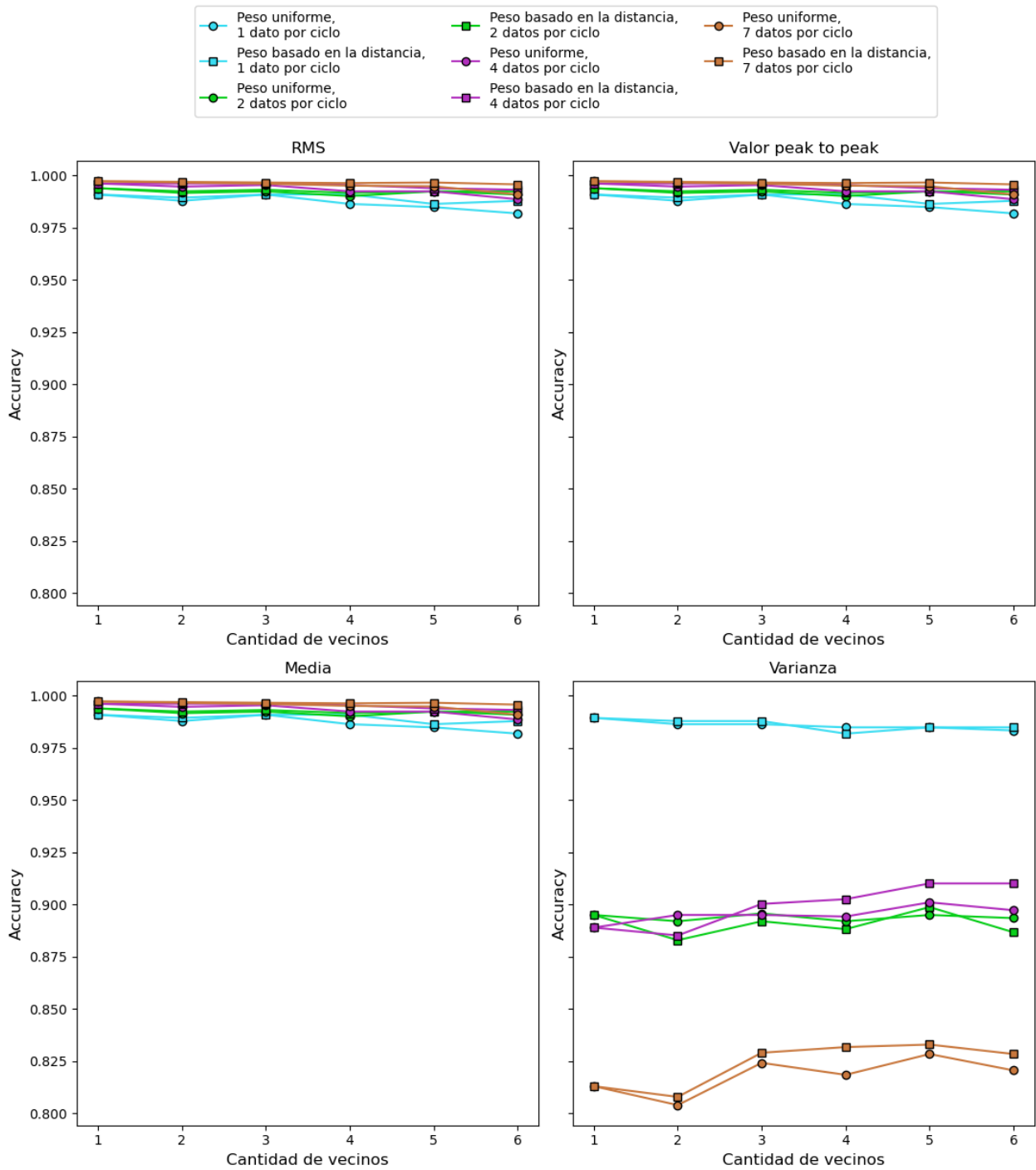


Figura A.168: Valores de accuracy obtenidos de modelos KNN.

