Universidad Veracruzana

Centro de Investigación en Inteligencia Artificial

Selección del Modelo Completo en Bases de Datos Temporales a través de Metaheurísticas

TESIS

Tesis presentada en cumplimiento para obtener el grado de:

Doctora en Inteligencia Artificial

PRESENTA:

Nancy Pérez Castro

DIRECTOR:
Dr. Héctor Gabriel Acosta-Mesa

CO-ASESORES:
Dr. Efrén Mezura-Montes
Dr. Nicandro Cruz-Ramírez

Xalapa, Ver., Octubre 2020

Selección del Modelo Completo en Bases de Datos Temporales a través de Metaheurísticas

por

Nancy Pérez-Castro

Tesis presentada en cumplimiento para obtener el grado de

Doctora en
Inteligencia Artificial

Centro de Investigación en Inteligencia Artificial,

Director:	Dr. Héctor Gabriel Acosta-Mesa
Co-asesores:	Dr. Efrén Mezura-Montes
	Dr. Nicandro Cruz-Ramírez

RESUMEN

Las series de tiempo son una clase especial de datos temporales que representan una colección de observaciones realizadas cronológicamente en sucesivos periodos de tiempo. Este tipo de datos se pueden encontrar en diversas aplicaciones como: análisis de mercado de valores, mediciones de fenómenos naturales, tratamientos médicos, entre otras.

La minería de datos de series de tiempo revela diversas etapas de complejidad. Dentro de los problemas más destacados se presenta la alta dimensionalidad que caracteriza a las series de tiempo, así como la dificultad de definir una forma de similitud basada en la percepción humana. Esto dificulta el procesamiento de las series de tiempo a traves de los métodos tradicionales Minería de Datos, produciendo un bajo rendimiento.

Se han propuesto una gran variedad de métodos para el procesamiento de series de tiempo que se enfocan principalmente en tres actividades: pre-procesamiento, representación y aprendizaje supervisado.

La selección de cualquier tipo de método así como el ajuste de sus parámetros resultan ser tareas poco triviales para usuarios poco expertos. En la literatura se han reportado estudios que se enfocan en la tarea de seleccionar un algoritmo para una tarea en particular o de trabajar en el ajuste de parámetros de un método en especial.

En este trabajo se presenta una propuesta para tratar con el problema de Selección de Modelo Completo en bases de datos temporales. Este problema consiste en encontrar una combinación adecuada de métodos y la optimización de sus hiperparámetros que mejor predigan a un conjunto de datos.

El trabajo se acota a la búsqueda de modelos completos que incluyan la combinación adecuada de un sub-conjunto de métodos y la optimización de sus hiperparámetros para llevar a cabo cuatro tareas que son: suavizado de ruido, reducción de dimensionalidad (representación de series de tiempo), selección de instancias y clasificación de series de tiempo. Las metaheurísticas, en especial aquellas que están inspirada en la naturaleza, representan una atractiva alternativa como herramienta de búsqueda de modelos completos.

Este trabajo se divide en tres estudios principalmente. El primero es un estudio comparativo de variantes micro de la metaheurística de Evolución Diferencial, denominada en este trabajo como μ-DEMS. El segundo estudio se centra en la comparación entre metaheurísticas basadas en población y metaheurísticas basadas en trayectoria de una sola solución. Para el caso de las variantes de metaheurísticas basadas en trayectoria se explora la codificación mixta y la codificación binaria de las soluciones. Finalmente, en el tercer estudio se plantea una propuesta basada en una metaheurística para la optimización multi-objetivo, denominada como N2FMS. Se propone la optimización de dos objetivos, uno que optimiza el rendimiento basado en la tasa de error del modelo y otro que optimiza el tiempo que consume un modelo durante la etapa de entrenamiento. Además se proponen cuatro estrategias para la selección de una solución del conjunto de soluciones óptimas de Pareto.

Agradecimientos

Quiero hacer los siguientes agradecimientos:

- Gracias Dios por sostenerme y nunca abandonarme, a pesar de mis tropiezos!
- Quiero agradecer a mi familia, en especial a mi mamá *Bertha Castro Díaz* por creer en mi y brindarme su apoyo constante y sus miles de bendiciones. A mi hermana Gladys por animarme en los días grises.
- Gracias a mi esposo por apoyarme y acompañarme, Te amo $\pi\xi$
- A mi Director de tesis y Co-asesores por otorgarme su valiosa opinión y apoyo.
- Al personal académico y administrativo del Centro de Investigación en Inteligencia Artificial (CIIA) por brindarme su apoyo durante el periodo que cursé el doctorado.
- A mis amigos y compañeros del doctorado, gracias por su apoyo y su amistad.
- Al consejo Nacional de Ciencia y Tecnología (CONACyT) por su patrocinio y apoyo en la realización de mis estudios de doctorado, con el número de beca 259655.
- Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), por permitirme realizar mi estancia doctoral y brindarme apoyo a través del proyecto CONACyT CB-0241306.

Índice general

Índice de figuras VII

Índice de tablas XXIII

I Introducción 1

1 Introducción 2
1.1 Motivación 2
1.2 Planteamiento del problema 4
1.3 Objetivos 4
1.3.1 Objetivo general 4
1.3.2 Objetivos específicos 5
1.4 Hipótesis 5
1.5 Justificación 6
1.6 Contribuciones 7
1.7 Estructura del documento 7

2 Revisión de la literatura 9
2.1 Propuestas basadas en Meta-aprendizaje (MT) 9
2.2 Propuestas basadas en métodos embebidos (EM) 11
2.3 Enfoques de envoltorio evolutivo (EW) 12

II Marco Teórico 15

3 Minería de datos temporal 16
3.1 Minería de Datos y Descubrimiento de Conocimiento 16
3.2 Bases de datos temporales 17
3.2.1 Series de tiempo 18
3.3 Pre-procesamiento de datos temporales 19
3.3.1 Reducción de ruido 19
3.3.1.1 Promedios móviles 19
3.3.1.2 Filtro Savitzky-Golay 20
3.3.1.3 Regresión Local: loess y lowess 20

3.4 Medidas de similitud de series de tiempo 21
3.4.1 Distancia Euclidiana (norma L_2) 22
3.4.2 Distancia Manhattan (norma L_1) 23
3.4.3 Distancia Chebyshev (norma L_∞) 23
3.4.4 Alineación Dinámica Temporal 23
3.4.4.1 Definición de DTW 24
3.4.5 Límite inferior de Keogh (LB_Keogh) 26
3.5 Representación de Series de Tiempo 28
3.5.1 Aproximación Agregada de Piezas (PAA) 29
3.5.2 Aproximación Simbólica de Piezas (SAX) 29
3.5.3 Análisis de Componentes Principales (PCA) 31
3.6 Reducción de datos 31
3.6.1 Monte Carlo 1 32
3.6.2 INSIGHT 32

4 Aprendizaje Automático 34
4.1 Aprendizaje Supervisado: Clasificación 34
4.1.1 k-Vecinos más Cercanos 35
4.1.2 Clasificador Bayesiano 36
4.1.2.1 Teorema de Bayes 37
4.1.2.2 Naïve Bayes 37
4.1.3 Árbol de decisión 38
4.1.4 Máquinas de Soporte Vectorial 40
4.1.5 Ensamble 41
4.2 Medidas de Evaluación y Validación 42
4.2.1 Validación Cruzada 43

5 Optimización 45
5.1 Optimización global 46
5.1.1 Algoritmos para optimización global 46
5.1.1.1 Metaheurísticas basadas en trayectoria 48
5.1.1.2 Metaheurísticas basadas en la población 49
5.2 Optimización multi-objetivo 50
5.2.1 Conceptos básicos 50
5.2.2 Búsqueda y toma de decisiones 52
5.2.3 Algoritmos para optimización Multi-objetivo 53
5.2.3.1 Métodos clásicos 54
5.2.3.2 Métodos alternativos 56

III Experimentos y Resultados 58

6 Hacia la selección del modelo completo en bases de datos temporales mediante micro-Evolución Diferencial 59
6.1 Antecedentes 59

6.2 Propuesta: μ-DE para la selección del modelo ($\mu - DEMS$) 60
6.2.1 Codificación de la solución 61
6.2.2 Decodificación de la solución 63
6.2.3 Función de aptitud 63
6.2.4 Función de restricción del límite 64
6.2.5 Variantes 64
6.3 Experimentos y resultados 64
6.3.1 Configuración experimental 64
6.3.2 Experimento 1: Comparación estadística de resultados numéricos . . 66
6.3.3 Experimento 2: Análisis de convergencia 66
6.3.4 Experimento 3. Análisis de la solución final 68
6.4 Conclusiones 72

7 FMS en bases de datos temporales: comparación de metaheurísticas basada en población y basada en un solo punto 76
7.1 Antecedentes 76
7.2 Descripción de la propuesta 77
7.2.1 Codificación de la solución 77
7.2.1.1 Codificación real 77
7.2.1.2 Codificación binaria 78
7.2.1.3 Codificación mixta 80
7.2.2 Función de aptitud 80
7.2.3 Metaheurísticas para el proceso de búsqueda 80
7.2.3.1 Enfoque PM 80
7.2.3.2 Enfoque SM 80
7.3 Experimentos y resultados 81
7.3.1 Configuración experimental 82
7.3.2 Experimento 1. Resultados estadísticos 84
7.3.3 Experimento 2. Análisis de convergencia 86
7.3.4 Experimento 3. Análisis de diversidad en el enfoque PM 95
7.3.5 Experimento 4. Análisis de los modelos finales 95
7.3.6 Experimento 5. Análisis de frecuencia del uso de los métodos 99
7.4 Conclusiones 99

8 Selección del modelo completo multi-objetivo en bases de datos temporales: optimizando tiempo y rendimiento 102
8.1 Antecedentes 102
8.2 Descripción de la propuesta 103
8.2.1 NSGA-II para resolver el problema FMS (N2FMS) 104
8.2.2 Estrategias para el manejo de preferencias 107
8.2.2.1 S-N2FMS (*Single-NSFMS*) 107
8.2.2.2 NNE-N2FMS (*Near Neighbors Ensemble-N2FMS*) 108
8.2.2.3 IE-N2FMS (*Increasing Ensemble-N2FMS*) 108
8.2.2.4 CE-N2FMS (*Complete Ensemble-N2FMS*) 108
8.3 Experimentos y resultados 109

8.3.1 Experimento 1. Comparación entre estrategias para manejo de preferencias . . . 109
8.3.2 Experimento 2. Comparación entre S-N2FMS y μ-DEMS . . . 110
8.3.3 Experimento 3. Análisis de los frentes acumulados de Pareto . . . 111
8.4 Conclusiones . . . 112

IV Conclusiones finales 118

9 Conclusiones y trabajo futuro 119
9.1 Conclusiones . . . 119
9.2 Trabajo Futuro . . . 121

10 Modelos obtenidos por cada variante de μ-DEMS 123
10.1 Modelos de base de datos Beef . . . 123
10.2 Modelos de base de datos Coffee . . . 125
10.3 Modelos de base de datos ECG200 . . . 127
10.4 Modelos de base de datos OliveOil . . . 129
10.5 Modelos de base de datos FaceFour . . . 131
10.6 Modelos de base de datos GunPoint . . . 133

11 Modelos obtenidos por cada Metaheurística 136
11.1 Modelos de base de datos Beef . . . 136
11.2 Modelos de base de datos CBF . . . 139
11.3 Modelos de base de datos Coffee . . . 142
11.4 Modelos de base de datos ECG200 . . . 145
11.5 Modelos de base de datos FaceFour . . . 148
11.6 Modelos de base de datos GunPoint . . . 151
11.7 Modelos de base de datos Lightning-2 . . . 154
11.8 Modelos de base de datos Lightning-7 . . . 157
11.9 Modelos de base de datos OliveOil . . . 160
11.10 Modelos de base de datos Trace . . . 163

Bibliografía 167

Índice de figuras

1.1 Ejemplo de dos series de tiempo que representa el espectro de una variedad diferente de café. 3
1.2 Representación gráfica del problema de FMS. 5

3.1 Proceso de descubrimiento de conocimiento en datos [53]. 17
3.2 Representación gráfica de la distancia Euclideana y DTW, imágenes tomadas de la fuente Keogh and Ratanamahatana [81]. 23
3.3 Definición de trayectoria de deformación y ajuste de ventana de tamaño r para la alineación de dos series de tiempo, imagen tomada de la fuente Sakoe and Chiba [152]. 25
3.4 Ejemplos de restricciones de la pendiente para $P = \frac{2}{3}$, donde 3 movimientos del punto c_k en la dirección del eje vertical u horizontal, obliga a efectuar al menos dos movimientos en la dirección de la diagonal. (a) Pendiente máxima y (b) pendiente mínima, imágenes tomadas de la fuente Sakoe and Chiba [152]. 26
3.5 Representación gráfica de las restricciones globales de (a) Sakoe-Chiba y (b) Itakura. El área azul muestra el alcance de la trayectoria de deformación, imagen tomada de la fuente Ratanamahatana and Keogh [137]. 27
3.6 Representación gráfica del envoltorio producido por LB_Keogh para una serie donde se usa (a) Sakoe-Chiba e (b) Itakura, imágenes tomadas de la fuente Keogh and Ratanamahatana [81]. 28
3.7 Representación PAA, reducción de una serie de tiempo de 128 de longitud a 8 dimensiones, imagen tomada de la fuente Lin et al. [95]. 29
3.8 Serie de tiempo representada mediante SAX, con $n = 128$, $w = 8$ y $a = 3$, imagen tomada de la fuente Lin et al. [95]. 30

4.1 Ejemplo de k-NN con un valor de k amplio [170]. 36
4.2 Ejemplo de un DT que determina si un cliente comprará una computadora [68]. 39
4.3 Representación lógica del método de aprendizaje por ensambles [170]. 41
4.4 Matriz de confusión para un problema de clasificación binario. Se muestran las instancias positivas correctamente predichas (VP), las instancias positivas incorrectamente predichas (FN), las instancias negativas incorrectamente predichas (FP) y las instancias negativas correctamente predichas (VN). . . . 42
4.5 Validación cruzada con $k = 4$ para un solo modelo de clasificación. 43

5.1 Taxonomía de algoritmos de optimización global. 47
5.2 Representación del espacio de decisión y el objetivo correspondiente [38]. . . 51
5.3 Ejemplo de un frente de Pareto. 52
5.4 Representación gráfica del punto ideal y punto nadir. 53
5.5 Taxonomía de métodos de optimización multi-objetivo. 54

6.1 Codificación de una solución en μ-DEMS. 62
6.2 Decodificación de una solución en μ-DEMS. 63
6.3 Prueba post-hoc de Bonferroni utilizando (a) los mejores valores y (b) los valores de las medianas del conjunto de bases de datos. 68
6.4 Gráficas de convergencia para las bases de datos Beef y Coffee. 69
6.5 Gráficas de convergencia para las bases de datos ECG200 y FaceFour. 70
6.6 Gráficas de convergencia para las bases de datos Gun Point y OlieOil. 71
6.7 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 74
6.8 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 75

7.1 Metodología general del enfoque de FMS para bases de datos temporales. . . 77
7.2 Representación gráfica de un vector solución usado en la codificación real y mixta. 79
7.3 Representación gráfica de la codificación binaria. 79
7.4 Ejemplo de vecino generado por el operador de LS en codificación binaria. . 81
7.5 Ejemplos de vecino generado por el operador de LS en codificación mixta. . . 82
7.6 Prueba post hoc de Tukey utilizando (1) los mejores valores y (2) los valores en las medianas sobre todo el conjunto de bases de datos. 85
7.7 Gráficas de convergencia para las bases de datos Beef y CBF. 90
7.8 Gráficas de convergencia para las bases de datos Coffee y ECG200. 91
7.9 Gráficas de convergencia para las bases de datos FaceFour y GunPoint. 92
7.10 Gráficas de convergencia para las bases de datos Lightning2 y Lightning7. . . 93
7.11 Gráficas de convergencia para las bases de datos OliveOil y Trace. 94
7.12 Gráfica de diversidad global, donde se promedió cada una de las iteraciones de todas las bases de datos. 96

7.13 Análisis de frecuencia de los métodos que fueron considerados en el proceso de búsqueda de las metaherísticas. Donde S1-S4 representan los métodos de suavizado, R1-R3 corresponde a los métodos de representación de series de tiempo, NR1-NR2 son los métodos de reducción de numerosidad y C1-C6 representan los clasificadores. 100

8.1 Representación gráfica del enfoque multi-objetivo N2SFMS. 103
8.2 Representación gráfica de estrategias para el manejo de preferencias. 108
8.3 Prueba Post-hoc de Bonferroni, usando los mejores valores del conjunto de base de datos. 111
8.4 Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos Beef y CBF. 113
8.5 Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos GunPoint y Lightning-2. 114
8.6 Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos Lightning-2 y OliveOil. 115
8.7 Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos Synthetic_control y Trace. 116

10.1 Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 123
10.2 Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 124
10.3 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 124

10.4 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 125

10.5 Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 125

10.6 Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 126

10.7 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 126

10.8 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 127

10.9 Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 127

10.10 Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 128

10.11 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 128

10.12 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 129

10.14 Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 129

10.13 Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 130

10.15 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 130

10.16 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 131

10.17 Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 131

10.18 Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 132

10.19 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 132

10.20 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 133

10.21 Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 133

10.22 Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 134

10.23 Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 134

10.24 Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 135

11.1 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 136

11.2 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 137

11.3 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 137

11.4 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 138

11.5 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 138

11.6 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 139

11.7 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 139

11.8 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 140

11.9 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 140

11.10 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 141

11.11 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 141

11.12 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 142

11.13 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 142

11.14 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 143

11.15 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 143

11.16 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 144

11.17 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 144

11.18 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 145

11.19 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 145

11.20 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 146

11.21 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 146

11.22 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 147

11.23 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 147

11.24 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 148

11.25 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 148

11.26 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 149

11.27 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 149

11.28 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 150

11.29 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 150

11.30 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 151

11.31 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 151

11.32 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 152

11.33 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 152

11.34 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 153

11.35 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 153

11.36 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 154

11.37 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 154

11.38 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 155

11.39 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 155

11.40 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 156

11.41 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 156

11.42 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 157

11.43 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 157

11.44 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 158

11.45 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 158

11.46 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 159

11.47 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 159

11.48 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 160

11.49 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 160

11.50 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 161

11.51 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 161

11.52 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 162

11.53 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 162

11.54 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 163

11.55 Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 163

11.56 Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 164

11.57 Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 164
11.58 Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 165
11.59 Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias. 165
11.60 Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias. 166

Índice de tablas

6.1 Descripción de métodos incluidos en el espacio de búsqueda explorado por $\mu - DEMS$. TLS significa longitud de serie de tiempo, S: suavizado, R: representación y C: clasificación. 62

6.2 Variantes de DE usadas en los experimentos, $rand_j \in [0,1]$ es un número real aleatorio, $j_{rand} \in [1,D]$ es un número entero aleatorio . $\vec{x}_{best,g}$ es el mejor vector de la generación g. 65

6.3 Características de las bases de datos de series de tiempo utilizadas en experimentos con μ-DEMS. 65

6.4 Configuración del equipo de cómputo utilizado en cada experimento. 65

6.5 Configuración de parámetros utilizada por μ-DEMS. 66

6.6 Resultados estadísticos de CVER (B: mejor, M: media, W: peor, SD: Desviación estándar, p: p-valor) obtenidos por cada variante de μ-DEMS. Los valores en **negritas** representan los mejores valores encontrados. 67

6.7 Modelos sugeridos por cada variante de μ-DEMS. 73

7.1 Descripción de los métodos considerados y los hiperparámetros utilizados. TLS significa la longitud de la serie de tiempo. 78

7.2 Descripción de bases de datos. CS significa separación entre clases, NL significa nivel de ruido y SBS significa similitud entre el conjunto de entrenamiento y prueba. 83

7.3 Configuración de parámetros de las metaheurísticas. I significa iteraciones, NP es el tamaño de la población, CR es la taza de cruza, F es el factor de escalamiento, N es el número de soluciones a reiniciar, R es la frecuencia de reemplazos y N_k es el tamaño del vecindario. 84

7.4 Configuración del equipo de cómputo utilizado en cada experimento. 84

7.5 Mejores valores y medianas de cada metaheurística. Donde B: significa mejor valor y M: significa valor de la mediana. 86

7.6 Comparación múltiple del rendimiento promedio de las seis versiones de metaheurísticas por cada base de datos. Los valores a la derecha del símbolo ± representa la desviación estándar y los valores entre paréntesis representan los rangos calculados por la prueba de Friedman. Los valores en **negritas** representan los valores más bajos encontrados o el mejor ranking. 87

7.7 Comparación por pares entre la metaheurística de control S-LSMS1 y el resto de las metaheurísticas. (-) significa que se presentó una diferencia significativa a favor de la metaheurística de control. (+) implica que hubo una diferencia significativa a favor de la metaheurística comparada. (=) significa que no presentó diferencia significativa entre las metaheurísticas comparadas. Los valores en **negritas** representan los mejores valores encontrados. 88
7.8 Mejores modelos finales de las bases de datos Beef, CBF, Coffee, ECG200 y FaceFour obtenidos por cada meta-heurística. 97
7.9 Mejores modelos finales de las bases de datos Gun_Point, Lightning-2, Lightning-7, OliveOIl y Trace. 98

8.1 Descripción de las bases de datos usadas en los experimentos del enfoque multi-objetivo. 104
8.3 Configuración de parámetros usada por N2FMS. N es el tamaño de la población, G el número de generaciones, CR el porcentaje de cruza y MR el porcentaje de mutación. 109
8.2 Configuración del equipo de cómputo utilizado en cada experimento. 109
8.4 Resultados numéricos del modelo construido a partir de las estrategias de N2FMS para cada base de datos. Los valores entre paréntesis representan la desviación estándar y los valores en **negritas** significa el mejor valor encontrado. 110
8.5 Comparación entre S-N2FMS y μ-DEMS. Los valores entre paréntesis representan la desviación estándar y los valores en **negrita** representan los mejores valores encontrados. 112

I

Introducción

1

INTRODUCCIÓN

Antes de convencer al intelecto, es imprescindible tocar y predisponer el corazón.

—B. Pascal

1.1. MOTIVACIÓN

De unos años a la fecha, el incremento en la generación y uso de datos temporales, especialmente, series de tiempo [7, 29, 186], ha motivado el desarrollo de nuevas formas para la extracción de conocimiento útil a través del proceso de descubrimiento de conocimiento (KDD, *Knowledge Discovery in Databases*) y técnicas de minería de datos (DM, *Data Mining*).

Las series de tiempo, son una clase de datos temporales que representan una colección de valores obtenidos a partir de mediciones secuenciales realizadas a lo largo del tiempo [50]. Las principales características de las series de tiempo son la alta dimensionalidad, presencia de ruido aleatorio y la fuerte dependencia de sus variables [60].

En el área de quimiometría, las series de tiempo suelen ser útiles para representar información que es obtenida a partir de espectrometria infrarroja para clasificar alimentos. En la Figura 1.1 se muestran dos series de tiempo que son el resultado de un análisis de la composición química de los granos de café molido de dos especies (Robusta y Arábiga). Los valores que forman a las series de tiempo, son valores numéricos que equivalen a niveles de absorbancia de los espectros obtenidos de un grano de café de una especie determinada en un intervalo del espectro.

Por lo tanto, una serie de tiempo se define por su naturaleza numérica y continua, y a menudo es considerada como un todo en lugar de atributos numéricos independientes como en las bases de datos estáticas (bases de datos instantáneas, ver Sección 3.2), donde puede existir uno o varios atributos que muestran una vista en el tiempo y no en instantes de tiempo como en las series de tiempo.

La mayoría de métodos que se han propuesto para el procesamiento y análisis de datos, están hechos para tratar con datos no dependientes. Cuando estas técnicas son aplicadas a

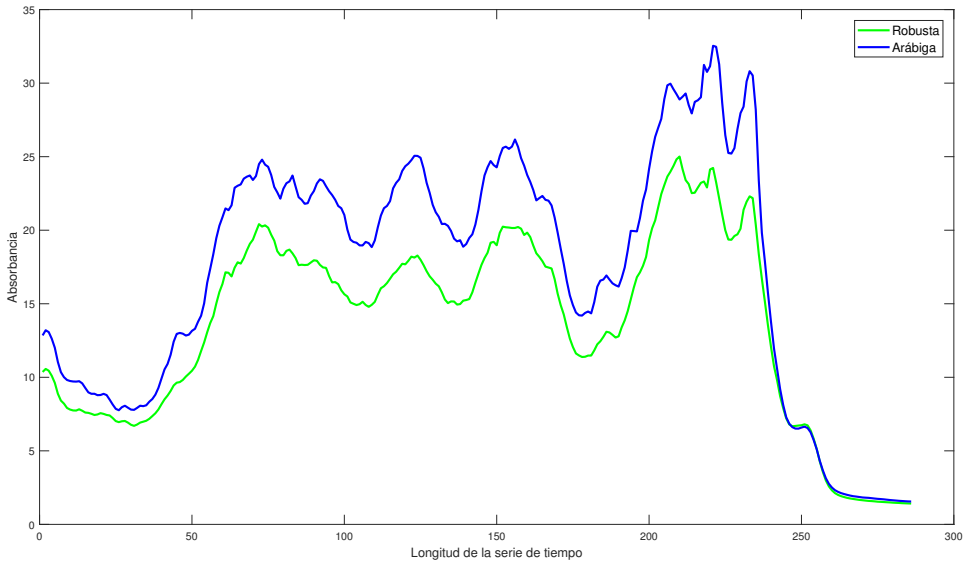


Figura 1.1: Ejemplo de dos series de tiempo que representa el espectro de una variedad diferente de café.

series de tiempo, el resultado puede ser deficiente o inadecuado [62]. En el área de DM surge una sub-área que se denomina minería de datos temporal (TDM, *Temporal Data Mining*) que involucra una gran variedad de métodos para llevar a cabo diferentes tareas con series de tiempo como: pre-procesamiento, consulta por contenido, segmentación, agrupación, clasificación, entre otras [7, 108].

Cuando las series de tiempo son involucradas en un proceso de minería de datos, la calidad de los patrones minados puede depender de dos cuestiones importantes; la primera es la elección del método apropiado para una determinada tarea [197], mientras que la segunda es la selección adecuada de hiperparámetros [1] que puedan producir un rendimiento relativamente bueno.

La selección de un método para resolver una determinada tarea es conocida como selección del algoritmo (AS, *Algorithm Selection*) [144], mientras que a la selección del modelo se refiere al problema de involucrar un conjunto de datos para seleccionar un modelo (con sus hiperparámetros) a partir de una lista de modelos candidatos (MS, *Model selection*) [181]. Encontrar

De acuerdo al Teorema del *No-Free-Lunch* [187], no existe un algoritmo de aprendizaje universal. Por lo tanto, la tarea de encontrar el mejor método y sus respectivos hiperparámetros que funcionen para cualquier conjunto de datos resulta ser una tarea muy compleja.

En la literatura relacionada, para resolver los problemas de AS y MS, se han propuesto reglas generales, copiar métodos y configuraciones que han funcionado para otros problemas o buscar la mejor opción a través de prueba y error. En la mayoría de los casos, ambos proble-

[1] Un hiperparámetro es una configuración externa al modelo de aprendizaje y cuyo valor no puede estimarse a partir de los datos de entrenamiento [67]. Por ejemplo el tipo de kernel en máquinas de soporte vectorial, el valor de K en vecinos más cercanos o la profundidad del árbol en un árbol de decisión.

mas se han tratado de forma separada. Sin embargo, pueden ser tratados de manera simultánea a través del concepto de selección del modelo completo (FMS, *Full Model Selection*) [45]. FMS se puede definir como un problema que consiste en encontrar una combinación adecuada de métodos y su respectiva configuración de hiperparámetros para tratar un conjunto de datos en particular, donde diferentes tareas de KDD y DM pueden estar involucradas.

Motivados por lo anterior, en este trabajo se presenta un enfoque que se clasifica dentro de la categoría de Envoltorio Evolutivo, descrita en el Capítulo 2, para dar solución al problema de FMS en bases de datos temporales. Se explora el uso de metaheurísticas de optimización global mono-objetivo a través de metaheurísticas tanto poblacionales como de un solo punto. Además se extiende la propuesta para la optimización de dos objetivos a través del algoritmo NSGA-II. Tanto el enfoque mono-objetivo como el multi-objetivo involucran cuatro tareas importantes dentro de la TDM que son: la reducción de ruido por medio de métodos de suavizado, la reducción de dimensionalidad de las series de tiempo, reducción de numerosidad y el aprendizaje automático por medio de la tarea de clasificación. Se busca aportar una herramienta eficiente para la selección de modelos completos para el tratamiento de datos temporales así como la configuración automática de los hiperparámetros relacionados.

1.2. PLANTEAMIENTO DEL PROBLEMA

El problema de FMS es un término propuesto por Escalante [45], que consiste en seleccionar una combinación de métodos adecuados que pueden realizar diferentes tareas de minería de datos, además de la optimización automática de los hiperparámetros relacionados. El problema de FMS en bases de datos temporales puede ser tratado como un problema de optimización mono-objetivo o multi-objetivo, donde en ambos enfoques se buscan modelos completos que involucren una combinación adecuada de métodos de suavizado de ruido, reducción de la dimensionalidad de las series de tiempo, reducción de numerosidad y clasificación, además de la automática configuración de cada uno de los hiperparámetros. La Figura 1.2 representa de manera gráfica el problema de FMS en bases de datos temporales, donde existe una base de datos temporales que entra a un proceso formado de cuatro tareas (suavizado, representación, reducción de datos y clasificación) en el que existe un subconjunto de métodos con sus respectivos hiperparámetros. La salida del proceso es una solución o varias que posea una combinación adecuada de métodos y sus hiperparámetros optimizados.

Una instancia del problema de FMS en bases de datos temporales se presenta en la Tabla 7.1, donde se presentan los métodos de TDM utilizados para el tratamiento de datos temporales.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Diseñar e implementar un asistente de búsqueda de modelos completos útiles y competitivos, de extremo-a-extremo[2], que involucre simultáneamente la optimización de hiperparámetros y la selección apropiada de un conjunto de métodos para el procesamiento y

[2]Modelos de aprendizaje que involucran diversas tareas que son llevadas a cabo una después de la otra, coloquialmente conocidas como soluciones de tubería.

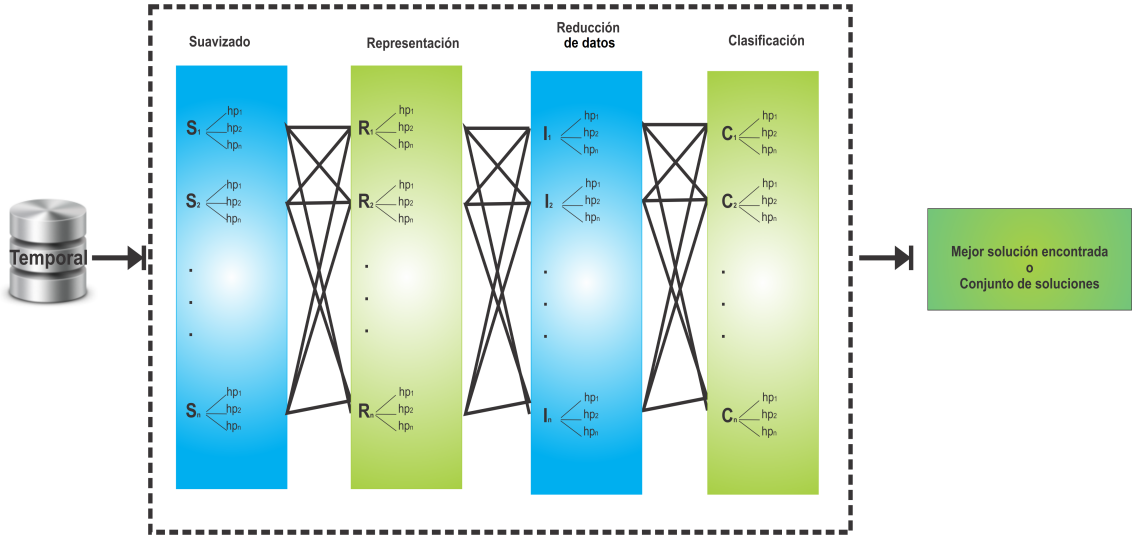


Figura 1.2: Representación gráfica del problema de FMS.

clasificación de datos temporales, a través de metaheurísticas poblacionales y de trayectoria.

Para lograr el objetivo general se define una serie de objetivos específicos que se muestran a continuación:

1.3.2. OBJETIVOS ESPECÍFICOS

1. Revisar y conocer el estado del arte sobre el problema de FMS.
2. Explorar qué algoritmos de optimización global y multi-objetivo han sido utilizados para resolver el problema de FMS.
3. Establecer las tareas primordiales que pueden ser incluidas como parte de un modelo completo para bases de datos temporales.
4. Diseñar, implementar y evaluar un enfoque evolutivo de un solo objetivo que resuelva el problema de FMS para bases de datos temporales.
5. Diseñar, implementar y evaluar un enfoque evolutivo multi-objetivo que resuelva el problema de FMS para bases de datos temporales.
6. Proponer estrategias de selección de preferencias para elegir una solución de entre un conjunto de soluciones proporcionadas por el enfoque multi-objetivo propuesto.

1.4. HIPÓTESIS

La principal hipótesis que persigue este trabajo se describe a continuación:

La selección automática de un modelo completo para clasificar bases de datos temporales logrará un desempeño competitivo mediante la asistencia de metaheuristicas poblacionales y de trayectoria, con la finalidad de minimizar la tasa de error en clasificación y el costo de ejecución derivado de la cantidad de híperparámetros a optimizar.

1.5. JUSTIFICACIÓN

Este trabajo de investigación se centra en tres aspectos que sustentan el desarrollo del enfoque propuesto, que son: (1) minería de datos libre de parámetros, (2) asistencia en la mejora de la calidad del conocimiento de expertos humanos y (3) supervivencia del conocimiento obtenido. Estos aspectos se describen a continuación.

1. *Minería de datos libre de parámetros.* La mayoría de los métodos propuestos para el minado de datos suelen tener un conjunto de parámetros que requieren ser ajustados. En minería de datos temporales surge el paradigma de minería de datos libre de parámetros [83], donde la idea principal es el diseño de algoritmos que no tienen parámetros y que son universalmente aplicables en todas las áreas. Con la elaboración de este trabajo de investigación se propone el uso de metaheurísticas para la selección automática de una serie de métodos adecuados, con sus hiperparámetros relacionados, para múltiples tareas de minería de datos temporal. A partir de dicha contribución, ofrecer un nuevo panorama a la comunidad de minería de datos temporal sobre la forma de ajuste de los métodos, a través de una herramienta basada en una metaheurística, sin necesidad de quitar los hiperparámetros originales de los métodos, además de poder obtener un modelo que involucre múltiples tareas.

2. *Asistir en la mejora de la calidad del conocimiento de expertos humanos.* En este punto se pretende ofrecer soporte a expertos humanos para construir conocimiento útil. En la mayoría de los dominios, el proceso de adquisición de conocimiento suele ser costoso debido a que los ingenieros del conocimiento no poseen los mismos conocimientos que el experto, por lo que la comunicación entre experto e ingeniero suele ser lenta. Por lo tanto, el experto humano con el uso de herramientas automáticas que faciliten el minado de datos, se entrena en el uso de herramientas asistentes para obtener modelos con características que él espera.

3. *Conseguir la supervivencia del conocimiento.* Se dice que la transformación de las sociedades modernas en sociedades del conocimiento representa uno de los principales cambios en la estructura de las economías de los países avanzados y en desarrollo. Ya que la fuente del crecimiento económico y las actividades que aportan valor, se basan cada vez más en el conocimiento [165]. El conocimiento puede ser producido por diferentes agentes humanos como gerentes empresariales, investigadores, estudiantes y/o individuos proactivos. La adquisición del conocimiento lleva tiempo y a menudo se basa en capacidades y habilidades cognitivas intermedias [185]. Se dice que el conocimiento científico y técnico es especialmente importante en los sistemas sociales modernos por el soporte a la toma de decisiones [165]. Por lo tanto la supervivencia y conservación del conocimiento es un aspecto de gran importancia. En ausencia del personal especializado en algún dominio, surge la necesidad de contar con alguna herramienta que facilite la tarea de obtener conocimiento útil sin la necesidad de tener presente físicamente al experto, ayudando en la supervivencia del conocimiento [116, 174]. Cabe aclarar que no se pretende desplazar a los expertos, más bien se pretende aportar una herramienta que sirva de apoyo en la obtención de información valiosa a partir de un conjunto de datos donde un experto puede o no estar presente.

1.6. CONTRIBUCIONES

- Una herramienta de búsqueda de modelos completos para bases de datos temporales, basada en una variante de micro-Evolución Diferencial (μ-DE, *Differential Evolution*), capaz de encontrar soluciones competitivas en un número reducido de generaciones.

- Una herramienta de búsqueda de modelos completos libre de parámetros, basada en la estructura de un buscador local con representación mixta o binaria.

- Una herramienta de búsqueda basada en un enfoque multi-objetivo, capaz de encontrar modelos completos con un costo computacional bajo y un rendimiento competitivo, además de proporcionar de forma visual el resultado del mejor modelo encontrado para un conjunto de datos en particular.

Como resultado de la realización de este trabajo de tesis, se derivan algunas publicaciones que se listan a continuación:

- Congresos
  - Pérez-Castro, N., Acosta-Mesa, H.G., Mezura-Montes, E., & Cruz-Ramírez, N., 2015. *Towards the Full Model Selection in Temporal Databases by Using Micro-Differential Evolution, An Empirical Study*. En Power, Electronics and Computing (ROPEC), 2015 IEEE International Autumn Meeting on. IEEE, 2015. p. 1-6.
  - Pérez-Castro, N., Acosta-Mesa, H. G., Mezura-Montes, E., & Escalante, H. J. (2016, November). *Multi-objective Full Model Selection in temporal databases: Optimizing time and performance*. En Power, Electronics and Computing (ROPEC), 2016 IEEE International Autumn Meeting on (pp. 1-6). IEEE.
  - Pérez-Castro, N., Márquez-Grajales, A., Acosta-Mesa, H. G., & Mezura-Montes, E. (2017, June). *Full Model Selection issue in temporal data through evolutionary algorithms: A brief review*. En Evolutionary Computation (CEC), 2017 IEEE Congress on (pp. 2451-2457). IEEE.

- Revista
  - Pérez-Castro, N., Acosta-Mesa, H.G., Mezura-Montes, E., Cruz-Ramírez, N., 2020. *Full Model Selection Problem in Temporal Databases: contrasting population-based and single point search metaheuristics*. Revista Ingeniería e Investigación Universidad Nacional de Colombia. ***Estatus: Sometido***.
  - Pérez-Castro, N., Acosta-Mesa, H.G., 2020. Towards a Transfer Learning Strategy in Full Model Selection Algorithm for Temporal Data Mining. *Research in Computing Science*, 149 (2020): 65-73.

1.7. ESTRUCTURA DEL DOCUMENTO

Este documento está estructurado en cuatro partes, incluida la parte introductoria. Cada una de las partes con sus respectivos capítulos se describen de forma general a continuación.

- Parte I, corresponde a la parte introductoria de este documento.

  - Capítulo 1: Hace una breve introducción hacia el tema en cuestión, así como la definición de hipótesis, objetivos y justificación.
  - Capítulo 2: Se presenta la revisión de la literatura.

- Parte II, se compone de tres capítulos que definen conceptos básicos así como la descripción de los métodos utilizados:
  - Capítulo 3: Describe los conceptos básicos sobre la minería de datos temporal.
  - Capítulo 4: Muestra un marco teórico acerca del aprendizaje automático, especialmente de la tarea de clasificación.
  - Capítulo 5: Proporciona un marco de referencia sobre los conceptos de optimización global y multi-objetivo.