Accuracias a partir del modelo KNN, clasificación: Estado del acumulador

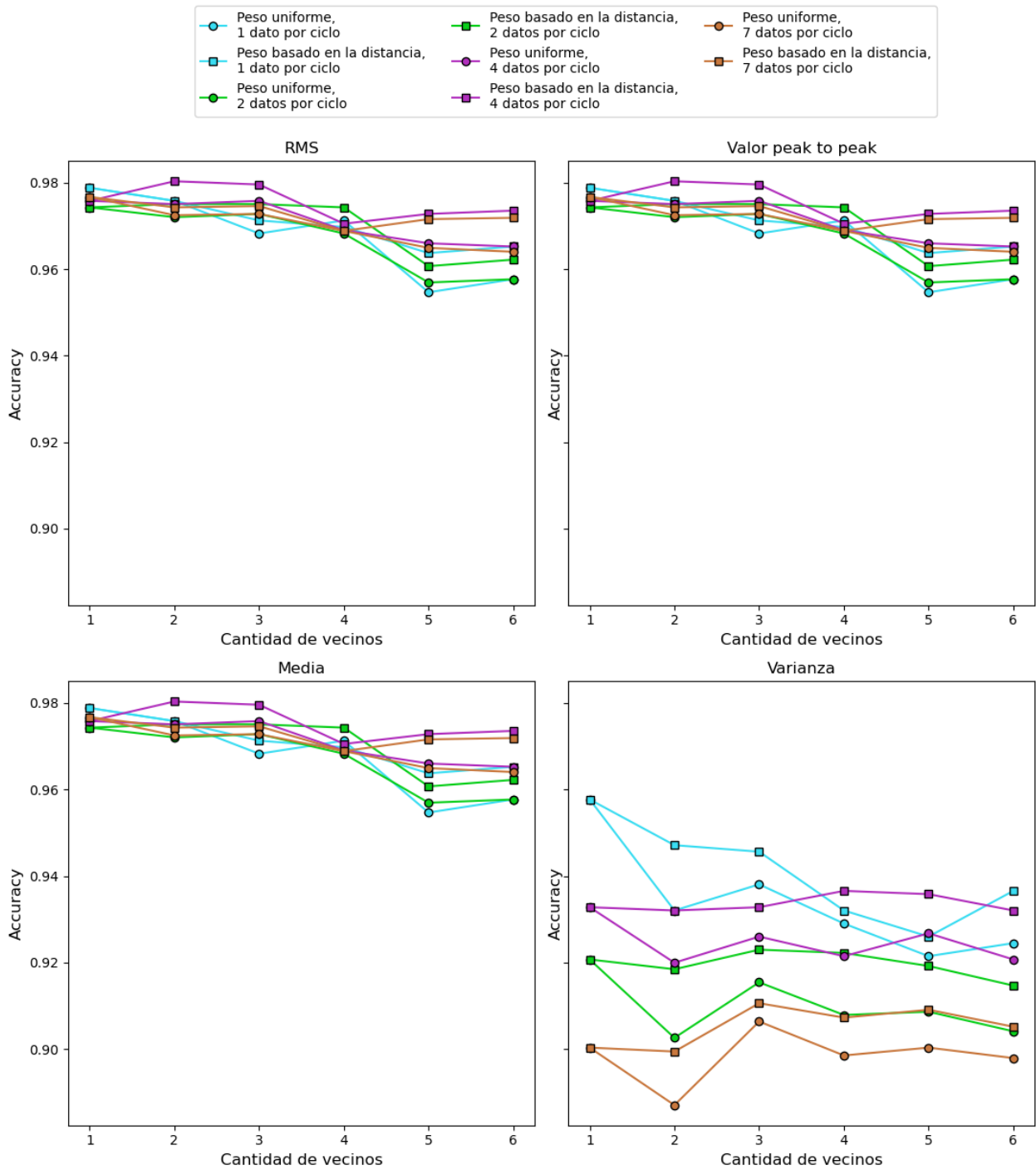


Figura A.169: Valores de accuracy obtenidos de modelos KNN.

Accuracias a partir del modelo KNN, clasificación: Estabilidad del sistema

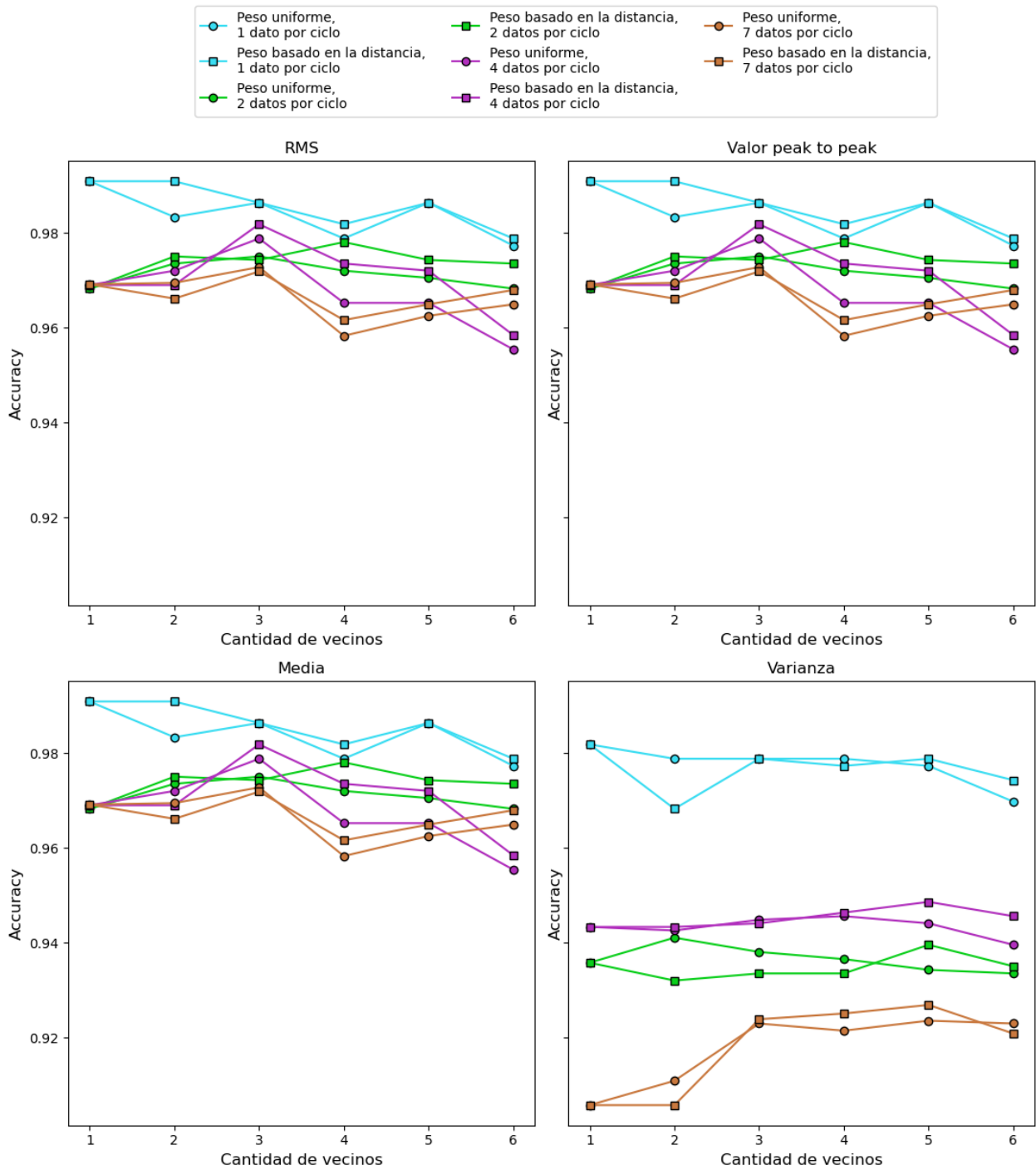


Figura A.170: Valores de accuracy obtenidos de modelos KNN.