- Parte III, se compone de 3 capítulos que proporcionan la descripción de las contribuciones realizadas a partir de este trabajo de tesis.
  - Capitulo 6: Presenta un estudio empírico donde se comparan diferentes variantes del algoritmo de evolución diferencial en su versión micro para resolver el problema de FMS en bases de datos temporales.
  - Capítulo 7: Presenta la comparación de un enfoque basado en una metaheurística basada en población y una metaheurística basada en un solo punto para resolver el problema de FMS en bases de datos temporales.
  - Capítulo 8: Presenta la evaluación de un enfoque multi-objetivo para resolver el problema de FMS en bases de datos temporales, donde se usan como funciones objetivo el error de clasificación del modelo y el tiempo de ejecución que tarda dicho modelo en ser generado.

- Parte IV, describe las conclusiones finales así como el trabajo futuro.

2

REVISIÓN DE LA LITERATURA

> La educación genera confianza. La confianza genera esperanza. La esperanza genera paz.
>
> —Confucio

En este capítulo se describen algunos trabajos relevantes que aportan información útil sobre la forma en que se le ha dado solución al problema de FMS. La revisión presentada se divide en tres categorías que muestran de qué manera se ha tratado de resolver el problema de FMS y sus sub-problemas (AS y MS).

(1) Propuestas basadas en meta-aprendizaje (MT, *Meta-Learning*).

(2) Las propuestas basadas en un método embebido (EM, *Embedded Method*).

(3) Enfoques de envoltorio evolutivo (EW, *Evolutionary Wrapper*).

2.1. PROPUESTAS BASADAS EN META-APRENDIZAJE (MT)

Los primeros intentos por seleccionar el algoritmo más adecuado para determinadas tareas, fue a través del concepto de meta-aprendizaje. El término de MT comienza a ser utilizado dentro de la comunidad de aprendizaje automático y tiene el objetivo de descubrir formas de buscar dinámicamente la mejor estrategia de aprendizaje a medida que aumenta la cantidad de tareas [178]. A continuación se listan y se describen algunos trabajos relevantes.

- Uno de los primeros esfuerzos de caracterizar un problema donde se involucra la tarea de clasificación y así poder examinar su impacto en el comportamiento de un conjunto de algoritmos, fue el trabajo de Rendell y Chao (1990) [142]. En esta propuesta los autores mencionan que la visualización de conceptos como funciones sobre el espacio de las instancias de un conjunto de datos conduce a la formación de características geométricas como la proporción de instancias positivas o la concentración. De acuerdo a sus experimentos ellos muestran que algunas de las caracterizaciones encontradas en los datos afectan drásticamente la precisión del aprendizaje conceptual.

- Las ideas de Rendel y Cho fueron extendidas por Aha (1992) [1], quien presenta un estudio empírico donde propone la creación de reglas que sean capaces de describir cuándo ciertos algoritmos superan significativamente a los demás en algunas medidas dependientes. Las características que fueron consideradas en este enfoque son: número de clases, número de instancias, prototipo por clases, número de atributos relevantes e irrelevantes y el rango de la distribución tanto de las instancias como de los prototipos.

- Posteriormente, Brodley (1993) [18] se basa en las ideas de Aha y sugiere una búsqueda más dinámica para seleccionar el mejor algoritmo. Este autor propone el uso de reglas para reconocer cuándo un algoritmo no está funcionando correctamente y así seleccionar una mejor opción durante la etapa de ejecución.

- Un trabajo importante fue el de StatLog (1995) [84], que consistió en un conjunto de pruebas comparativas de aprendizaje estadístico y lógico para tareas de clasificación. El principal objetivo de StatLog fue proporcionar una evaluación objetiva de las fortalezas y debilidades de diversos enfoques de clasificación. El análisis de los resultados obtenidos de esta prueba aportan información relevante acerca de la relación que existe entre el rendimiento de los algoritmos y la caracterización de los conjuntos de datos. En esta evaluación fue posible evaluar el comportamiento de 23 algoritmos de aprendizaje automático y se usaron 22 bases de datos estáticas, donde fue posible sacar 17 características diferentes de cada base de datos.

- Después de StatLog, otra propuesta importante fue la denominada como NOEMON (1999) [77], donde se propone el uso de algoritmos en pares para la creación de modelos de meta-aprendizaje. El par de algoritmos era capaz de recomendar un algoritmo sobre el otro o declarar un empate basado en pruebas estadísticas significativas. Cada algoritmo aprendiz tenía la tarea de personalizar la caracterización de los datos para garantizar la relevancia del par de algoritmos, y posteriormente obtener una recomendación donde se combinaban los resultados de los meta-datos individuales.

- Las propuestas anteriores presentaban el problema de que en la mayoría de las veces el esfuerzo en el cálculo de la caracterización de los datos era mayor que la ejecución de un algoritmo simple. Posteriormente se plantea una nueva dirección basada en saber predecir el rendimiento de diferentes algoritmos en relación al rendimiento de los mismos. Esta nueva dirección fue denominada como enfoque *Landmarking* (2000) [124]. Este enfoque intenta determinar la ubicación de un problema de aprendizaje específico en el espacio de todos los problemas de aprendizaje a través de la medición directa del rendimiento de los algoritmos. Los autores de este enfoque muestran cómo el uso de los valores históricos pueden ayudar a distinguir áreas del espacio de aprendizaje que favorecen a diferentes algoritmos aprendices.

- De acuerdo a las ideas del proyecto StatLog y al enfoque Landmarking, surge un nuevo proyecto denominado como METAL (2013) [64]. Dicho proyecto es concebido como un asistente de meta-aprendizaje para brindar soporte a usuarios de aprendizaje automático y minería de datos. Este asistente incluyó la tarea de regresión además de la

clasificación, teniendo como principal objetivo el desarrollo de métodos para la selección del modelo así como la combinación de métodos a través de un entorno en línea.

- Después de StatLog y METAL, se observa una serie de estudios regidos por las ideas de estos enfoques, donde sólo se realizan extensiones en el número de algoritmos y bases de datos, como el caso de Lim (2000) [94] y Smith (2002) [163] o agregando nuevos elementos de caracterización o medidas de desempeño [129]. Después de esta ola de propuestas surgen otras que se enfocan en una sola clase de algoritmos de clasificación con el propósito de ajustar sus parámetros y/o hiperparámetros. Tal es el caso de Ali y Smith-Miles (2006) [3], quienes se enfocan en encontrar la mejor función *kernel* para ser usada por máquinas de soporte vectorial.

- En relación al uso de datos temporales, es posible encontrar algunos enfoques que extienden las ideas de StatLog o METAL, que se enfocan en el dominio de *forecasting*. Tal es el caso de Arinze (1997) [5], quien se basa en StatLog para la generación de reglas para la selección de métodos de forecasting.

- Venkatachalam y Sohl (1999) [176], propusieron un enfoque basado en una red neuronal de dos etapas, donde en la primera etapa se determina cual de los 9 métodos disponibles, agrupados en tres grupos, es el más indicado para los datos. Una vez que se determina qué grupo es el más adecuado para una serie de tiempo, se aplica la segunda red neuronal para determinar cual de los 3 algoritmos dentro de un grupo es probable que produzca el error de pronóstico más pequeño.

- En el caso de Prudêncio y Ludermir (2004) [131] hicieron uso del enfoque de NOEMON, evaluaron diferentes series de tiempo y usaron 3 algoritmos para *forecasting*[1].

2.2. PROPUESTAS BASADAS EN MÉTODOS EMBEBIDOS (EM)

En esta categoría se sitúan propuestas que se enfocan en la selección del modelo para una determinada clase de algoritmos de aprendizaje. Los enfoques basados en EM requieren de una estrategia de búsqueda y de una función de evaluación. La principal característica de este tipo de propuestas es que logran explotar las características específicas de la arquitectura del algoritmo de aprendizaje para realizar inferencias de varios niveles, por lo tanto se requiere conocer a detalle la naturaleza y estructura del algoritmo que se va a tratar [67]. Los enfoques EM han llamado la atención dentro de la comunidad de aprendizaje automático debido a la elegancia matemática de algunos de los métodos propuestos. A continuación se describen algunos trabajos representativos:

- Uno de los enfoques más utilizados en esta categoría es la inferencia multi-nivel. Ésta se puede definir como un problema de aprendizaje organizado en una jerarquía de problemas de aprendizaje. Tal es el caso de la propuesta de Bennet (2006) [9], quien propuso un enfoque de programación bi-nivel para máquinas de soporte vectorial con el objetivo de hacer regresión. El autor divide un determinado conjunto de datos en conjuntos de entrenamiento y prueba, y posteriormente se busca un conjunto de hiperparámetros. En el primer nivel se trata de resolver el problema de entrenamiento

[1] Tarea para llevar acabo pronósticos basados en el análisis de tendencias.

óptimo para cada grupo de entrenamiento, mientras que en el segundo nivel se intenta reducir la pérdida sobre los conjuntos de datos de prueba. El problema de optimización resultante es un programa de dos niveles, donde cada función de aprendizaje es optimizada en su problema de entrenamiento correspondiente con determinados hiperparámetros.

- En el contexto bayesiano, se han propuesto enfoques embebidos donde el muestreo de hiperparámetros se maneja de forma matemática. Sin embargo, resulta complejo encontrar métodos que puedan tratar este tipo de formulaciones. Una de las principales aportaciones que se destacan en este ámbito es el trabajo de Boullé (2007) [13], quien propuso métodos de selección del modelo para clasificación y regresión que se caracterizan por ser completamente bayesianos, pero hacen uso de los antecedentes previos dependientes de los datos.

- Kunapul y otros (2008) [90] proponen la integración de la búsqueda de hiperparámetros en un programa de optimización de múltiples niveles. Este enfoque fue propuesto especialmente para la selección del modelo en máquinas de soporte vectorial.

2.3. ENFOQUES DE ENVOLTORIO EVOLUTIVO (EW)

A diferencia de las dos categorías anteriores, los enfoques de envoltorio consideran a los algoritmos de aprendizaje como cajas negras donde es posible ajustar internamente los hiperparámetros a partir de la entrada de un determinado conjunto de datos. En este tipo de enfoques no es necesario tener conocimiento de la arquitectura de los métodos o de los datos. Al igual que los enfoques embebidos, los envoltorios necesitan de alguna estrategia de búsqueda para explorar el espacio de los hiperparámetros. Sin embargo, en los enfoques de envoltorio es posible hacer una selección de una combinación de métodos para diversas tareas al mismo tiempo que se van optimizando sus hiperparámetros. Algunas de las estrategias de búsqueda que se han utilizado en este tipo de propuestas son los algoritmos evolutivos (EAs, *Evolutionary Algorithms*) y los algoritmos de inteligencia colectiva (SIAs, *Swarm Intelligence Algorithms*), si se tiene una sola función de evaluación. Se han reportado enfoques donde se manejan dos funciones de evaluación, donde es posible utilizar algoritmos de optimización multi-objetivo como NSGA-II o MOEA/D.

- Uno de los primeros esfuerzos de investigación y del cual emerge el concepto de FMS, es el trabajo presentado por Escalante (2009) [45]. El autor sugiere el uso del algoritmo de cúmulo de partículas (PSO) como herramienta de búsqueda para la selección del modelo, el enfoque es nombrado PSMS (*Particle Swarm Model Selection*). En este enfoque cada modelo candidato es codificado como una partícula donde es posible integrar diversos métodos para realizar tareas como preprocesamiento, selección de características y algoritmos de aprendizaje; la selección del modelo incluye la selección de los hiperpámetros relacionados. El autor hace hincapié en el uso de PSO en lugar de EAs debido a la simplicidad, rápida convergencia y generalidad para aplicarlo. A pesar de que la propuesta está limitada a trabajar con bases de datos estáticas binarias, el enfoque reporta resultados competitivos con bases de datos de referencia (*benchmark*) [44], así como en la solución de problemas como la verificación de autoría de textos [43, 46] y detección de enfermedades como la leucemia [48].

- A partir de la primera versión de PSMS, Escalante (2010) [47] incorpora la construcción de ensambles de clasificación binaria a través de modelos obtenidos de PSMS. Se menciona que muchos de los modelos que son potencialmente útiles, para el problema de clasificación, no son considerados para la construcción del modelo final. Por lo tanto, se propone aprovechar la variedad de modelos evaluados por PSMS para construir ensambles de clasificación que sean capaces de superar la mejor solución encontrada por PSMS original. El autor explora tres formulaciones para construir ensambles a partir de soluciones intermedias de PSMS, donde la mejor estrategia para la construcción del ensamble fue considerar la colección de modelos que obtuvieron el mejor rendimiento en cada iteración, sin importar si superaban o no a la mejor solución global.

- Inicialmente la propuesta de PSMS estaba limitada a problemas de clasificación binaria, posteriormente Escalante (2012) presenta una extensión del enfoque para la clasificación multi-clase con el propósito de dar solución al problema de anotación automática de imágenes [49]. En esta extensión del enfoque se incorpora la estrategia de uno contra todos (OVA, *one-vs-all*), dicha estrategia es capaz de convertir un problema multi-clase en un conjunto de problemas de clasificación binaria.

- Otro trabajo representativo es el propuesto por Sun (2013) [168], donde se presenta un marco de referencia para el diseño de algoritmos que resuelvan el problema de FMS. Los autores proponen un enfoque basado en la combinación de un algoritmo genético (GA, *Genetic Algorithm*) y el algoritmo PSO, dicha combinación es nombrada como GPS en la cual el GA es utilizado para buscar la estructura óptima de una solución de minería de datos, mientras que PSO es usado para buscar los hiperparámetros óptimos para una solución en particular. Dentro de las tareas que se consideran en esta propuesta son el muestreo de datos, limpieza de datos, transformación se características y selección de características para bases de datos estáticas.

- Dentro de los enfoques multi-objetivo se sitúa el trabajo de Rosales-Pérez (2014) [147], quien aborda el problema de FMS como un problema multi-objetivo para la tarea de clasificación de bases de datos estáticas. El error de entrenamiento y la complejidad del modelo (formulada a través de la dimensión de Vapnik-Chervonenkis) son adoptados como las funciones de evaluación. El algoritmo de búsqueda utilizado fue el algoritmo evolutivo multi-objetivo basado en descomposición (MOEA/D, *Multiobjective Evolutionary Algorithm Based on Decomposition*). Los resultados experimentales en los conjuntos de datos indican la efectividad del enfoque propuesto. Además, un estudio comparativo muestra que los modelos obtenidos son altamente competitivos, en términos de rendimiento de generalización.

Como se puede observar, la pluralidad de las propuestas ha generado la dificultad de una comparación justa entre los diversos enfoques. Recientemente, Shang et al. [159] muestran evidencia de dicho problema de comparación y proponen un punto de partida para futuros desarrollos a través de la democratización de las propuestas, es decir, establecer criterios en cuanto al tipo de procesos a considerar así como el tipo de medidas de evaluación.

En este capítulo se pudo analizar el panorama actual del problema se selección del modelo completo; se encontraron nichos de investigación como: proponer una herramienta para

la selección del modelo completo para bases de datos de series de tiempo, explorar alternativas para medir la complejidad de modelos, evaluar diferentes formas de representación de las soluciones.

II

Marco Teórico

3

MINERÍA DE DATOS TEMPORAL

> Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a donde van.
>
> —Leonardo Da Vinci

En este capítulo se presentan los conceptos básicos relacionados con minería de datos, aprendizaje automático y bases de datos temporales.

3.1. MINERÍA DE DATOS Y DESCUBRIMIENTO DE CONOCIMIENTO

La era actual de la información se caracteriza por el constante crecimiento de los datos que se generan en múltiples áreas [130]. Dicho crecimiento ha ocasionado una gran demanda de nuevas y poderosas herramientas para convertir datos en conocimiento útil, orientado a tareas [167].

Con el fin de cubrir dicha necesidad, los investigadores y profesionales han estado explorando ideas y métodos desarrollados en áreas como: el aprendizaje automático, el reconocimiento de patrones, análisis de datos estadísticos, visualización de datos e imágenes, entre otras. Todas estas áreas han llevado a la integración de dos importantes áreas de investigación que son la minería de datos (DM, *Data Mining*) y el descubrimiento de conocimientos en datos (KDD, *Knowledge Discovery in Data*).

Con frecuencia, KDD y DM son considerados como la misma cosa [63, 167], sin embargo, la principal diferencia entre KDD y DM es que, KDD se refiere al proceso general de descubrir conocimiento en datos, mientras que DM se refiere a la aplicación de algoritmos para extraer patrones de los datos sin los pasos adicionales del proceso de KDD.

En la Figura 3.1 se muestra de forma gráfica el proceso de KDD, que se compone de cinco etapas: selección de los datos, pre-procesamiento, transformación de los datos, minería de datos e interpretación/evaluación. El objetivo del proceso de KDD es obtener conocimiento útil a partir de los datos que se proporcionen como entrada.

La DM clásica puede ser definida como una etapa del proceso de KDD que consiste en

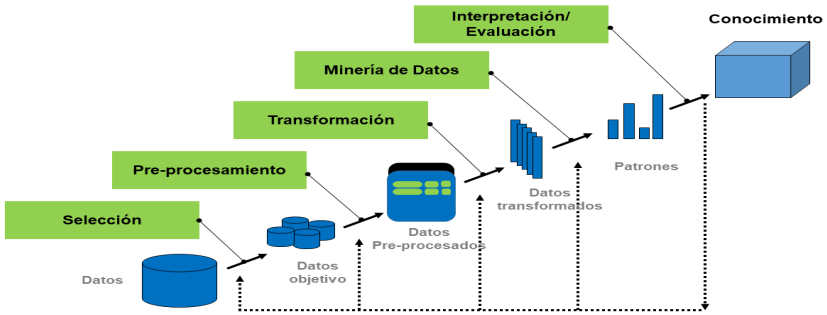


Figura 3.1: Proceso de descubrimiento de conocimiento en datos [53].

la búsqueda de patrones en los datos a partir del uso y aplicación de diferentes algoritmos [68]. A partir del área de DM se desprende otra importante área conocida como minería de datos temporal (TDM, *Temporal Data Mining*) que se refiere a la extracción de datos de grandes conjuntos de datos secuenciales que incorporan la variable tiempo [78, 155]. TDM tiene la capacidad de detectar actividades en lugar de sólo estados, es decir, permite encontrar aspectos de comportamientos en lugar de simplemente describir sus estados en un determinado momento. Se puede decir que TDM es más reciente que la clásica DM, por lo tanto sus restricciones y objetivos son diferentes. Entre las diferencias más importantes que se pueden encontrar entre DM y TDM se describen a continuación:

- Una diferencia importante radica en el tamaño y la naturaleza de los conjuntos de datos y la forma en que éstos son recolectados.
- La segunda gran diferencia radica en el tipo de información que se desea estimar u obtener de los datos. El alcance de TDM se extiende más allá del pronóstico estándar o el análisis de series de tiempo.
- Además, los parámetros exactos de un modelo final, pueden ser de poco interés en el contexto de DM clásica.

3.2. BASES DE DATOS TEMPORALES

Las bases de datos temporales son una clase de base de datos que se caracterizan por tener un sello de tiempo (time-stamping), que es usualmente aplicado a cada instancia o a cada atributo [108]. El sello de tiempo se puede realizar de la siguiente manera:

- Con *tiempo válido*, que es el momento en el cual la información del elemento es verdadera en el mundo real. Por ejemplo, *el cliente fue atendido el 19 de Febrero de 2017 a las 4:30 pm.*
- Con un *tiempo de transacción*, que es el momento en que la información del elemento es ingresada en la base de datos.
- *Bi-temporal*, hace referencia a la unión de las dos primeras, es decir, con un tiempo válido y un tiempo de transacción.

Las bases de datos que soportan tiempo pueden ser divididas en cuatro categorías:

- Base de datos instantáneas (*snapshot*): Guardan la versión más reciente de los datos, las bases de datos tradicionales entran en esta categoría.
- Bases de datos de reversión (*rollback*): Este tipo de base de datos solo soportan el concepto de tiempo de transacción.
- Bases de datos históricas (*historical*): Sólo soportan el concepto de tiempo válido.
- Bases de datos temporales (temporal): Soportan ambos conceptos, tiempo válido y de transacción.

De acuerdo con Mitsa [108], los datos temporales pueden ser clasificados en tres categorías: a) secuencias temporales, b) datos temporales semánticos y c) series de tiempo; siendo las últimas a las que se hará referencia en este documento.

3.2.1. SERIES DE TIEMPO

Las series de tiempo son una clase importante de objetos temporales, que pueden ser obtenidos de diversas fuentes y representan mediciones ordenadas de valores numéricos, usualmente reales, obtenidos en intervalos regulares de tiempo [50, 60].

Definición 1 *Una serie de tiempo* $\mathbf{t}$ *es una secuencia ordenada de valores reales representada a través de un vector j-dimensional [50], ver Ecuación 3.1.*

$$\mathbf{t} = (t_1, t_2, ..., t_j) \tag{3.1}$$

donde cada t_j *denota el j-ésimo elemento de la serie de tiempo* $\mathbf{t}$.

Resultando en un proceso subyacente en el cual se obtienen los valores de las mediciones realizadas en instantes de tiempo uniformemente espaciadas y de acuerdo con una tasa de muestreo determinada. Entonces, una serie de tiempo puede definirse como un conjunto de instantes de tiempo contiguos [8].

Sea $\boldsymbol{T}$ el conjunto de todas las series de tiempo consideradas, donde $\mathbf{t}_i \in \boldsymbol{T}$. A partir de $\boldsymbol{T}$ se pueden formar subconjuntos masivos de series de tiempo que se pueden almacenar en bases de datos etiquetadas, es decir, que contienen subgrupos denominados clases.

Definición 2 *Una base de datos etiquetada* $\boldsymbol{D} = \{(\mathbf{t}_i, c_i)\}$ *se define como un conjunto desordenado de n series de tiempo junto con sus etiquetas de clase* c_i *[50].*

Las series de tiempo pueden ser multivariadas o univariadas. Una serie multivariada se forma de varias series que abarcan simultáneamente múltiples dimensiones dentro del mismo rango de tiempo, mientras que una serie univariada es considerada como una sola variable subyacente [108]. Las series de tiempo resultan ser muy utilizadas en muchas aplicaciones como: análisis del mercado de valores, pronósticos económicos y de ventas, análisis presupuestarios, proyecciones de utilidades, proceso y control de calidad, observación de fenómenos naturales (atmosféricos, temperatura, viento, terremotos, etc.), experimentos científicos e ingenieriles, tratamientos médicos, entre otras [60, 108, 135].

3.3. PRE-PROCESAMIENTO DE DATOS TEMPORALES

Se dice que para que el proceso de minería de datos pueda producir resultados significativos, los datos deben ser pre-procesados apropiadamente para limpiarlos [78, 91, 108]. Dicho proceso de limpieza puede dirigirse hacia dos aspectos: el manejo de datos faltantes o la reducción de ruido. La presencia de ruido en las series de tiempo limita severamente la capacidad de extraer información útil [88]. El ruido se define como un error aleatorio que puede ocasionarse por varios factores como equipos de medición defectuosos y/o factores ambientales.

3.3.1. REDUCCIÓN DE RUIDO

La reducción de ruido es un término amplio donde el objetivo es eliminar de una serie de tiempo algún componente no deseado que a su vez puede estar compuesto por ruido [52, 108]. De una forma más precisa, sea $\mathbf{t}_i$ una serie de tiempo ruidosa, se asume que ésta puede descomponerse como en la Ecuación 3.2.

$$\mathbf{t}_i = \mathbf{y}_i + \epsilon_i \tag{3.2}$$

Donde $\mathbf{y}_i$ es la parte deseada de la serie de tiempo y ϵ_i es el ruido que se desea eliminar. Existe una gran diversidad de métodos para la reducción de ruido en series de tiempo, dichos métodos se pueden clasificar en cuatro categorías: graduación, predicción, filtrado y suavizado [92].

- Graduación. Supone que la señal ha finalizado, permitiendo así el uso de todos los datos que se utilizarán para reducir el ruido.
- Predicción. La predicción implica estimar los futuros valores libres de ruido usando datos antiguos.
- Filtrado. Los filtros estiman el ruido actual usando toda la información disponible.
- Suavizado. Los métodos de suavizado son idénticos a los filtros, con la excepción de un retraso adicional.

A continuación se describen los métodos de reducción de ruido que fueron utilizados en este trabajo.

PROMEDIOS MÓVILES

El método de promedios móviles (MV, *Moving Average*) corresponde a la categoría de los filtros de respuesta al impulso finito (FIR) [112]. Un filtro de promedio móvil, funciona promediando un número de puntos de una señal de entrada para generar cada punto en la señal de salida [66]. Dicho procedimiento se repite de tal manera que se hace uso de un intervalo o ventana de m puntos para calcular el promedio del conjunto de datos. MV es expresado en la Ecuación 3.3.

$$y_i = \begin{cases} \dfrac{\sum_{j=1}^{m}(t_{i,j})}{m} & \text{Si } i > 0 \wedge i < n - m \\ 0 & \text{de otra manera} \end{cases} \tag{3.3}$$

Donde $t_{i,j}$ es cada punto de una serie de tiempo de entrada, y_i es una serie de tiempo de salida, m es el número de puntos en la ventana y n es el total de puntos. La idea básica es que al promediar los valores de forma local, las componentes de frecuencia alta de la entrada serán promediadas y las variaciones de frecuencia baja serán mantenidas.

FILTRO SAVITZKY-GOLAY

El filtro de Savitzky-Golay (SG), también conocido como suavizado polinómico o filtro de mínimos cuadrados, es la generalización de promedios móviles [154]. Este filtro es capaz de preservar de una mejor manera el contenido de alta frecuencia de la señal deseada, a expensas de no eliminar tanto ruido como el método basado en promedios. El funcionamiento del filtro SG se basa en una convolución[1] de mínimos cuadrados simplificada para suavizar y calcular derivadas de un conjunto de valores consecutivos. Matemáticamente, la convolución puede ser expresada como en la Ecuación 3.4

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \tag{3.4}$$

Los autores del filtro SG desmostraron que a partir de conjunto de números enteros se pueden obtener coeficientes de ponderación para llevar a cabo el proceso de suavizado. El uso de estos coeficientes de ponderación, conocidos como enteros de convolución, resulta ser exactamente equivalente a ajustar subconjuntos de datos sucesivos de puntos de datos adyacentes con un polinomio de bajo grado, que es computacionalmente más eficaz y rápido [175]. La convolución se puede entender como un filtro de promedio móvil ponderado, donde la ponderación está dada como un polinomio de cierto grado. La Ecuación 3.5 muestra la forma general de la convolución simplificada de mínimos cuadrados.

$$y_i = \frac{\sum_{n=-m}^{m} c_n y_{i,j+n}}{N} \tag{3.5}$$

Donde y_i es el valor resultante de la combinación lineal para el j-ésimo punto de datos, c_n es el coeficiente para j-ésimo valor del filtro, N es el número de enteros que se combinan y que son iguales a la ventana de suavizado $(2m+1)$, donde m es el ancho medio de la ventana de suavizado.

La idea del filtrado de Savitzky-Golay es aproximar la función subyacente dentro de la ventana móvil no por una constante (cuya estimación es el promedio), sino por un polinomio de orden superior, típicamente cuadrático o cuártico.

REGRESIÓN LOCAL: LOESS Y LOWESS

Dado un conjunto de datos que consta de varias variables y múltiples observaciones, el principal objetivo de un método de suavizado es construir una relación funcional entre las variables [96]. La regresión local es uno de los métodos más comunes para llevar a cabo un proceso de suavizado, donde se supone que las observaciones disponibles tienen la forma que muestra la Ecuación 3.6.

$$\{(t_{i,j}, y_{i,j}); i = 1, ..., n\}, \tag{3.6}$$

[1] Una convolución es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión invertida y desplazada en un tiempo τ de g.

donde $t_{i,j}$ es una medida de la variable predictora (o independiente), y $y_{i,j}$ es la respuesta correspondiente. Por lo tanto, un modelo funcional que relaciona las variables toma la forma que se muestra en la Ecuación 3.7.

$$y_{i,j} = g(t_{i,j}) + \epsilon_{i,j}, \tag{3.7}$$

donde g es la función de regresión y $\epsilon_{i,j}$ es un error aleatorio, la idea básica de la regresión local es que en un predictor $t_{i,j}$, la función de regresión $g(t_{i,j})$ puede aproximarse localmente por el valor de una función en una clase paramétrica especificada. Dicha aproximación local se obtiene ajustando una superficie de regresión a los puntos de datos dentro de un vecindario elegido para el punto $t_{i,j}$.

Loess y Lowess son dos métodos representativos de la regresión local, sus nombres son derivados de *locally weighted scatter plot smooth* [28, 109]. Ambos métodos utilizan regresión lineal ponderada para llevar a cabo el proceso de suavizado de los datos. El proceso de suavizado se considera local porque, al igual que el método de promedio móvil, cada valor suavizado se determina mediante puntos de datos vecinos definidos dentro de un intervalo o ventana. El proceso se pondera porque se define una función de ponderación de regresión para los puntos de datos contenidos dentro de la ventana. Además de la función de peso de regresión, puede usar una función de peso robusta, que hace que el proceso sea resistente a valores atípicos. Los métodos se diferencian por el modelo utilizado en la regresión: lowess usa un polinomio lineal, mientras que loess usa un polinomio cuadrático [99].

A continuación se se describen los pasos que sigue el proceso de regresión local aplicado a cada punto.

1. Calcular los pesos de regresión para cada punto de datos en la ventana, dichos pesos están dados por la Ecuación 3.8.

$$w_i = \left(1 - \left|\frac{x - x_i}{d(x)}\right|\right)^3 \tag{3.8}$$

   donde x es el predictor asociado con el valor de respuesta a ser suavizado, x_i son los vecinos más cercanos de x definidos por la ventana, y $d(x)$ es la distancia a lo largo de la abscisa desde x hasta el valor del predictor más distante dentro de la ventana.

2. Se realiza una regresión lineal ponderada de mínimos cuadrados. Para lowess, la regresión usa un polinomio de primer grado, mientras que loess usa un polinomio de segundo grado.

3. El valor suavizado es obtenido por la regresión ponderada al valor predictivo de interés.

Finalmente, se puede decir que si el cálculo del suavizado involucra el mismo número de puntos de datos vecinos en cualquier lado (adyacente derecho o adyacente izquierdo) del punto de interés, la función de ponderación es simétrica, de lo contrario se dice que es asimétrica.

3.4. MEDIDAS DE SIMILITUD DE SERIES DE TIEMPO

Un problema importante al momento de trabajar con series de tiempo es determinar su similitud entre pares, es decir, el grado en que una serie de tiempo es parecida a otra. Para la

comparación entre pares de series de tiempo existen las medidas de similitud (o disimilitud) que representan una pieza fundamental para muchas tareas como la clasificación de series de tiempo.

Definición 3 *Sea* $\mathbf{X}$ *un conjunto no vacío de elementos denominados como puntos. Una métrica en* $\mathbf{X}$ *es una función de distancia* $d(p,q)$ *asociada a cada par de elementos del conjunto que cumpla las siguientes características [160]:*

a) $d(p,q) \geq 0$ para todo $p,q \in \mathbf{X}$, y $d(p,q) = 0$ si y solo si $p = q$.

b) $d(p,q) = d(q,p)$ para todo $p,q \in \mathbf{X}$.

c) $d(p,z) \leq d(p,q) + d(q,z)$ para todo $p,q,z \in \mathbf{X}$

Entonces el par $(\mathbf{X}, d)$ es denominado espacio métrico, y d es conocida como métrica o medida. Existen diversas formas de medir la distancia entre dos elementos, y elegir la métrica adecuada es crucial para obtener resultados competitivos en cualquier aplicación. Con frecuencia se utilizan funciones que miden la similitud.

Definición 4 *Una medida de similitud* d *es una función matemática que asocia un valor numérico a un par de descriptores. Dadas dos series de tiempo* $\mathbf{q}$ *y* $\mathbf{p}$ *como entrada y calcula la distancia entre ellas,* $d(\mathbf{q},\mathbf{p})$ *[180].*

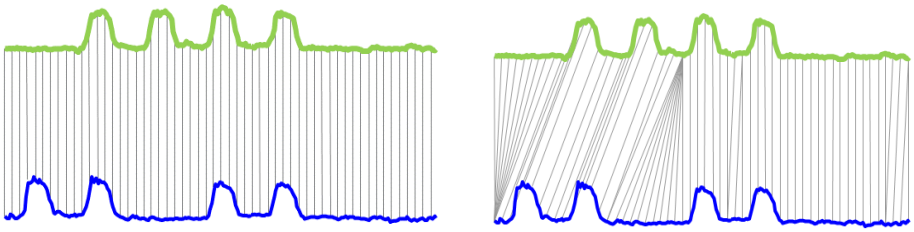
Dentro de la literatura se han reportado diversas medidas de similitud que han sido clasificadas en cuatro categorías [180]:

- **De paso cerrado (lock-step).** Medidas que comparan el j-ésimo punto de una serie de tiempo con el j-ésimo punto de otra.
- **Elásticas.** Son medidas de distancia que calculan la similitud en ciertas variaciones no lineales en la dimensión de las series de tiempo. Ésto se lleva a cabo mediante la comparación de uno-a-muchos o uno-a-ninguno.
- **Basadas en umbrales.** La idea principal de esta categoría es el uso de un parámetro de umbral τ para el cálculo de la similitud.
- **Basadas en patrones.** Este tipo de medidas se basa en encontrar segmentos (patrones) que coincidan dentro de la serie de tiempo completa, permitiendo desplazamientos y escalas tanto en la dimensión temporal como en la de amplitud.

A continuación se describen algunas medidas de similitud que fueron consideradas en este trabajo de tesis.

3.4.1. DISTANCIA EUCLIDIANA (NORMA L_2)

La distancia Euclidiana (ED, *Euclidean Distance*), es la medida de similitud más utilizada [51], también conocida como norma L_2. Esta medida de distancia es catalogada como una



(a) ED (b) DTW

Figura 3.2: Representación gráfica de la distancia Euclideana y DTW, imágenes tomadas de la fuente Keogh and Ratanamahatana [81].

función de distancia métrica, ya que obedece a las tres propiedades métricas fundamentales: no-negatividad, simetría y desigualdad triangular [21]. La ED es expresada a través de la Ecuación 3.9, donde **p** y **q** son dos series de tiempo y n es la dimensión de las series.

$$ED(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{j=1}^{n} (p_j - q_j)^2} \tag{3.9}$$

La Figura 3.2(a) muestra un ejemplo de dos series de tiempo que tienen una forma similar pero que no están alineadas en el eje del tiempo. La ED, asume que el punto medio de una secuencia está alineado con el punto medio de la otra, produciendo una medición de disimilitud pesimista. Por otro lado, la Figura 3.2(b), muestra como la medida DTW permite calcular una medida de similitud más intuitiva.

3.4.2. DISTANCIA MANHATTAN (NORMA L_1)

La distancia de Manhattan (MD, *Manhattan Distance*) es una métrica también conocida como norma L_1, que es utilizada como medida de similitud entre dos series de tiempo. En esta métrica, la distancia entre dos puntos es la suma de las diferencias absolutas de los valores de presentes en cada serie de tiempo. La Ecuación 3.10 expresa la distancia MD.

$$MD(\mathbf{p},\mathbf{q}) = \sum_{j=1}^{n} |\,(p_j - q_j)\,| \tag{3.10}$$

3.4.3. DISTANCIA CHEBYSHEV (NORMA L_∞)

La distancia de Chebyshev (CD, *Chebyshev Distance*), también se denomina métrica máxima o métrica L_∞. La distancia de Chebyshev entre dos series de tiempo es la mayor diferencia absoluta en los valores por cada serie. En la Ecuación 3.11 se expresa la CD.

$$CD(\mathbf{p},\mathbf{q}) = \max_{0<j\leq n} |\,(p_j - q_j)\,| \tag{3.11}$$

3.4.4. ALINEACIÓN DINÁMICA TEMPORAL

La Alineación Dinámica Temporal (DTW, *Dynamic Time Warping*) es una técnica utilizada para encontrar una alineación óptima entre dos secuencias dadas (dependientes del

tiempo) bajo ciertas restricciones [111]. Fue propuesta por Hiroaki Sakoe y Seibi Chiba con el objetivo de dar solución a problemas vinculados al reconocimiento de voz [152].

DEFINICIÓN DE DTW

Suponga que se tienen dos series de tiempo, una serie **p** de longitud I, y una serie **q** de tamaño J, de acuerdo a Ecuación 3.12.

$$\begin{aligned} \mathbf{p} &= p_1, p_2, ..., p_j, ..., p_J \\ \mathbf{q} &= q_1, q_2, ..., q_i, ..., q_I \end{aligned} \tag{3.12}$$

Para llevar a cabo la alineación de dos series de tiempo usando DTW, primero se debe construir una matriz de $I \times J$ donde el elemento (i, j) de la matriz corresponde a la distancia cuadrada, $d(p_i, q_j) = (p_i - q_j)^2$, que es la alineación entre los puntos p_i y q_j.

La Figura 3.3 muestra como se puede construir una matriz a partir de dos series de tiempo **p** y **q**, que se establecen a lo largo del eje i y el eje j respectivamente. El plano de la matriz se puede condicionar para explorar las múltiples rutas de alineación entre las series de tiempo.

Para encontrar la mejor alineación entre dos series de tiempo se calcula una ruta o trayectoria a través de la matriz que minimiza la distancia total acumulada entre ellas. Se dice que la trayectoria optima es aquella que logra minimizar el costo de la deformación, mostrado en la Ecuación 3.13.

$$DTW(\mathbf{p}, \mathbf{q}) = \min_{c} \left\{ \sqrt{\sum_{k=1}^{K} c_k} \right\} \tag{3.13}$$

Donde c_k representa la matriz de elementos $(i, j)_k$ que también pertenece al k elemento de una trayectoria de deformación C, un conjunto contiguo de elementos de matriz que representan un mapeo entre **p** y **q**. La trayectoria de deformación puede ser encontrada mediante el uso de programación dinámica para evaluar la recurrencia expresada en la Ecuación 3.14.

$$\gamma(i, j) = d(p_i, q_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{3.14}$$

Donde $d(p_i, q_j)$ representa la distancia calculada en la actual celda de la matriz, y $\gamma(i, j)$ es la distancia acumulada de $d(p_i, q_j)$ y la mínima distancia acumulada de las tres celdas adyacentes.

Durante el cálculo de DTW se pueden generar múltiples trayectorias, para disminuir el número de trayectorias consideradas se han propuesto diversas restricciones que tienen que ser tomadas en cuenta y que se describen a continuación.

1. **Condición de borde:** $c_1 = (1, 1)$ y $c_k = (I, J)$. Los puntos de inicio y fin de la trayectoria de deformación deben ser el primero y el último de las series de tiempo alineadas.

2. **Condición de monotonicidad:** $i_{k-1} \leq i_k$ y $j_{k-1} \leq j_k$. Esta condición preserva el orden del tiempo en cada punto.