A.5. Resultados a partir de los modelos SVM

A.5.1. Accuracy en función de la cantidad de ventanas de entrenamiento para distintos kernel

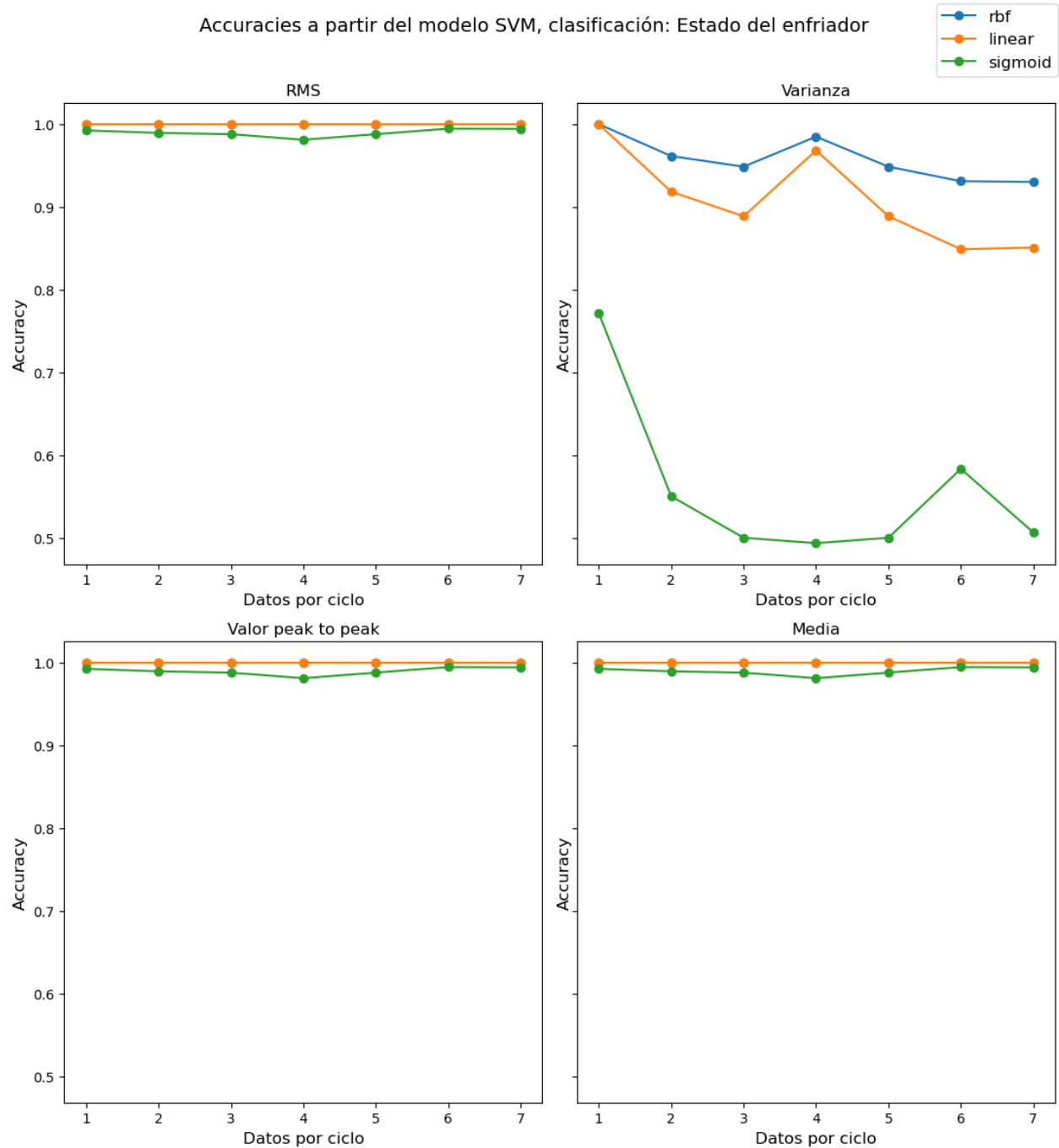


Figura A.171: Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.

Accuracias a partir del modelo SVM, clasificación: Estado de la válvula

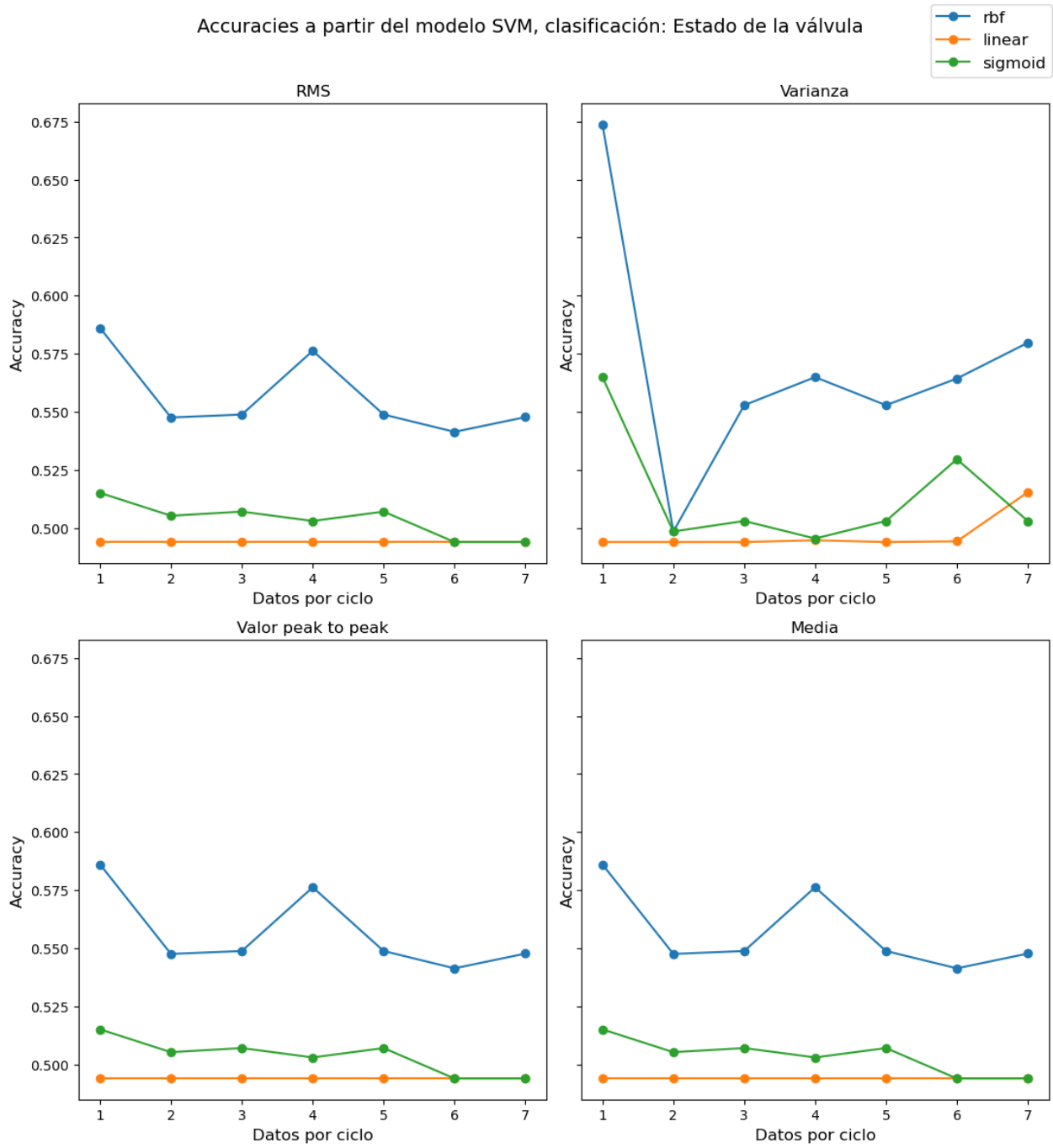


Figura A.172: Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.