3. **Condición de continuidad:** $i_k - i_{k-1} \leq 1$ y $j_k - j_{k-1} \leq 1$. Como resultado de esta condición y la de monotonicidad, la relación expresada en la Ecuación 3.15 se mantiene

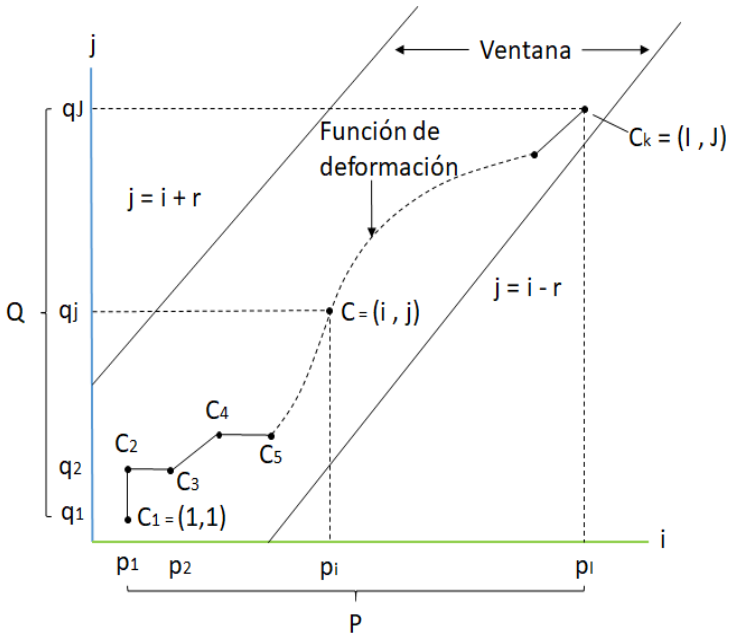


Figura 3.3: Definición de trayectoria de deformación y ajuste de ventana de tamaño r para la alineación de dos series de tiempo, imagen tomada de la fuente Sakoe and Chiba [152].

entre dos puntos consecutivos.

$$c_{k-1} = \begin{cases} (i_k, j_k - 1) \\ (i_k - 1, j_k - 1) \\ \text{o } (i_k - 1, j_k) \end{cases} \tag{3.15}$$

4. **Condición de la ventana de ajuste:** $i_k - j_k$ es menor que o aproximadamente igual a r, donde r es un entero positivo denominado como longitud de ventana. Esta condición corresponde al hecho de que la fluctuación del eje de tiempo en casos normales nunca causa una diferencia de tiempo excesiva, ver Figura 3.3.

5. **Condición de la restricción de la pendiente:** esta condición controla el comportamiento de la trayectoria de deformación, ya que no se debe permitir un gradiente demasiado empinado ni demasiado suave, con el objetivo de impedir que un segmento corto de una de las series corresponda con un segmento largo de la otra. La condición consiste en que, si el punto c_k se mueve m-veces de forma consecutiva en dirección del eje i (o j), entonces el punto c_k no puede dar un paso más en la misma dirección antes de avanzar al menos n-veces en la dirección diagonal. La intensidad efectiva de la restricción de la pendiente puede ser evaluada a través de la expresión mostrada en la Ecuación 3.16.

$$P = \frac{n}{m} \tag{3.16}$$

Cuanto mayor es la medida P, mayor es la restricción para la pendiente. Si $P = 0$ significa que no hay restricción (peor de los casos) y si $P = \infty$ la trayectoria está restringida a moverse

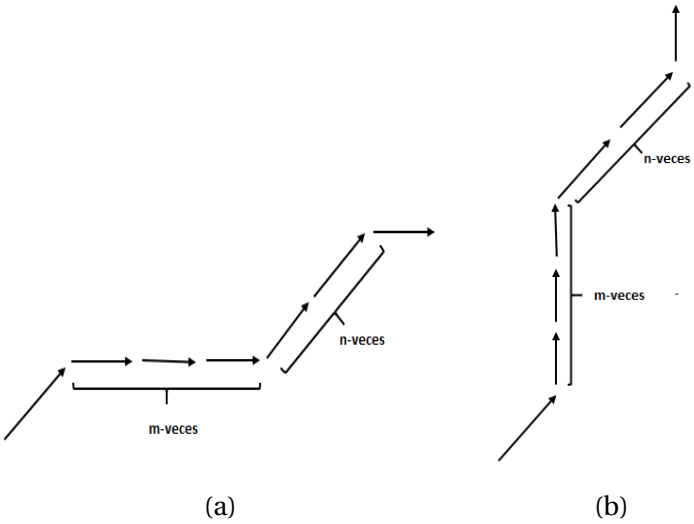


Figura 3.4: Ejemplos de restricciones de la pendiente para $P = \frac{2}{3}$, donde 3 movimientos del punto c_k en la dirección del eje vertical u horizontal, obliga a efectuar al menos dos movimientos en la dirección de la diagonal. (a) Pendiente máxima y (b) pendiente mínima, imágenes tomadas de la fuente Sakoe and Chiba [152].

sobre la diagonal, es decir que no permite deformación alguna. En la Figura 3.4 se muestran dos ejemplos de restricción de la pendiente, donde de forma concreta se puede decir que, si el punto c_k avanza en dirección del eje i (o j) m veces consecutivas, entonces el punto c_k no puede dar un paso más en la misma dirección antes de visitar al menos n veces en la dirección diagonal.

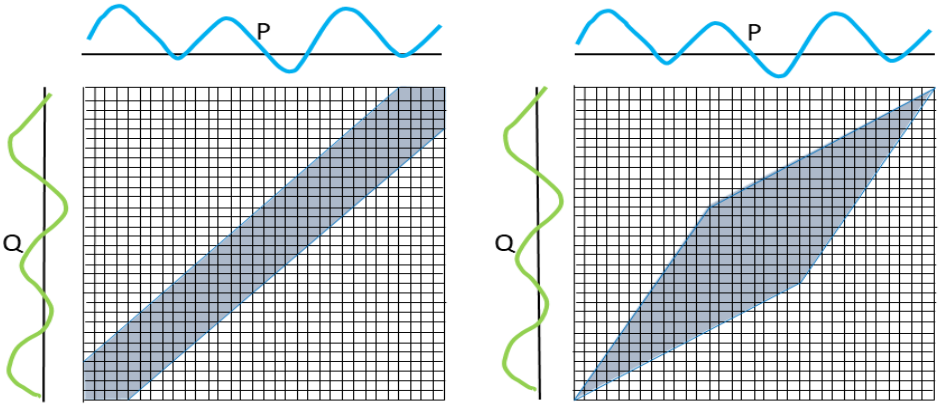
Además de las restricciones para la trayectoria de deformación, DTW ofrece la posibilidad de limitar la trayectoria de deformación de una forma global al restringir la distancia que puede desviarse de la diagonal. A este subconjunto restringido de la matriz que la trayectoria es capaz de visitar es denominado como ventana (warping). Las restricciones globales más utilizadas son la banda de Sakoe-Chiba [152] y el paralelogramo de Itakura [76], en la Figura 3.5 se muestra un ejemplo de estas restricciones globales. El objetivo de usar la restricción global es acelerar ligeramente el cálculo de la distancia DTW. Sin embargo, la razón más importante es la prevención de deformaciones *patológicas*[2], es decir, donde una sección relativamente pequeña de una serie de tiempo se asigna a una sección relativamente grande de otra.

3.4.5. LÍMITE INFERIOR DE KEOGH (LB_KEOGH)

Una de las grandes desventajas de la medida DTW es el alto consumo de tiempo en CPU. Por lo tanto una de las formas para abordar dicho problema es el uso de una función de límite inferior (lower bound) para ayudar a eliminar secuencias que podrían no ser la mejor opción. LB_Keogh es una función de límite inferior propuesta por Keogh [81], conocida como la primera función no-trivial de límite inferior para DTW.

De forma general, en el Algoritmo 1 se muestra cómo la función de límite inferior es in-

[2]Adjetivo que se encuentra en la literatura, que hace referencia a rutas extrañas de alineación que se pueden producir entre dos series temporales.



(a) Sakoe-Shiba (b) Itakura

Figura 3.5: Representación gráfica de las restricciones globales de (a) Sakoe-Chiba y (b) Itakura. El área azul muestra el alcance de la trayectoria de deformación, imagen tomada de la fuente Ratanamahatana and Keogh [137].

corporada en el procedimiento del cálculo de similitud de series de tempo a través de DTW.

Algoritmo 1 Límite_Inferior_Secuencial

Entrada: Q

1: $mejor_hasta_momento$ = infinity;
2: **for** (todas las series de tiempo en la base de datos) **do**
3: LB_distancia = LB_KEOGH(P_i, Q);
4: **if** LB_distancia < $mejor_hasta_momento$ **then**
5: distancia_real = DTW(P_i, Q)
6: **if** distancia_real < $mejor_hasta_momento$ **then**
7: $mejor_hasta_momento$ = distancia_real;
8: $indice_mejor_encontrado$ = i;
9: **end if**
10: **end if**
11: **end for**

Una restricción global o local de DTW se pueden ver como una restricción de los índices de la trayectoria de deformación $c_k = (i, j)_k$ tal que $j - r \leq i \leq j + r$, donde r es un término que define el alcance o rango permitido de deformación para un determinado punto dado de una serie de tiempo. En el caso de la banda de Sakoe-Chiba, r es independiente de i, mientras que para el paralelogramo de Itakura, r es una función de i. Para el caso de LB_Keogh es necesario definir dos nuevas series de tiempo, **u** y **l** como se muestra en la Ecuación 3.17.

$$\begin{aligned} u_i &= \text{máx}(q_{i-r} : q_{i+r}) \\ l_i &= \text{mín}(q_{i-r} : q_{i+r}) \end{aligned} \tag{3.17}$$

u y **l** representan el límite superior e inferior, respectivamente. Ambas series forman una banda envolvente sobre la serie de tiempo **q**, como se puede observar en la Figura 3.6. Por lo

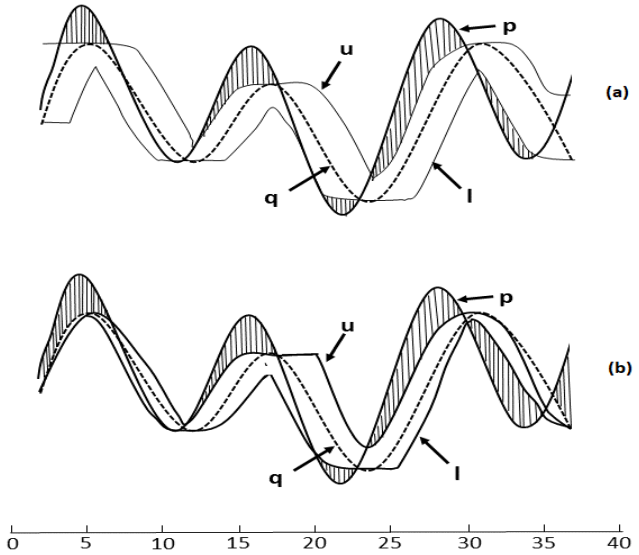


Figura 3.6: Representación gráfica del envoltorio producido por LB_Keogh para una serie donde se usa (a) Sakoe-Chiba e (b) Itakura, imágenes tomadas de la fuente Keogh and Ratanamahatana [81].

tanto, una obvia e importante propiedad de **u** y **q** es la que se expresa en la Ecuación 3.18.

$$\forall u_i \geq q_i \geq l_i \tag{3.18}$$

Una vez definidas **u** y **l**, éstas son usadas para calcular la medida de límite inferior para DTW, expresada en la Ecuación 3.19. Esta función se puede visualizar como la distancia Euclidiana entre cada parte de la serie de tiempo que no se encuentra dentro de la envolvente y la sección correspondiente más cercana a la envoltura.

$$LB_Keogh(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n} \begin{cases} (q_i - u_i)^2 & \text{si } q_i > u_i \\ (q_i - l_i)^2 & \text{si } q_i < l_i \\ 0 & \text{si no} \end{cases}} \tag{3.19}$$

3.5. REPRESENTACIÓN DE SERIES DE TIEMPO

Las series de tiempo se caracterizan por ser datos de alta dimensión, por tal motivo trabajar con estos datos en su formato real sin procesar resulta ser muy costoso en términos de costo de procesamiento y almacenamiento. Por lo tanto, se vuelve indispensable desarrollar métodos de representación que sean capaces de reducir la dimensionalidad de las series de tiempo sin perder las características fundamentales de un conjunto de datos en particular [180, 186].

Definición 5 *Una representación de una serie de tiempo* **t** *de longitud* n *es un modelo* $\hat{\mathbf{t}}$ *de dimensionalidad* m*, donde* $(m \ll n)$ *tal que* $\hat{\mathbf{t}}$ *se aproxima a* **t** *[186].*

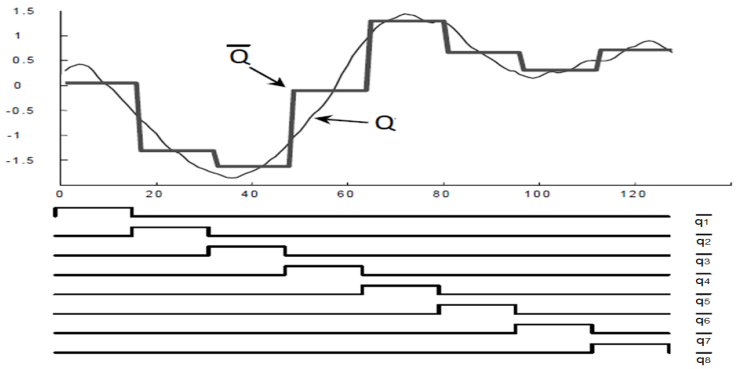


Figura 3.7: Representación PAA, reducción de una serie de tiempo de 128 de longitud a 8 dimensiones, imagen tomada de la fuente Lin et al. [95].

Se han propuesto diferentes métodos para la representación de series de tiempo con dimensionalidad reducida, por ejemplo: Transformación Discreta de Fourier (DFT) [51], Transformación Discreta de Wavelet (DWT) [24], Aproximación Agregada de Piezas (PAA) [82], Aproximación Simbólica de Piezas (SAX) [95], Análisis de Componentes Principales (PCA) [161], Descomposición de Valor Único (SVD) [51], entre otras. A continuación se describen tres métodos de representación de series de tiempo que son utilizados en este trabajo.

3.5.1. APROXIMACIÓN AGREGADA DE PIEZAS (PAA)

PAA es un método simple de representación de series de tiempo basada en piezas [82]. En el método PAA, una serie de tiempo $\mathbf{q}$ de longitud n puede representarse en un espacio w-dimensional por un vector $\mathbf{q} = q_1, q_2, ..., q_w$, donde cada $\mathbf{q}_i$ se calcula de acuerdo a la Ecuación 3.20.

$$\mathbf{q}_i = \frac{w}{n} \sum_{j=\frac{w}{n}(i-1)+1}^{\frac{w}{n}i} q_j \tag{3.20}$$

Se dice que para reducir una serie de tiempo de n dimensiones a w dimensiones, los datos se dividen en w marcos o ventanas del mismo tamaño. Se calcula el valor medio de los datos que caen dentro de una ventana y un vector de estos valores se convierte en la representación de datos reducidos. La Figura 3.7 muestra la representación de PAA y cómo busca aproximar una serie de tiempo Q a través de la combinación lineal de funciones básicas de caja. En este ejemplo, una serie de tiempo de longitud 128 es reducida a 8 dimensiones que son representados como líneas horizontales en la serie de tiempo $\bar{Q}$.

3.5.2. APROXIMACIÓN SIMBÓLICA DE PIEZAS (SAX)

SAX es un método de representación de series de tiempo propuesto en [95]. Este método está basado en el método PAA, heredando la simplicidad y la baja complejidad computacional mientras provee sensibilidad y selectividad satisfactorias en el procesamiento de consulta de un rango. SAX transforma una serie de tiempo $\mathbf{q}$ de longitud n en otra representada mediante una cadena de longitud w, donde $w \ll n$, usando un alfabeto A de tamaño $a \geq 2$. El método SAX consta de dos pasos principalmente:

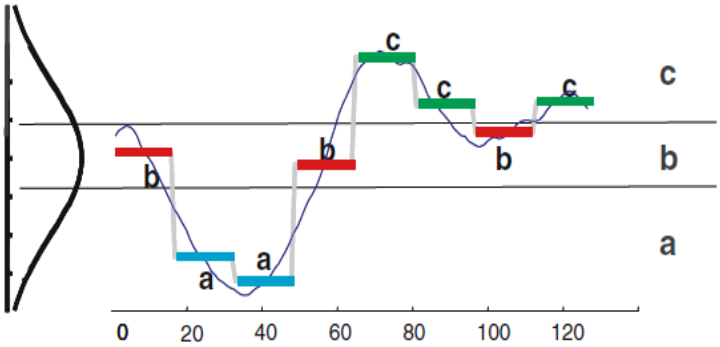


Figura 3.8: Serie de tiempo representada mediante SAX, con $n = 128$, $w = 8$ y $a = 3$, imagen tomada de la fuente Lin et al. [95].

1. *Reducción de dimensionalidad a través de PAA.* Durante el primer paso se transforma la serie de tiempo original en una representación PAA.

2. *Discretización de PAA.* Una vez que se tiene una representación PAA, se procede a aplicar una transformación adicional para obtener una representación discreta. Es deseable usar una técnica de discretización capaz de producir símbolos con equiprobabilidad. Ésto se logra fácilmente ya que las series normalizadas tienen una distribución Gaussiana. A partir de la curva Gaussiana se determinan los puntos de corte que producirán a áreas de igual tamaño. Todos los coeficientes PAA que se encuentran por debajo del punto de corte más pequeño se les asigna el símbolo a, todos los coeficientes mayores que o igual al punto de corte más pequeño y menor al segundo punto de corte más pequeño se les asigna el símbolo b y así sucesivamente.

A partir de los dos pasos principales de SAX es necesario definir dos conceptos:

- **Puntos de corte.** Los puntos de corte son una lista ordenada de números $B = \beta_1, ..., \beta_{a-1}$, tal que el área bajo una curva Gaussiana de β_i a β_{i+1} es igual a $1/a$ (β_0 y β_a son definidas como $-\infty$ y $+\infty$ respectivamente).

- **Palabra.** Una subsecuencia $\mathbf{q}$ de tamaño n puede ser representada como una palabra $\hat{\mathbf{q}} = \hat{q}_1, ..., \hat{q}_w$ de la siguiente forma. Sea α_i el i-ésimo elemento del alfabeto, es decir, $\alpha_1 = \mathbf{a}$ y $\alpha_2 = \mathbf{b}$. Entonces la transformación de una aproximación PAA $\mathbf{q}$ a una palabra $\hat{\mathbf{q}}$ es obtenida mediante la Ecuación 3.21.

$$\hat{q}_i = \alpha_j \iff \beta_{j-1} \leq \bar{q}_i < \beta_j \tag{3.21}$$

En la Figura 3.8 se muestra un ejemplo de una representación discreta de una serie de tiempo obtenida mediante SAX. La serie de tiempo del ejemplo tiene una longitud de 128 y es reducida a 8 segmentos.

3.5.3. ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

PCA es un método estadístico efectivo para la síntesis de información, o reducción de la dimensión, bajo la premisa de la pérdida escasa o nula de la información en bruto[93]. PCA busca reducir la cantidad de variables, donde las nuevas variables son consideradas componentes principales o factores que serán la combinación lineal de las variables originales, y además serán independientes entre sí.

Se asume que $\mathbf{X}$ es una matriz con p variables y n muestras o instancias, donde $\mathbf{x}$ es un vector de datos $x_1, x_2, ..., x_p$ del que se desea obtener un vector reducido z de nuevas n variables $z_1, z_2, ..., z_n$, conocido como componente principal.

Cada componente principal z_i se obtiene por la combinación lineal de las variables originales. La primera componente de un grupo de variables $x_1, x_2, ..., x_p$ es la combinación normalizada de dichas variables con la mayor varianza, como se muestra en la Ecuación 3.22.

$$z_1 = a_{11}x_1 + a_{12}x_2 + ...a_{1p}x_p \tag{3.22}$$

Donde los términos $a_{11}, ..., a_{1p}$ representan el peso o la importancia que tiene cada variable en cada componente y, por lo tanto, ayudan a conocer la calidad de la información que representa cada componente.

Debido a que la primer componente principal se calcula de manera que represente la mayor variación posible en el conjunto de datos, se podría hacer que la varianza de z_1 sea lo más grande posible eligiendo valores grandes para los pesos $a_{11}, a_{12}, ...a_{1p}$. Para evitar esto, los pesos se calculan con la restricción de que su suma de cuadrados sea 1, como se expresa en la Ecuación 3.23.

$$a_{11}^2 + a_{12}^2 + ... + a_{1p}^2 = 1 \tag{3.23}$$

Dada una matriz de datos $\mathbf{X}$, el proceso a seguir para calcular la primera componente principal es:

- Normalización de las variables, se resta a cada valor la media de la variable a la que pertenece. Con esto se logra que todas las variables tengan una media cero.
- Encontrar el valor adecuado de los pesos con los que se maximiza la varianza. Ésto se resuelve mediante un problema de optimización donde se calculan los valores y vectores propios de la matriz de covarianza S de los datos originales.

El proceso se repite de forma iterativa hasta encontrar todas las posibles componentes.

3.6. REDUCCIÓN DE DATOS

La reducción de datos (NR, *Numerosity Reduction*) es la tarea de reducir el número de instancias en una base de datos mientras conserva algunas propiedades esenciales [188]. También es conocida como selección de prototipos, selección de instancias, condensación o edición de datos [173]. La mayor parte de los métodos de NR se han propuesto en el contexto de la clasificación de vecinos más cercanos con el objetivo de acelerar el proceso de clasificación [114]. Los métodos de NR construyen un pequeño grupo representativo de los

datos de entrenamiento inicial, este conjunto es denominado como conjunto de condensación y posee los beneficios de un bajo costo computacional y almacenamiento, manteniendo la precisión en niveles altos. En la literatura relacionada, se ha propuesto clasificar a los métodos NR, en dos categorías: (1) Selección de prototipos (PS) y (2) Abstracción de prototipos (PA) [119]. Ambas categorías persiguen el mismo objetivo que es la reducción de datos. La diferencia entre estas dos categorías es que los métodos PS seleccionan algunos elementos de entrenamiento y son utilizados como elementos representantes, mientras que los métodos PA generan nuevos representantes a partir de una muestra de elementos similares en el conjunto de entrenamiento. Partiendo de esta clasificación, se describen a continuación dos métodos representativos de la categoría PS para la reducción de datos en bases de datos temporales.

3.6.1. MONTE CARLO 1

Monte Carlo 1 (MC1) es un método simple de muestreo repetido del conjunto de datos, donde el muestreo se realiza con reemplazo. El método MC1 trabaja en conjunto con el clasificador k-vecinos más cercanos. Los parámetros considerados por MC1 son k (número de k vecinos), m (la cantidad de prototipos, que es el tamaño de la muestra) y n (la cantidad de muestras tomadas) [162]. El funcionamiento de MC1 se puede resumir de la siguiente manera. Suponiendo que el conjunto de datos ha sido dividido en un conjunto de entrenamiento y de prueba.

- Seleccionar n muestras aleatorias, cada muestra es seleccionada con reemplazo de m instancias del conjunto de entrenamiento.
- Por cada muestra, se calcula la precisión (accuracy) de clasificación en el conjunto de entrenamiento a través del algoritmo 1-vecino más cercano.
- Seleccionar la muestra con la precisión de clasificación más alta en el conjunto de entrenamiento.

Este método es simple de aplicar, sin embargo puede resultar altamente costoso al momento de calcular la precisión de clasificación de cada una de las muestras establecidas.

3.6.2. INSIGHT

INSIGHT es un método propuesto para la selección de instancias en bases de datos de series de tiempo, basado en la cobertura de grafos y el concepto de *hubness* [19]. INSIGHT realiza la selección de instancias asignando una puntuación a cada instancia del conjunto de entrenamiento y posteriormente selecciona las instancias con las puntuaciones más altas. Para el manejo de las puntuaciones, INSIGHT hace uso de una función de puntuación basada en la propiedad *hubness*. Esta propiedad indica que, para datos con alta dimensionalidad (intrínseca), como la mayoría de las series de tiempo, algunos objetos tienden a convertirse en vecinos más cercanos con mucha más frecuencia que otros. Para expresar la propiedad hubness, para un conjunto de datos $\boldsymbol{D}$ se definen k-ocurrencias de una instancia $\mathbf{x} \in \boldsymbol{D}$, denotadas como $f_N^k(\mathbf{x})$, que es el número de instancias de $\boldsymbol{D}$ que tienen a $\mathbf{x}$ entre sus k vecinos.

En presencia de datos etiquetados, se puede diferenciar buen-hubness y mal-hubness. Se dice que la instancia $\mathbf{y}$ es un buen (mal) k-vecino más cercano de la instancia $\mathbf{x}$ si:

1. **y** es uno de los k-vecinos más cercanos de **x**,
2. ambas tienen el mismo (diferente) etiqueta de clase.

En el contexto de series de tiempo, se puede definir una buena (mala) k-ocurrencia de una serie de tiempo **x** como $f_G^k(\mathbf{x})$ y $f_B^k(\mathbf{x})$ respectivamente, que es el número de otras series de tiempo que tienen a x como uno de sus buen (mal) k-vecinos más cercanos.

El método INSIGHT propone tres funciones de puntuación, sin embargo una de las tres presenta mejores resultados que las otras. La función que reporta los mejores resultados toma en cuenta la buena 1-ocurrencia de una instancia **x** expresada en la Ecuación 3.24.

$$f_G(\mathbf{x}) = f_G^1(\mathbf{x}) \tag{3.24}$$

El método INSIGHT trabaja en función de la clasificación a través del algoritmo de k-vecinos más cercanos y el uso de la función de puntuación, en el Algoritmo 2 se resume su funcionamiento.

Algoritmo 2 Método INSIGHT

Entrada: Base de datos de series de tiempo $\boldsymbol{D}$, Función de puntuación, Número de instancias a seleccionar (N)

Salida: Conjunto de instancias seleccionadas $\boldsymbol{D}'$

1: Calcular función de puntuación Ecuación 3.24 para todo $\mathbf{x} \in \boldsymbol{D}$
2: Ordenar todas las series de tiempo en $\boldsymbol{D}$ de acuerdo a sus puntuaciones
3: Seleccionar las N mejores instancias

4

Aprendizaje Automático

> La verdadera señal de inteligencia no es el conocimiento sino la imaginación.
>
> —Albert Einstein

Se dice que la capacidad de aprender es una de las características distintivas del comportamiento inteligente. En un contexto general, se puede decir que el proceso de aprendizaje incluye la adquisición de nuevos conocimientos declarativos, el desarrollo de habilidades a través de instrucciones o de la práctica y el descubrimiento de nuevos hechos y teorías a través de la observación y la experimentación [22]. En ciencias de la computación se encuentra en concepto de aprendizaje automático (ML, *Machine Learning*) que es el estudio y el modelado del proceso de aprendizaje en sus múltiples manifestaciones [26, 107]. De acuerdo con Camastra [22], una de las categorías en las que se divide el ML es el aprendizaje basado en ejemplos, donde dado un conjunto de ejemplos de un cierto dominio, un aprendiz (algoritmo) induce una descripción general del dominio que mejor describa a los ejemplos. El aprendizaje basado en ejemplos se subdivide en cuatro familias: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje por refuerzo y aprendizaje semi-supervisado. Este documento se enfoca en la familia de aprendizaje supervisado que se describe a continuación.

4.1. Aprendizaje Supervisado: Clasificación

En el aprendizaje supervisado, los datos son una muestra de patrones de entrada-salida, donde los datos que se consideran ya están asociados con un determinado valor objetivo, por ejemplo, clases y valores reales [68]. Una de las tareas más populares del aprendizaje supervisado es la clasificación que consiste en encontrar un modelo o función que describa y distinga clases de datos u objetos, con el propósito de poder usar el modelo para predecir la clase de objetos cuya etiqueta de clase es desconocida. El modelo derivado se basa en el análisis de un conjunto de datos de entrenamiento (es decir, objetos de datos cuya etiqueta de clase es conocida) [170].

Definición 6 *La clasificación es la tarea de aprender una función objetivo f que asigna a una instancia de datos* $\mathbf{x}$ *una de las etiquetas de clase predefinidas* $c_i \in \boldsymbol{C}$. *Informalmente, la función objetivo es conocida como modelo de clasificación [170].*

La clasificación se puede resumir en dos principales etapas que son:

1. En la primera etapa, se construye un modelo a través de un algoritmo de clasificación (clasificador) que describa un conjunto predeterminado de clases de datos o conceptos. Esta fase es denominada como fase de aprendizaje (o de entrenamiento), donde un clasificador analiza un conjunto de datos de *entrenamiento* $\boldsymbol{D}_{train}$, compuesto por instancias y sus etiquetas de clase asociadas.

2. La segunda etapa, el modelo construido durante la etapa de entrenamiento, es utilizado para clasificación. En esta etapa se hace uso de un conjunto de datos de *prueba* $\boldsymbol{D}_{test}$, compuesto por sus instancias de prueba y sus respectivas etiquetas de clase asociadas a cada instancia. El conjunto de prueba es totalmente independiente del conjunto de entrenamiento, por lo tanto permite validar qué tan bueno o malo es el modelo para llevar a cabo la tarea de clasificación.

Dentro de la literatura especializada, se han reportado diversos enfoques de algoritmos de clasificación. A continuación se describen algunos clasificadores que fueron utilizados en este trabajo.

4.1.1. K-VECINOS MÁS CERCANOS

El método de k-vecinos más cercanos (k-NN, por sus siglas en inglés) se describe por primera vez a principios de la década de 1950 [68]. k-NN se basa en el aprendizaje por analogía, es decir, compara una instancia de prueba con instancias de entrenamiento que son similares a ella. k-NN representa cada instancia como un punto en un espacio p-dimensional, donde p es el número de atributos o variables. Dada una instancia de prueba, se calcula su proximidad al resto de los puntos en el conjunto de entrenamiento, utilizando alguna medida de distancia o similitud descritas en la Sección 3.4.

Se dice que los k vecinos más cercanos de una instancia dada $\mathbf{z}$ se refiere a los k puntos más cercanos a $\mathbf{z}$. Cada punto se clasifica de acuerdo a las etiquetas de clase de sus vecinos. En el caso donde los vecinos tienen más de una etiqueta, el punto se asigna a la clase mayoritaria de sus vecinos. En el caso donde se presenta un empate entre clases se puede elegir aleatoriamente una de ellas para clasificar el punto. La asignación de la clase para un punto dado subraya la importancia de elegir adecuadamente el valor para k, ya que si k es demasiado pequeño, entonces el clasificador puede ser susceptible al sobreajuste (overfitting) [170]. Por otro lado, si k es demasiado grande, el clasificador puede clasificar erróneamente la instancia de prueba.

En la Figura 4.1, se muestra un ejemplo de un conjunto de datos categorizados por la clase (+) y la clase (-). Se observa una instancia desconocida representada como (x). Para poder clasificar la instancia desconocida, k-NN necesita calcular la distancia que existe entre la instancia desconocida (x) y cada instancia del conjunto de datos disponible. Posteriormente, k-NN debe seleccionar los k vecinos más cercanos, en el caso de la Figura 4.1 el valor de k es muy amplio, ya que considera un radio de 21 vecinos cercanos. En este ejemplo, la instancia

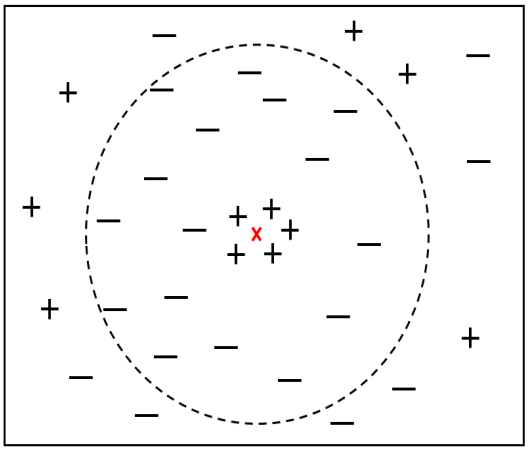


Figura 4.1: Ejemplo de k-NN con un valor de k amplio [170].

desconocida será clasificada como clase (-) por ser la clase con mayor número de miembros dentro del radio de los vecinos más cercanos. Si el valor de k es muy amplio, el vecindario puede incluir instancias de otras clases y dando como resultado una clasificación incorrecta.

El procedimiento del clasificador k-NN es presentado en el Algoritmo 3, donde se calcula la distancia entre cada instancia $\mathbf{z} = (\mathbf{x}', c')$ y todas las instancias de entrenamiento $(\mathbf{x}, c) \in \boldsymbol{D}_{train}$ para determinar el conjunto de sus vecinos más cercanos $\boldsymbol{D}_z$.

Algoritmo 3 Clasificador k-NN

Entrada: k número de vecinos, $\boldsymbol{D}_{train}$ conjunto de entrenamiento

1: **for** cada instancia de prueba $\mathbf{z} = (\mathbf{x}', c')$ **do**
2: Calcular $d(\mathbf{x}', \mathbf{x})$, la distancia entre $\mathbf{z}$ y cada instancia $(\mathbf{x}, c) \in \boldsymbol{D}_{train}$
3: Seleccionar $\boldsymbol{D}_z \subseteq \boldsymbol{D}$, el conjunto de k instancias de entrenamiento cercanas a $\mathbf{z}$
4: $c' = \text{máx}_v \sum_{(x_i, c_i) \in D} I(v = c_i)$
5: **end for**

Una vez que se obtiene el conjunto de vecinos más cercanos, la instancia de prueba es clasificada basada en una votación de la clase mayoritaria de sus vecinos, de acuerdo a la Ecuación 4.1. Donde v es una etiqueta de clase, c' es la etiqueta de clase para uno de los vecinos más cercanos e $I(\cdot)$ es un indicador de función que devuelve 1 si su argumento es verdadero y 0 de lo contrario.

$$\text{Votación Mayoritaria: } c' = \underset{v}{\text{máx}} \sum_{(x_i, c_i) \in D_z} I(v = c_i) \tag{4.1}$$

4.1.2. CLASIFICADOR BAYESIANO

Existe una gama de clasificadores conocidos como *clasificadores bayesianos* que son capaces de predecir las probabilidades de pertenencia de la clase, es decir, la probabilidad de que una instancia determinada pertenezca a una clase en particular. La clasificación bayesiana se basa en el teorema de Bayes que se describe a continuación:

TEOREMA DE BAYES

El teorema de Bayes lleva el nombre de Thomas Bayes, un clérigo inglés que realizó trabajos tempranos en la probabilidad y la teoría de la decisión durante el sigo XVIII [68]. Sea $\mathbf{x}$ una instancia de datos de n atributos, en términos bayesianos $\mathbf{x}$ es considerada como *evidencia*. Sea h una hipótesis, como que la instancia de datos $\mathbf{x}$ pertenece a una clase c específica. Para problemas de clasificación se desea determinar $P(h \mid \mathbf{x})$, la probabilidad de que la instancia $\mathbf{x}$ pertenezca a la clase c, dado que se conoce la descripción de atributos de $\mathbf{x}$. $P(h \mid \mathbf{x})$, es la probabilidad a posteriori de h condicionada en $\mathbf{x}$, en contraste $P(h)$ es la probabilidad a priori de h y $P(\mathbf{x})$ es la probabilidad a priori de $\mathbf{x}$. Estas probabilidades pueden ser calculadas a través del teorema de Bayes que se expresa en la Ecuación 4.2.

$$P(h \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid h)P(h)}{P(\mathbf{x})} \tag{4.2}$$

NAÏVE BAYES

El clasificador Naïve Bayes (NB) se basa en el teorema de Bayes para llevar a cabo la tarea de clasificación de acuerdo a los siguientes pasos:

1. Sea $\boldsymbol{D}_{train}$ un conjunto de datos de entrenamiento con sus respectivas etiquetas de clase, donde cada instancia es representada por un vector n-dimensional de atributos $\mathbf{x} = x_1, x_2, ..., x_n$.

2. Suponga que hay m clases, $c_1, c_2, ..., c_m$. Dada una instancia $\mathbf{x}$ el clasificador debe predecir que $\mathbf{x}$ pertenece a la clase que tiene la mayor probabilidad a posteriori, condicionada en $\mathbf{x}$. Es decir, NB predice que la instancia $\mathbf{x}$ pertenece a la clase c_i si y solo si la Ecuación 4.3.

$$P(c_i \mid \mathbf{x}) > P(c_j \mid \mathbf{x}) \quad \text{para } i \leq j \leq m, \quad j \neq i \tag{4.3}$$

   La clase c_i para la cual $P(c_i \mid \mathbf{x})$ se maximiza es llamada hipótesis máxima a posteriori. De acuerdo al teorema de Bayes se expresa la Ecuación 4.4.

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} \tag{4.4}$$

3. Como $P(\mathbf{x})$ es constante para todas las clases, solo $P(\mathbf{x} \mid c_i)P(c_i)$ necesita ser maximizada. Si no se conocen las probabilidades a priori de la clase, entonces se asume que las clases son igualmente probables $P(c_i) = P(c_2) = ... = P(c_m)$ y entonces se maximiza $P(\mathbf{x} \mid c_i)$, de otro modo se maximiza $P(\mathbf{x} \mid c_i)P(c_i)$. Debe tomarse en cuenta que las probabilidades a priori pueden ser estimadas a través de la Ecuación 4.5, donde $\mid c_{i,D} \mid$ es el número de instancias de entrenamiento de la clase c_i en $\boldsymbol{D}_{train}$.

$$P(c_i) = \frac{\mid c_{i,D_{train}} \mid}{\boldsymbol{D}_{train}} \tag{4.5}$$

4. En conjuntos de datos donde las instancias pueden ser de alta dimensionalidad, como el caso de las series de tiempo, sería computacionalmente costoso calcular $P(\mathbf{x} \mid c_i)$.

Para reducir este cálculo, el clasificador NB realiza la suposición ingenua de la independencia condicional de clase, ésto expresado en la Ecuación 4.6.

$$\begin{aligned} P(\mathbf{x} \mid c_i) &= \prod_{k=1}^{n} P(x_k \mid c_i) \\ &= P(x_1 \mid c_i) \times P(x_2 \mid c_i) \times \ldots \times P(x_n \mid c_i) \end{aligned} \tag{4.6}$$

Las probabilidades $P(x_1 \mid c_i), P(x_2 \mid c_i), \ldots, P(x_n \mid c_i)$ de las instancias de entrenamiento se pueden estimar fácilmente. Considere que x_k se refiere al valor de atributo A_k para la instancia $\mathbf{x}$. Por cada atributo se debe revisar si es categórico o de valor continuo, para el cálculo de $P(\mathbf{x} \mid c_i)$ se consideran los siguientes casos:

a) Si A_k es categórico, entonces $P(x_k \mid c_i)$ es el número de instancias de la clase c_i en $\boldsymbol{D}_{train}$ donde se presenta el valor x_k para A_k, dividido por $\mid c_{i,D_{train}} \mid$, el número de instancias de la clase c_i en $\boldsymbol{D}$

b) Si A_k es un valor continuo, se supone que un atributo de valor continuo tiene una distribución Gaussiana con una media μ y una desviación estándar δ, definidas en la Ecuación 4.7.

$$g(x, \mu, \delta) = \frac{1}{\sqrt{2\pi\delta}} e^{-\frac{(x-\mu)^2}{2\delta^2}} \tag{4.7}$$

De modo que se expresa la Ecuación 4.8,

$$P(x_k \mid c_i) = g(x_k, \mu_{c_i}, \delta_{c_i}) \tag{4.8}$$

5. Finalmente, para predecir la etiqueta de clase de $\mathbf{x}$, $P(\mathbf{x} \mid c_i)P(c_i)$ se evalúa para cada clase c_i. El clasificador predice que la etiqueta de la instancia $\mathbf{x}$ es la clase c_i si y solo si se cumple la Ecuación 4.9.

$$P(\mathbf{x} \mid c_i)P(c_i) > P(\mathbf{x} \mid c_j)P(c_j) \quad \text{para} \quad 1 \leq j \leq m, j \neq i \tag{4.9}$$

4.1.3. ÁRBOL DE DECISIÓN

Un árbol de decisión (DT, *Decision Tree*) consiste en una estructura compuesta de nodos y ramas que conectan a dichos nodos. Los nodos en la parte inferior del árbol se denominan hojas y hacen referencia a clases de un conjunto de datos.

El nodo superior del árbol, llamado raíz, contiene todas las instancias de entrenamiento que se dividirán en clases. Todos los nodos excepto las hojas, son llamados nodos de decisión, ya que especifican la decisión que se debe realizar en ese nodo en función de un único atributo.