Accuracies a partir del modelo SVM, clasificación: Fuga en la bomba

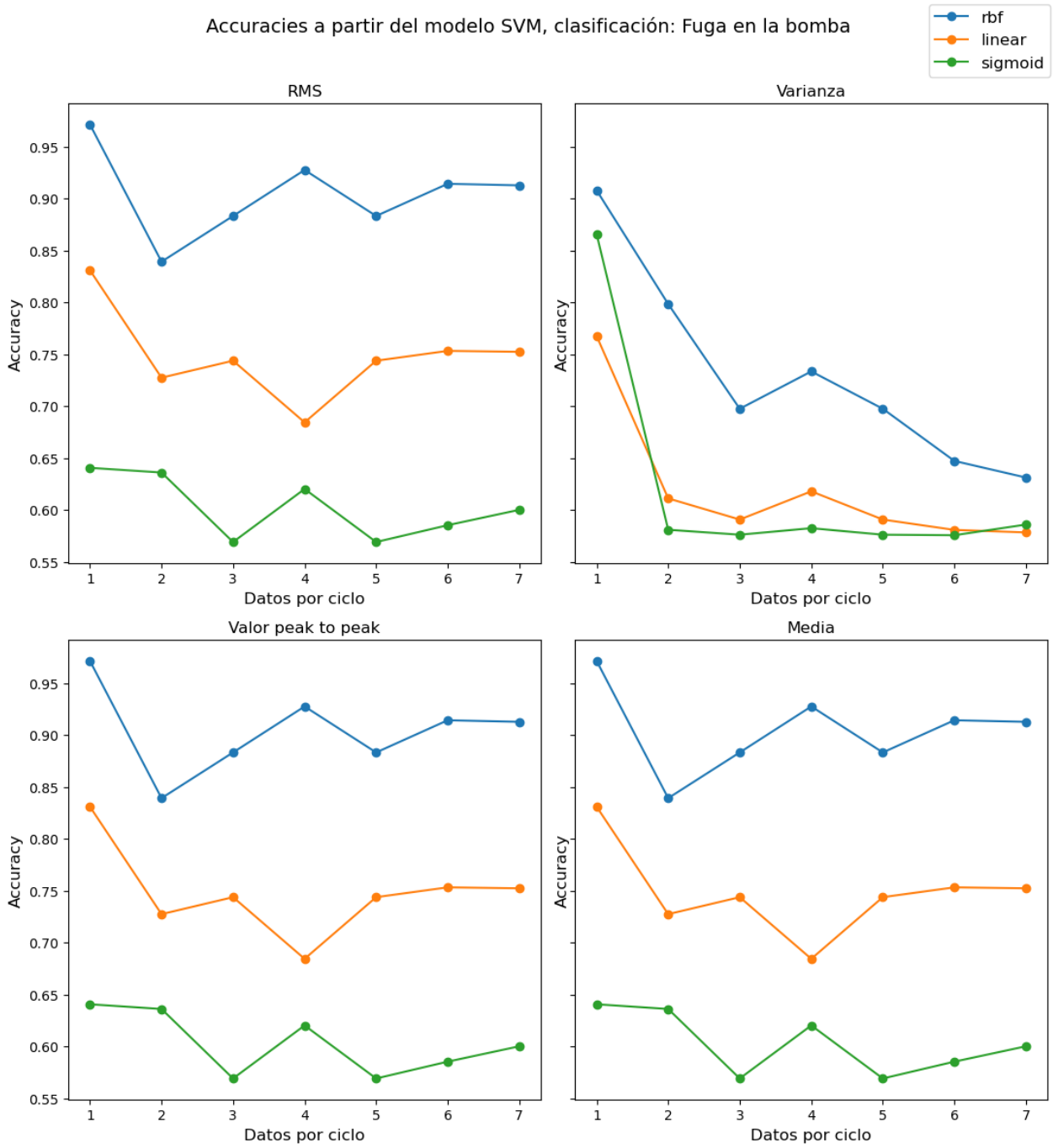


Figura A.173: Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.

Accuracias a partir del modelo SVM, clasificación: Estado del acumulador

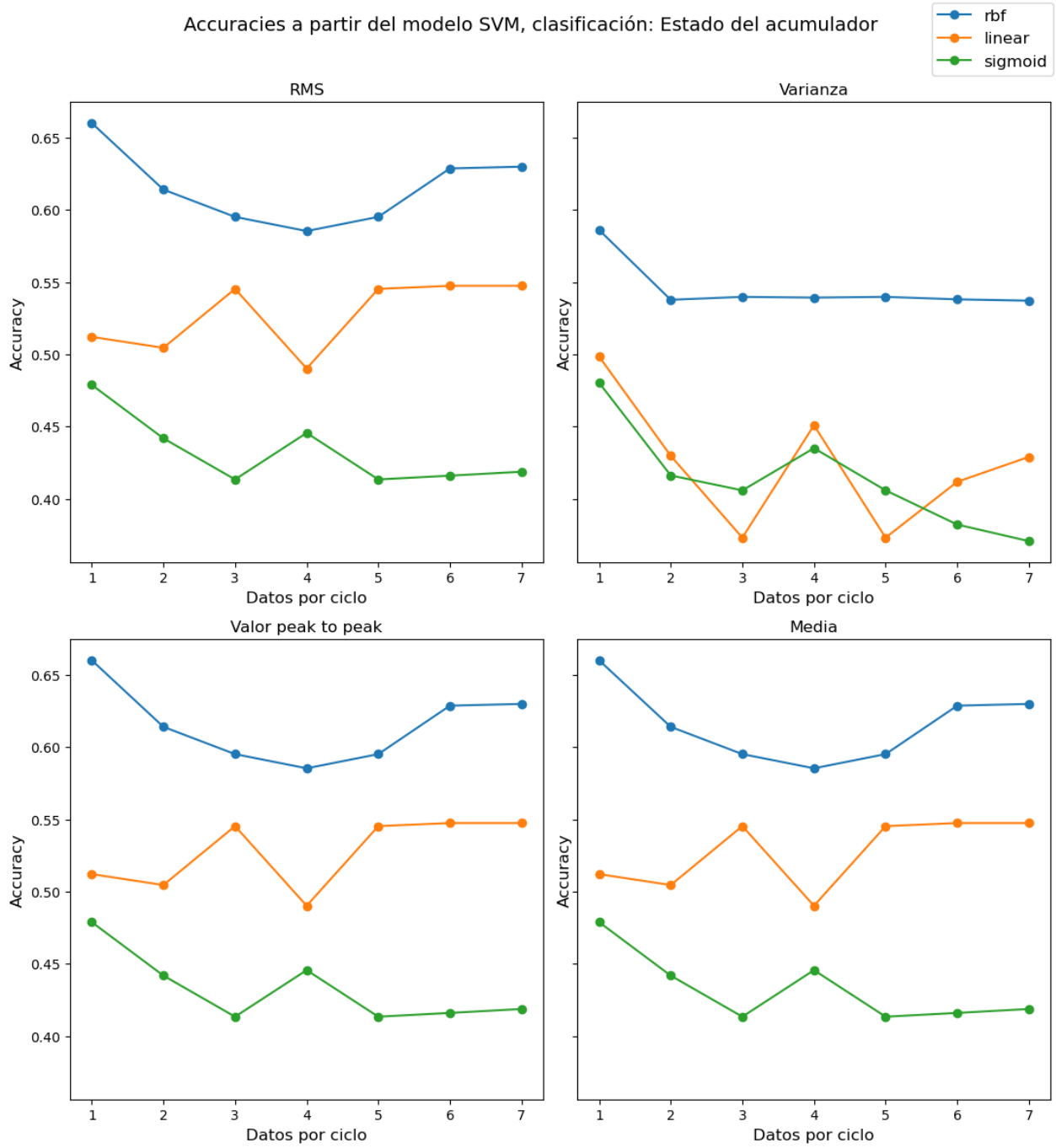


Figura A.174: Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.