Cada nodo de decisión posee una cantidad de nodos secundarios, igual al número de valores que tiene un atributo determinado. La Figura 4.2 muestra un ejemplo de un DT [27]. Se dice que, en principio, existe una cantidad exponencial de árboles de decisión que se pueden construir a partir de un conjunto de atributos.

Algunos árboles pueden ser más precisos que otros, encontrar el árbol óptimo resulta computacionalmente complejo debido al gran tamaño espacio de búsqueda. Sin embargo

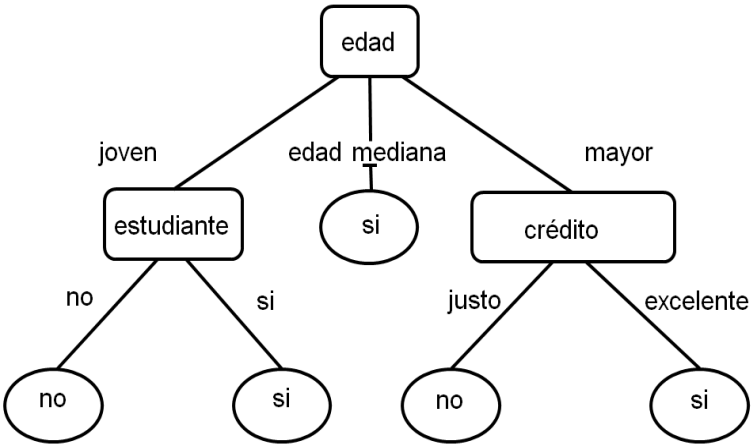


Figura 4.2: Ejemplo de un DT que determina si un cliente comprará una computadora [68].

se han propuesto algoritmos eficientes para inducir un árbol de decisión razonablemente preciso, que no es precisamente el óptimo pero es construido en un tiempo considerable [10, 170].

Estos algoritmos usualmente están basados en el algoritmo fundamental de Hunt de aprendizaje conceptual [75]. Este algoritmo incorpora un método utilizado por el ser humano cuando aprende conceptos simples, es decir, encontrar características claves distintivas entre dos categorías, representadas por instancias de entrenamiento positivas y negativas. El algoritmo de Hunt se basa en la estrategia de divide y vencerás, la tarea es dividir el conjunto $\boldsymbol{D}$ que consta de n instancias que pertenecen a c clases, en subconjuntos disjuntos que crean una partición de los datos en subconjuntos que contienen instancias de una sola clase. En el Algoritmo 4 se resume el algoritmo de Hunt.

Algoritmo 4 Algoritmo de Hunt

Entrada: $\boldsymbol{D}$ conjunto de entrenamiento
Salida: Árbol de decisión

1: Seleccionar el atributo más significativo
2: Ubicado en el atributo raíz, dividir todo el subconjunto D, en subconjuntos a través de una función seleccionada.
3: De forma recursiva, se busca el atributo más significativo para cada subconjunto generado en el paso 2 y luego dividir de arriba hacia abajo en subconjuntos. Si cada subconjunto contiene instancias que pertenecen a una sola clase (nodo hoja), detener, de lo contrario, ir al paso 3.

Algoritmos como ID3 [132], C4.5 [153] y CART [16] se basan en la idea del algoritmo de Hunt para construir un DT. Una parte fundamental para la construcción de los árboles es el criterio de división que mejor separe un conjunto de datos dado. Se han propuesto diversas medidas de selección. Estas medidas proporcionan una puntuación para cada atributo que describe las instancias del conjunto de entrenamiento [170]. El atributo que tiene la mejor puntuación de acuerdo a determinada medida, es elegido como atributo de división. Las

medidas de división más utilizadas son: entropía, razón de ganancia, clasificación errónea e índice de Gini.

4.1.4. MÁQUINAS DE SOPORTE VECTORIAL

Las Máquinas de Soporte Vectorial (SVMs, *Support Vector Machines*) son algoritmos de aprendizaje supervisado, propuestos por Vladimir Vapnik [32]. Las SVMs tienen sus orígenes en la teoría del aprendizaje estadístico y han mostrado resultados empíricos prometedores en muchas aplicaciones prácticas, además de trabajar muy bien con datos de alta dimensionalidad [170].

El objetivo de un algoritmo SVM es utilizar un mapeo no lineal para transformar los datos de entrenamiento originales en una dimensión más alta. Dentro de esta nueva dimensión, busca el hiperplano de separación óptimo lineal (límite de decisión) separando las instancias de una clase de la otra. Con un mapeo no lineal apropiado a una dimensión suficientemente alta, los datos de dos clases pueden estar separados por un hiperplano [72].

Considere un problema de clasificación binario, dado un conjunto de entrenamiento de pares de tipo instancia-etiqueta $(\mathbf{x}_i, c_i), i = 1, ..., n$ donde $\mathbf{X} \in \mathbb{R}^n$ y $c_i \in \{1, -1\}$. En una forma simple, SVM busca el hiperplano que separe los datos de entrenamiento con el margen más amplio. Las instancias de entrenamiento que se encuentren más cerca al hiperplano se denominan vectores de soporte. En una forma más general, SVM permite proyectar los datos de entrenamiento en el espacio $\mathbf{X}$ a un espacio de características de mayor dimensión a través de un operador denominado como kernel de Mercer K.

Un algoritmo SVM requiere obtener la solución a la Ecuación 4.10, que representa a un hiperplano que divida dicho conjunto de entrenamiento, donde las instancias con la misma etiqueta queden al mismo lado del hiperplano.

$$\begin{array}{ll} \text{mín}_{w,b,\xi} & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ \text{sujeto a:} & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{array} \tag{4.10}$$

Donde el vector $\mathbf{x}_i$ es transformado en un espacio de dimensión más alta (quizás infinita) por la función ϕ. SVM encuentra un hiperplano de separación lineal con el margen máximo de este espacio de dimensión superior con un costo de error $C > 0$.

La mayoría de los problemas de la vida real suelen ser más complejos y un algoritmo SVM debe tratar con más de dos variables, curvas no lineales de separación, datos que pueden ser no separables y/o clasificación con más de dos etiquetas de clase. Por lo tanto la diversificación de las funciones kernel resulta ser pieza importante en los algoritmos SVM. Algunas de las funciones kernel que más se utilizan son:

1. Lineal: $K(x_i, x_j) = x_i^T x_j$.
2. Polinomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$.
3. Función de base radial (RFB): $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$.
4. Sigmoidal: $K(x_i, x_j) = tanh(\gamma x_i^T, x_j + r)$

Donde, γ, r y d son parámetros propios de las funciones kernel.

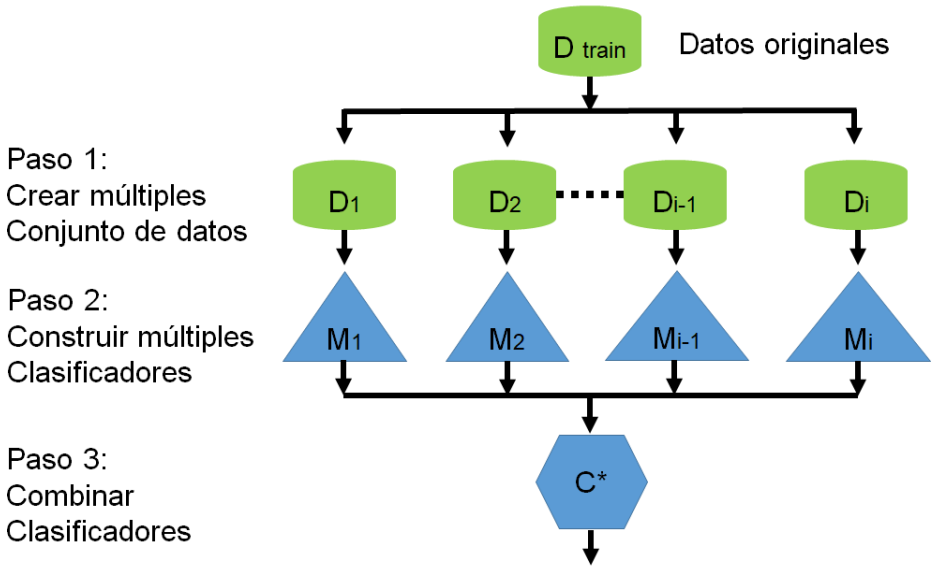


Figura 4.3: Representación lógica del método de aprendizaje por ensambles [170].

4.1.5. ENSAMBLE

Un ensamble para clasificación se puede definir como un modelo predictivo compuesto por una combinación ponderada de múltiples modelos de clasificación [170, 193]. En la literatura relacionada se citan diversas formas para construir un ensamble. Sin embargo una de las más utilizadas es a través de la manipulación del conjunto de entrenamiento [68]. El procedimiento general para construir un ensamble se describe a continuación, donde el primer paso es crear un conjunto de entrenamiento a partir de los datos originales $\boldsymbol{D}_{train}$.

Dependiendo del método utilizado para el ensamble, los conjuntos de entrenamiento pueden ser idénticos o ligeras modificaciones de $\boldsymbol{D}_{train}$. Posteriormente se construye un clasificador base M_i, por cada conjunto de entrenamiento $\boldsymbol{D}_i$ y finalmente cada instancia del conjunto de prueba es clasificada mediante la combinación de las predicciones hechas por cada clasificador o modelo. La etiqueta de clase es obtenida a través de un voto mayoritario en las predicciones individuales de cada clasificador base. El procedimiento general para la construcción de un ensamble de forma secuencial se resume en el Algoritmo 5, así como en la Figura 4.3 se muestra la representación lógica.

		Valor Predicho	
		C+	C-
Valor Actual	C+	VP Verdadero Positivo	FN Falso Negativo
	C-	FP Falso Positivo	VN Verdadero Negativo

Figura 4.4: Matriz de confusión para un problema de clasificación binario. Se muestran las instancias positivas correctamente predichas (VP), las instancias positivas incorrectamente predichas (FN), las instancias negativas incorrectamente predichas (FP) y las instancias negativas correctamente predichas (VN).

Algoritmo 5 Procedimiento general de un ensamble

Entrada: $\boldsymbol{D}_{train}$ conjunto de entrenamiento original, k número de clasificadores base, $\boldsymbol{D}_{test}$ conjunto de prueba

1: **for** $i = 1$ hasta k **do**
2: Crear un conjunto de entrenamiento, $\boldsymbol{D}_i$ de $\boldsymbol{D}_{train}$
3: Construir un clasificador base M_i para $\boldsymbol{D}_i$
4: **end for**
5: **for** cada instancia de prueba $\mathbf{x} \in \boldsymbol{D}_{test}$ **do**
6: $M^*(\mathbf{x}) = voto(M_1(\mathbf{x}), M_2(\mathbf{x}), ..., M_k(\mathbf{x}))$
7: **end for**

Bagging [15] y Boosting [56] son los métodos más representativos para la construcción de un ensamble de clasificación. En este trabajo se hizo uso de dos algoritmos de tipo Boosting que son de la familia de algoritmos AdaBoost [157]: AdaBoostM1 [58, 59] que construye ensambles para problemas de clasificación binaria y AdaBoostM2 [57] que es una extensión del primero para problemas de clasificación de múltiples clases.

4.2. MEDIDAS DE EVALUACIÓN Y VALIDACIÓN

Antes de describir las medidas para evaluar un modelo construido a partir de un conjunto de datos, es necesario definir lo que es un modelo. Un modelo se puede definir como *una descripción de las relaciones causales entre las variables de entrada y salida* [27]. Cuando se genera un modelo siempre existe un valor de error asociado a éste. El error del modelo se calcula como la diferencia entre el valor verdadero observado y el valor de salida del modelo. En términos de clasificación, la evaluación del rendimiento de un modelo de clasificación se basa en la cantidad de instancias de prueba correcta e incorrectamente predichas por el modelo [170]. Estas cantidades se tabulan en una tabla conocida como matriz de confusión. La Figura 4.4 muestra una matriz de confusión para un problema de clasificación binaria. A partir de la matriz de confusión se puede calcular un solo valor numérico por medio de algunas medidas de rendimiento como la exactitud (AC) y el error de clasificación (ER) que se describen en las Ecuaciones 4.11 y 4.12.

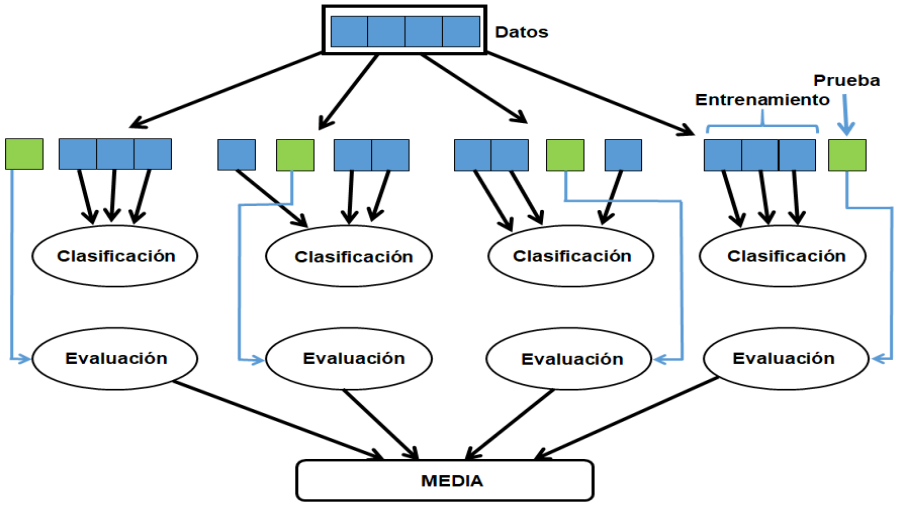


Figura 4.5: Validación cruzada con $k = 4$ para un solo modelo de clasificación.

$$AC = \frac{\text{No. de predicciones correctas}}{\text{Total de predicciones}} = \frac{VP + VN}{VP + FN + FP + VN} \tag{4.11}$$

$$ER = \frac{\text{No. predicciones incorrectas}}{\text{Total de predicciones}} = \frac{FN + FP}{VP + FN + FP + VN} \tag{4.12}$$

Cuando se genera un modelo de datos, se dice que el modelo se ajusta a los datos. Sin embargo, además de ajustar el modelo a los datos también se busca que dicho modelo sirva para la predicción de futuras instancias desconocidas. Cuando se generan varios modelos es necesario validarlos, no sólo por su bondad de ajuste, sino también por su bondad de predicción. En la literatura sobre aprendizaje automático, a la bondad de predicción de un modelo se le denomina ***error de generalización***. Uno de los métodos de validación más simple y probablemente más utilizado para estimar el error de predicción de un modelo es la validación cruzada (CV, *Cross Validation*), que se describe en el siguiente apartado.

4.2.1. VALIDACIÓN CRUZADA

La validación cruzada es un método de validación que divide un conjunto de datos en dos grupos, uno utilizado para entrenar un modelo y el otro para validar el modelo [141]. Uno de los tipos de validación cruzada que más se utiliza es el de k iteraciones, en la Figura 4.5 se presenta un ejemplo del funcionamiento de CV.

La validación cruzada (CV, *Cross Validation*) de k iteraciones funciona de acuerdo al siguiente procedimiento:

1. Dado un conjunto de entrenamiento $\boldsymbol{D}_{train}$ de n instancias, dividir el conjunto de entrenamiento $\boldsymbol{D}_{train}$ en k muestras separadas de aproximadamente el mismo tamaño, $\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_k$.

2. Por cada muestra $\boldsymbol{D}_i$ de tamaño $\frac{n}{k}$, se construye un clasificador M_i usando los datos restantes $\boldsymbol{D}_l = \cup_{j \neq i} \boldsymbol{D}_j$. Después, cada clasificador es evaluado en el subconjunto actual, dicho resultado es almacenado para calcular el riesgo empírico. La Ecuación 4.13, representa la función del riesgo empírico que se construye sobre el conjunto o subconjunto de entrenamiento compuesto de n instancias, donde L es una función de coste o pérdida (el error de clasificación puede ser usada como función de pérdida), w es un conjunto de parámetros del clasificador que se está entrenando y x_i los datos que predicen la variable y_i.

$$E(w) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i, w)) \tag{4.13}$$

3. Calcular la estimación de la predicción final, promediando las sumas del riesgo empírico $E(w)$ para $\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_k$, mediante la Ecuación 4.14.

$$CV = \frac{1}{k} \sum_{i=1}^{k} E(w) \tag{4.14}$$

5

OPTIMIZACIÓN

Si buscas resultados distintos, no hagas siempre lo mismo.

—Albert Einstein

La optimización es considerada como una herramienta en el área de la toma de decisiones y en el análisis de sistemas físicos. La optimización es útil para resolver diversos problemas de la vida real, por ejemplo un inversionista busca crear portafolios que eviten el riesgo excesivo mientras logra una tasa alta de rendimiento, un fabricante desea lograr la máxima eficiencia en el diseño y operación de sus procesos, un ingeniero ajusta los parámetros para optimizar el rendimiento de sus diseños, entre otros [115]. Para llevar a cabo un proceso de optimización, es necesario identificar una función objetivo y una medida cuantitativa del rendimiento del sistema en estudio. La función objetivo depende de ciertas características del sistema, conocidas como variables. La meta principal de la optimización, es encontrar valores para dichas variables (a menudo restringidas) que optimicen la función objetivo establecida. El proceso de identificación de objetivos, variables y restricciones para un problema dado se le conoce como *modelado*. Dentro de la literatura relacionada se dice que se pueden encontrar tres clases de problemas de optimización que son:

1. **Problemas de optimización combinatoria**. Son problemas que se definen en un espacio discreto finito (o infinito), un ejemplo muy conocido de este tipo es el problema del agente viajero.

2. **Problemas de optimización numérica**. Son problemas que se definen en espacios complejos que son subespacios de espacios numéricos como los vectores reales ($\mathbf{x} \in \mathbb{R}^n$) o complejos ($\mathbf{x} \in \mathbb{C}^n$). Problemas numéricos definidos sobre $\mathbb{R}^n$ (o cualquier tipo de espacio que lo contenga) son llamados problemas de optimización continua. Funciones de optimización numérica restringida [38, 125], optimización de problemas de ingeniería [4, 120], tareas de clasificación y minería de datos [127] son considerados como problemas comunes de optimización continua.

3. **Problemas mixtos**. Los problemas mixtos son conocidos como *programación lineal entera mixta*, donde los problemas son modelados en espacios discretos y continuos.

En este documento se hace referencia a problemas de optimización numérica que tienden a ser tratados como problemas de programación lineal entera mixta, de acuerdo al algoritmo de búsqueda que se utilice. En el contexto de la función objetivo, cuando un problema de optimización que modela algún sistema físico involucra sólo una función objetivo, la tarea de encontrar la solución óptima se conoce como optimización *mono-objetivo*. Por otro lado, cuando un problema de optimización involucra más de una función objetivo, la tarea de encontrar una o más soluciones óptimas es conocida como optimización *multi-objetivo*.

5.1. OPTIMIZACIÓN GLOBAL

La optimización global es una área importante de las matemáticas aplicadas y el análisis numérico que se enfoca en encontrar el mejor conjunto de condiciones admisibles para lograr un objetivo. Asumiendo que se resolverán problemas de optimización numérica, en términos matemáticos, la optimización global se define a continuación [182].