Accuracies a partir del modelo SVM, clasificación: Estabilidad del sistema

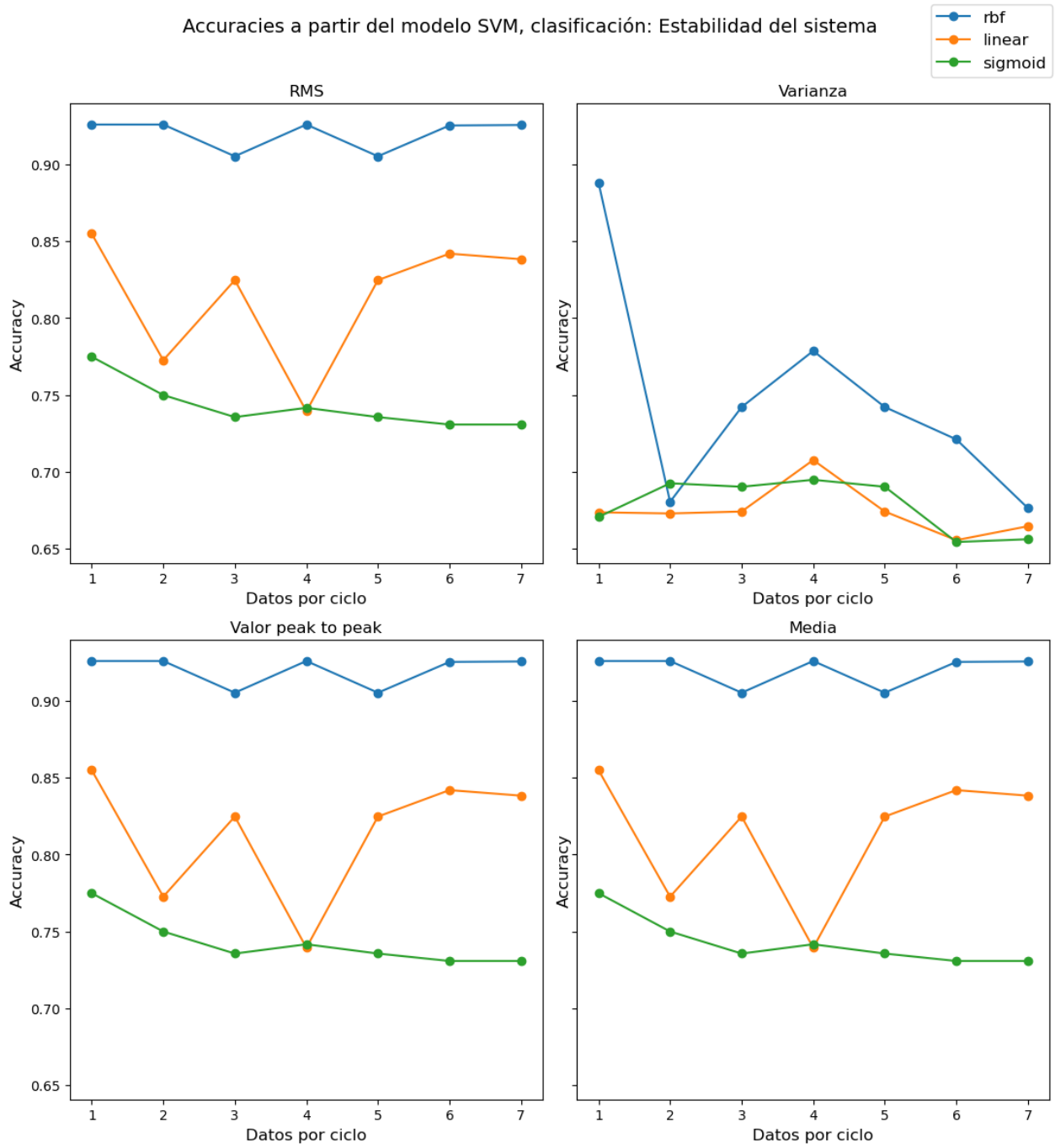


Figura A.175: Valores de accuracy obtenidos de modelos SVM para diferentes parámetros de tiempo.

A.5.2. Accuracy al variar los parámetros “C” y “gamma”

Accuracias obtenidas de la clasificación: Estado del enfriador,
usando 1 dato(s) por ciclo y un kernel rbf

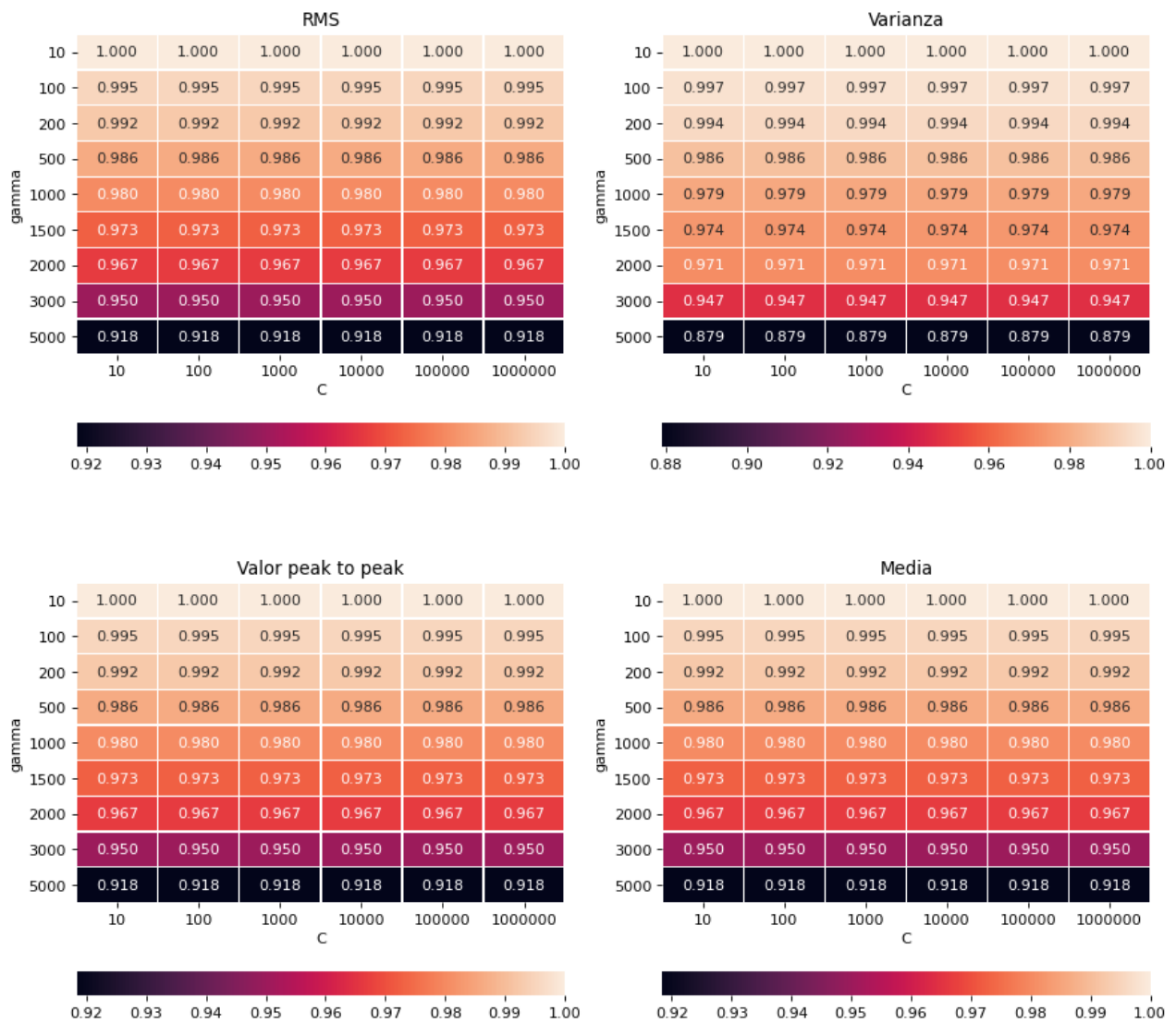


Figura A.176: Heatmap con la variación del accuracy en función de los parámetros c y γ , para un kernel ‘rbf’ y 1 dato extraído por ciclo.

Accuracias obtenidas de la clasificación: Estado de la válvula,
usando 1 dato(s) por ciclo y un kernel rbf

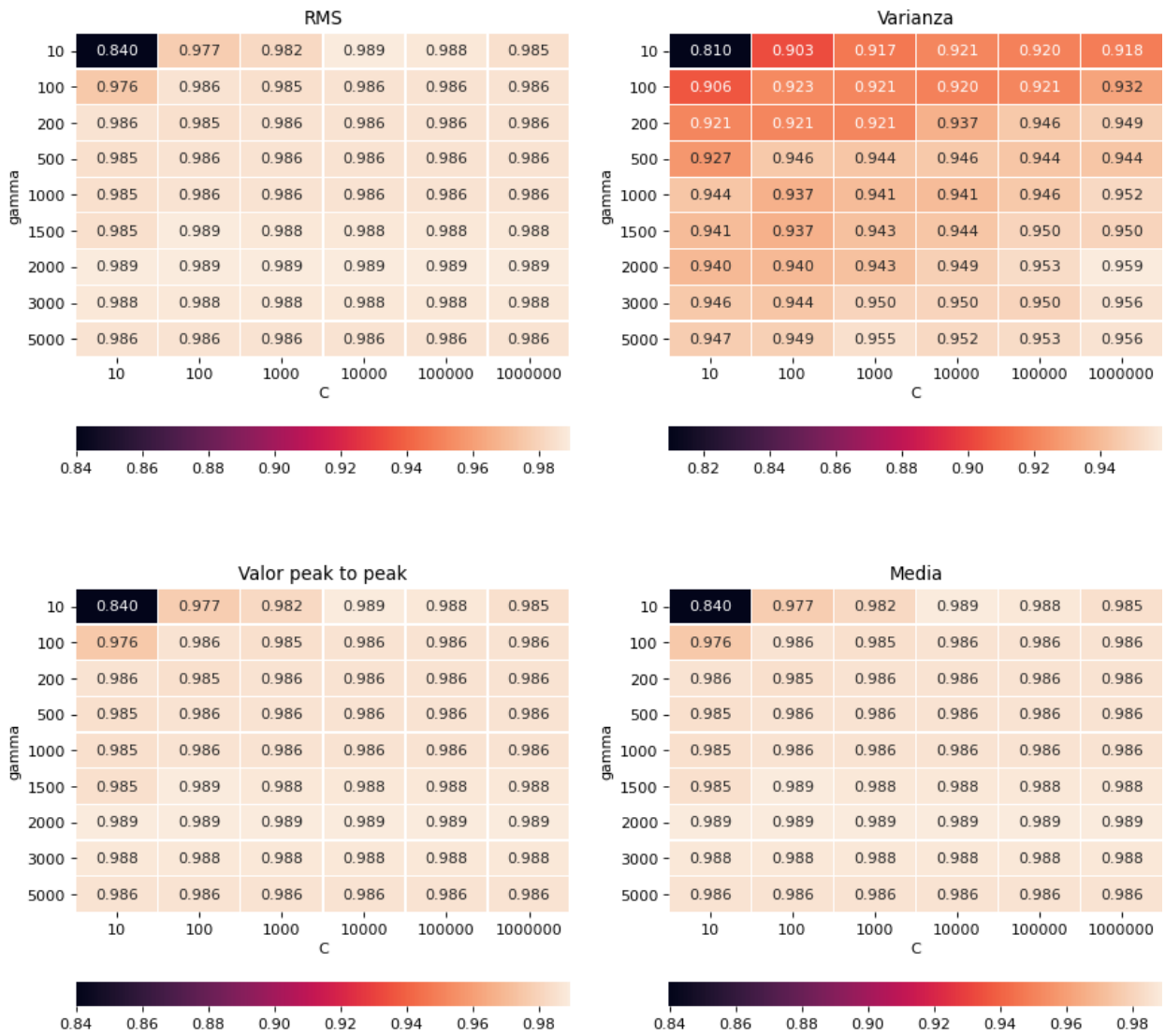


Figura A.177: Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel 'rbf' y 1 dato extraído por ciclo.

Accuracies obtenidas de la clasificación: Fuga en la bomba,
usando 1 dato(s) por ciclo y un kernel rbf



Figura A.178: Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel 'rbf' y 1 dato extraído por ciclo.

Accuracias obtenidas de la clasificación: Estado del acumulador,
usando 1 dato(s) por ciclo y un kernel rbf

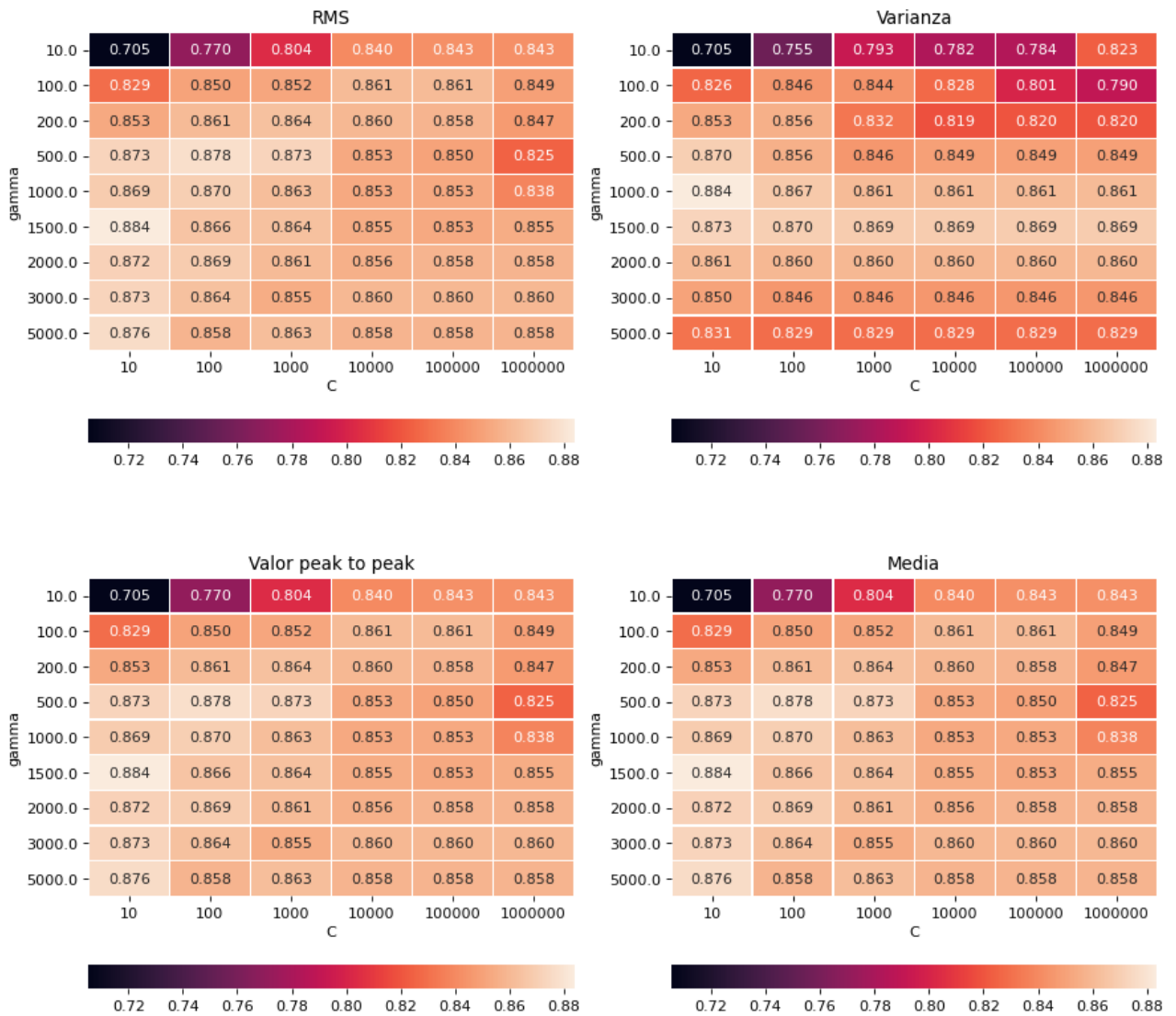


Figura A.179: Heatmap con la variación del accuracy en función de los parámetros c y gamma, para un kernel 'rbf' y 1 dato extraído por ciclo.

Accuracias obtenidas de la clasificación: Estabilidad del sistema,
usando 1 dato(s) por ciclo y un kernel rbf



Figura A.180: Heatmap con la variación del accuracy en función de los parámetros c y γ , para un kernel 'rbf' y 1 dato extraído por ciclo.