Definición 7 *La optimización global es la minimización o maximización de una función sujeta a restricciones en sus variables, formulada en las Ecuaciones 5.1- 5.4.*

$$\textit{Minimizar/Maximizar} \quad f(\mathbf{x}) \tag{5.1}$$

Sujeta a:

$$l_i \leq x_i \leq u_i \tag{5.2}$$

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, ..., J, \tag{5.3}$$

$$h_k(\mathbf{x}) = 0, \quad k = 1, ..., K \tag{5.4}$$

Donde $\mathbf{x} = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n$ *representa un vector solución y* $f(\mathbf{x})$ *es la función objetivo. La Ecuación 5.2 representa la restricción del límite de los valores que puede tomar una variable, donde cada variable* x_i *es limitada por un límite inferior y un límite superior que define el espacio de las variables de decisión D. Respecto a las Ecuaciones 5.3 y 5.4 se refieren a las restricciones relacionadas a la función objetivo, donde* $g_j(\mathbf{x})$ *es la j-ésima restricción de desigualdad y* $h_k(\mathbf{x})$ *es la k-ésima restricción de igualdad. Generalmente, las restricciones de igualdad son transformadas en restricciones de desigualdad de la siguiente manera:* $| h_k(\mathbf{x}) | - \epsilon \leq 0$*, donde* ϵ *es un valor de tolerancia permitido [149].*

5.1.1. ALGORITMOS PARA OPTIMIZACIÓN GLOBAL

Generalmente, los algoritmos de optimización global se pueden dividir en dos clases básicas: (1) algoritmos deterministas y (2) probabilistas. Los primeros, para los mismos datos de entrada siempre producirán los mismos resultados. Suelen encontrar una solución óptima para un determinado problema, pero tienen el grave inconveniente de que en problemas reales (que suelen ser NP-duros, en la mayoría de los casos) resulta complejo poderlos aplicar debido a la dificultad de obtener una solución en un tiempo polinomial.

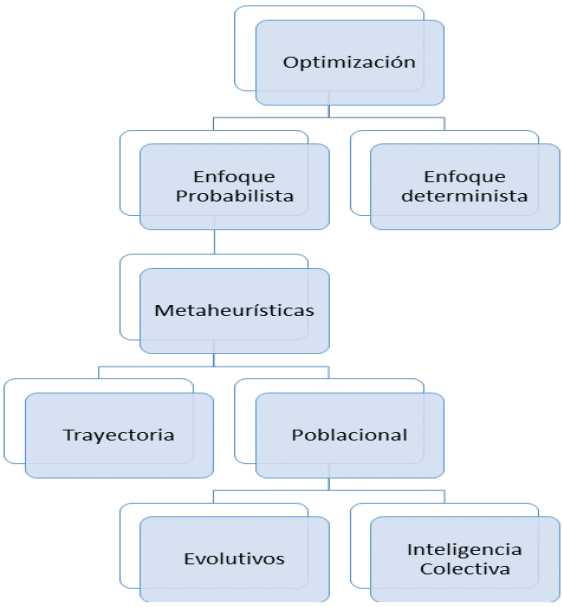


Figura 5.1: Taxonomía de algoritmos de optimización global.

Además, suelen requerir información que no siempre está disponible, por ejemplo algunos requieren la primera o segunda derivada de la función objetivo y en algunos problemas la función no es diferenciable o no se encuentra disponible en forma explicita. Por otro lado, los algoritmos probabilistas, que son considerados algoritmos estocásticos y por ende aproximados, no aseguran obtener la solución óptima global pero sí son capaces de obtener soluciones de alta calidad en un tiempo razonable. Presentan el inconveniente de no ser deterministas, es decir, que pueden producir resultados diferentes en cada ejecución incluso para la misma entrada. Sin embargo, con los algoritmos probabilistas se pueden resolver diversos problemas de la vida real [183]. En la Figura 5.1 se muestra una clasificación general de los algoritmos para resolver problemas de optimización global. Este trabajo se enfoca en el uso de algoritmos de la clase probabilista.

Dentro del enfoque probabilista se sitúan las metaheurísticas, que son esquemas generales que permiten abordar una amplia variedad de problemas adaptándose a las características de cada uno de ellos. Las metaheurísticas son formalmente definidas como un proceso de generación iterativo que guía a una heurística[1] subordinada, combinando de manera inteligente diferentes conceptos para explorar y explotar el espacio de búsqueda con el objeto de encontrar eficientemente soluciones cercanas al óptimo [118].

Al diseñar una metaheurística, se deben tener en cuenta dos criterios: 1) exploración del espacio de búsqueda (*diversificación*) y 2) explotación de las mejores soluciones encontradas (*intensificación*). Una región prometedora está determinada por las buenas soluciones obtenidas. En la intensificación, las regiones prometedoras son explotadas más a fondo con la esperanza de encontrar mejores soluciones, mientras que en la diversificación, se deben

[1] El término heurística proviene del griego heuriskein que significa encontrar o descubrir. La heurística en optimización es una función que califica la utilidad de un elemento y proporciona apoyo para la toma de decisiones.

visitar las regiones no exploradas para asegurar que todas las regiones del espacio de búsqueda se exploren de manera uniforme y que la búsqueda no se limite a un número reducido de regiones.

Existen muchos criterios para clasificar a las metaheurísticas [14, 40], por ejemplo se pueden clasificar respecto al tipo de ruta que siguen durante la búsqueda, el uso de memoria, el tipo de exploración de vecindarios que utilizan o el número de soluciones actuales que son llevadas de una iteración a otra. Sin embargo, una clasificación formal que se considera es dividir a las metaheurísticas en dos categorias que son: las metaheurísticas basadas en una única solución (basadas en trayectoria) y las metaheurísticas basadas en la población [169]. Las metaheurísticas basadas en una sola solución manipulan y transforman una única solución durante la búsqueda, mientras que las basadas en la población se enfocan en ir modificando una población completa de soluciones. Estas dos categorías poseen características complementarias, las metaheurísticas basadas en una única solución están orientadas a la explotación, es decir que tienen el poder de intensificar la búsqueda en regiones locales. Las metaheuristicas basadas en la población están orientadas a la exploración, ya que permiten una mejor diversificación en todo el espacio de búsqueda.

METAHEURÍSTICAS BASADAS EN TRAYECTORIA

Las metaheurísticas basadas en trayectoria aplican iterativamente los procedimientos de generación y reemplazo de la única solución actual. En la fase de generación, se genera un conjunto de soluciones candidatas a partir de la solución actual $\mathbf{s}$. Este conjunto $\boldsymbol{S}(\mathbf{s})$ se obtiene generalmente a través de transformaciones locales de la solución. En la fase de reemplazo, se selecciona una solución del conjunto de soluciones $\mathbf{s}' \in \boldsymbol{S}(\mathbf{s})$ que reemplazará a la solución actual. Este proceso se repite hasta satisfacer un determinado criterio, en el Algoritmo 6 se presenta el procedimiento general que sigue una metaheurística basada en trayectoria.

Algoritmo 6 Procedimiento general de una metaheurística basada en trayectoría.

Entrada: Solución inicial s_0
Salida: Mejor solución encontrada

1: $t = 0$
2: **repeat**
3: Generar $C(s_t)$ /* *Generar soluciones candidatas de* s_t */
4: $s_{t+1} = Seleccionar(C(s_t))$ /* *Seleccionar la solución que reemplazará la solución actual* */
5: $t = t + 1$
6: **until** Se satisface un criterio de paro

Los conceptos comunes para todas las metaheuristicas basadas en trayectoria son la definición del vecindario de soluciones candidatas y la generación de la solución inicial. Un vecindario se compone por un conjunto de soluciones candidatas $s' \in N(S)$, cada solución candidata es denominada como solución vecina que es generada por la aplicación de un operador de movimientos m que realiza una pequeña perturbación a la solución actual s. Un vecindario se caracteriza por tener la propiedad de localidad, dicha propiedad es el efecto sobre la solución al momento de realizar el movimiento de perturbación en la representación. Cuando se realizan cambios en la representación se visualizan dos casos. El primero es

cuando se aplican un cambio pequeño, entonces se dice que el vecindario posee una localidad fuerte. El segundo, cuando se realiza un cambio grande, se dice que el vecindario tiene una localidad débil. La definición del vecindario depende en gran parte de la representación asociada al problema a solucionar. La búsqueda local, búsqueda local guiada, búsqueda local iterada, búsqueda tabú, recocido simulado, entre otros, son ejemplos de metaheurísticas basadas en trayectoria que han sido utilizadas para solucionar múltiples problemas.

METAHEURÍSTICAS BASADAS EN LA POBLACIÓN

Las metaheurísticas basadas en la población comienzan con un conjunto inicial de soluciones. Luego, aplican iterativamente la generación de una nueva población y el reemplazo de la población actual. En la fase de generación, se crear una nueva población de soluciones a partir de la aplicación de diversos operadores. En la fase de reemplazo, se lleva a cabo una selección entre la población actual y la nueva. Este proceso itera hasta que se satisface un determinado criterio de paro. El Algoritmo 7 ilustra el procedimiento general que siguen las metaheurísticas basadas en la población.

Algoritmo 7 Procedimiento general de una metaheurística basada en la población.

Entrada: $\boldsymbol{P} = \boldsymbol{P}_0$ /* *Generación de la población inicial* */

1: $t = 0$

Salida: Mejor solución encontrada

2: **repeat**

3: Generar($\boldsymbol{P}'_t$) /* *Generación de la nueva población* */

4: $\boldsymbol{P}_{t+1}$ = Seleccionar-Población($\boldsymbol{P}_t \cup \boldsymbol{P}'_t$) /* *Seleccionar una nueva población* */

5: $t = t + 1$

6: **until** Se satisface un criterio de paro

Las metaheurísticas basadas en población se encuentran clasificadas dos familias importantes de algoritmos: Algoritmos Evolutivos (EAs, *Evolutionary Algorithms*) y Algoritmos de Inteligencia Colectiva (SIAs, *Swarm Intelligence Algorithms*). Ambas familias de algoritmos pertenecen al área de Cómputo Inteligente inspirado en la naturaleza. Los EAs, son algoritmos estocásticos que han sido ampliamente aplicados para resolver una gran cantidad de problemas.

Representan una clase de algoritmos de optimización que emulan la evolución de las especies. Algunos de los algoritmos representativos son: el Algoritmo Genético (GA, *Genetic Algorithm*) [65], Programación Evolutiva (EP, *Evolutionary Programming*) [54], Estrategias Evolutivas (ES, *Evolution Strategies*) [11], Programación Genética (GP, *Genetic Programming*) [89], Algoritmos de Estimación de Distribución (EDAs, *Estimation of Distribution Algorithms*) y Algoritmo de Evolución Diferencial (DE, *Differential Evolution*) [128, 166]. Los SIAs estan inspirados en el comportamiento de sistemas de cúmulos naturales. Un cúmulo es una gran cantidad de agentes simples y homogéneos que interactúan localmente entre ellos y su entorno para lograr tareas de forma cooperativa. Ejemplos de SIAs son: Optimización basada en Colonia de Hormigas (ACO, *Ant Colony Optmization*) [41], Optimización por Cúmulo de Partículas (PSO, *Particle Swarm Optimization*) [80], Colonia Artificial de Abejas (ABC, *Artficial Bee Colony*) [79], Forrajeo de Bacterias (BF, *Bacterial Foraging*) [122], entre otros.

5.2. OPTIMIZACIÓN MULTI-OBJETIVO

Existen diversos dominios (p.e. la ingeniería, bioinformática, logística, transporte, telecomunicaciones, medio ambiente, aeronáutica o finanzas) que se enfrentan a problemas complejos de optimización que pueden implicar diversos criterios, denominados como problemas de optimización multi-objetivo (MOPs, *Multi-Objective Optimization Problems*) [35, 194]. Esta clase de problemas son tratados por el área de la optimización multi-objetivo, dicha área tiene sus orígenes durante el siglo XIX con el trabajo de Edgeworth y Pareto en economía. A lo largo de los años, la optimización multi-objetivo se ha convertido en un área importante dentro de la ciencia y la ingeniería.

En esta sección se describen los conceptos básicos acerca de la optimización multi-objetivo además de la descripción general de los algoritmos que se han propuesto para dar solución a los MOPs.

5.2.1. CONCEPTOS BÁSICOS

La optimización multi-objetivo, a diferencia de la optimización global, se enfoca en encontrar un conjunto de soluciones denominadas como soluciones óptimas de Pareto. Este conjunto de soluciones óptimas representan el compromiso entre los diferentes objetivos en conflicto. Por lo tanto, el principal objetivo al solucionar un MOP es obtener el conjunto óptimo de Pareto y, en consecuencia, el frente de Pareto. A continuación se describen los conceptos principales relacionados con la optimización multi-objetivo, como la dominancia, la optimalidad de Pareto y el frente de Pareto. En estas descripciones, se asume minimización en todos los objetivos involucrados. A continuación se presentan algunas definiciones importantes para comprender el concepto optimización multi-objetivo [35].

Definición 8 *Un MOP involucra n variables de decisión y una serie de m funciones objetivo, en la Ecuación 5.5 se expresa un MOP en su forma general.*

$$\begin{array}{ll} \textit{Minimizar} & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x})]^T \\ \textit{sujeto a:} & \mathbf{x} \in \boldsymbol{S} \end{array} \tag{5.5}$$

Donde $m (m \geq 2)$ es el número de objetivos, $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ es el vector de las variables de decisión, $\boldsymbol{S}$ representa el conjunto de soluciones factibles con restricciones de igualdad y desigualdad y límites explícitos y $\mathbf{f}(\mathbf{x})$ es el vector de objetivos a ser optimizados.

El espacio de búsqueda $\boldsymbol{S}$ es conocido como espacio de decisión del MOP, mientras que el espacio al que pertenece el vector de objetivos es denominado como espacio objetivo. Por cada solución $\mathbf{x}$ en el espacio de decisiones, existe un punto (vector de objetivos correspondientes) $\mathbf{z} \in \mathrm{R}^m$ en el espacio objetivo, denotado por $\mathbf{f}(\mathbf{x}) = \mathbf{z} = z_1, z_2, ..., z_m]^T$, ver Figura 5.2.

En optimización multi-objetivo no existe una solución óptima única, sino más bien un conjunto de soluciones alternativas. Las soluciones óptimas se pueden definir a partir de un concepto matemático de ordenamiento parcial, donde el término de *dominancia de Pareto* es utilizado para dicho propósito.

Definición 9 ***Dominancia de Pareto.*** *Una solución $\mathbf{x}^{(1)}$ domina a otra solución $\mathbf{x}^{(2)}$, denotado por $(\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)})$ si se cumplen las siguientes condiciones:*

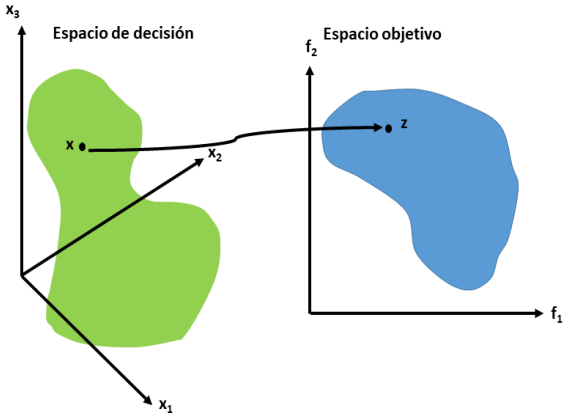


Figura 5.2: Representación del espacio de decisión y el objetivo correspondiente [38].

1. *Si la solución* $\mathbf{x}^{(1)}$ *no es peor que* $\mathbf{x}^{(2)}$ *en todos los objetivos.*
2. *Si la solución* $\mathbf{x}^{(1)}$ *es estrictamente mejor que* $\mathbf{x}^{(2)}$ *en al menos un objetivo.*

La dominancia de Pareto de puede resumir en la Ecuación 5.6.

$$\forall_{i\in\{1,\ldots,m\}} : f_i(\mathbf{x}^{(1)}) \leq f_i(\mathbf{x}^{(2)}) \wedge \exists_{i\in\{1,\ldots,m\}} : f_i(\mathbf{x}^{(1)}) < f_i(\mathbf{x}^{(2)}) \tag{5.6}$$

Se dice que estas soluciones son óptimas en el sentido más amplio ya que ninguna otra solución, en el espacio de búsqueda, es superior a ellas cuando se consideran todos los objetivos en conflicto, estas soluciones son conocidas como soluciones óptimas de Pareto.

Definición 10 ***Optimalidad de Pareto.*** *Una solución* $\mathbf{x}^* \in \boldsymbol{S}$ *es denominada como óptima de Pareto si y solo si no existe otra solución en* $\boldsymbol{S}$ *que domine a* $\mathbf{x}^*$*, tal como se expresa en la Ecuación 5.7.*

$$\nexists \mathbf{x}' \in \boldsymbol{S} : \mathbf{x}' \preceq \mathbf{x}^* \tag{5.7}$$

Al resolver un MOP se pueden encontrar un conjunto de soluciones, conocido como el conjunto óptimo de Pareto. Gráficamente, este conjunto en el espacio objetivo se conoce como el frente de Pareto, por ejemplo el que se muestra en la Figura 5.3.

Definición 11 ***Conjunto óptimo de Pareto.*** *Para un determinado problema multi-objetivo, el conjunto óptimo de Pareto es definido como* $\boldsymbol{P}^*$.

Definición 12 ***Frente de Pareto.*** *Para un determinado problema multi-objetivo y su conjunto óptimo de Pareto* $\boldsymbol{P}^*$*, el frente de Pareto es definido como* $\boldsymbol{PF}^* = \{\mathbf{f}(\mathbf{x}) : \mathbf{x} \in \boldsymbol{P}^*\}$

Un frente de Pareto obtenido mediante una búsqueda debe contener al menos soluciones, que deben estar lo más cerca posible al frente de Pareto verdadero, así como también deben estar uniformemente localizadas sobre el frente de Pareto. De otro modo, la aproximación del frente obtenido puede resultar poco útil para un tomador de decisiones. Cuando

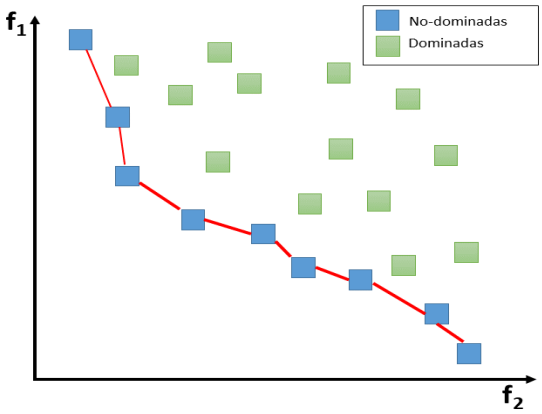


Figura 5.3: Ejemplo de un frente de Pareto.

se resuelve un MOP existen algunas soluciones especiales que son consideradas por algunos algoritmos de búsqueda, estas soluciones especiales son: el vector ideal, el punto de referencia y punto nadir.

Definición 13 ***Vector ideal.*** *Un punto* $\mathbf{y}^* = (y_1^*, y_2^*, ..., y_m^*)$ *es considerado como un vector ideal si minimiza cada función objetivo* f_i *en* $\mathbf{f}(\mathbf{x})$, $\mathbf{y}^* = \text{mín}(f_i(\mathbf{x})), \mathbf{x} \in \mathbf{S}, i \in [1, m]$.

Generalmente, el vector ideal es una solución utópica, ya que no es una solución factible en el espacio de decisión. En algunos casos, el encargado de tomar decisiones, define un vector de referencia que representa el objetivo a alcanzar por cada objetivo involucrado. Se pueden especificar niveles de aspiración z_i, $i \in [1, m]$ por cada función objetivo.

Definición 14 ***Punto de referencia.*** *Un punto de referencia* $\mathbf{z}^* = [z_1, z_2, ..., z_m]$, *es un vector que define el nivel de aspiración* z_i *a alcanzar por cada objetivo* f_i.

Definición 15 ***Punto nadir.*** *El punto nadir* $\mathbf{z}^{\mathbf{nad}}$ *representa una solución no deseada que representa el límite máximo de cada objetivo en el conjunto óptimo de Pareto completo.*

El punto ideal y el punto nadir proporcionan información acerca de los rangos del frente óptimo de Pareto, como se puede observar en la Figura 5.4.

5.2.2. BÚSQUEDA Y TOMA DE DECISIONES

El principal objetivo al solucionar un determinado problema multi-objetivo es ayudar al responsable de la toma de decisiones a encontrar una solución de Pareto que logre cubrir sus expectativas y preferencias. Uno de los aspectos fundamentales en la resolución del MOP está relacionado con la interacción entre el método de búsqueda (p.e., la metaheurística) y el tomador de decisiones. Para esta interacción, se han propuesto diferentes enfoques, tal es el caso de los propuestos por Hwang y Masud [101], quienes proponen las siguientes cuatro categorías:

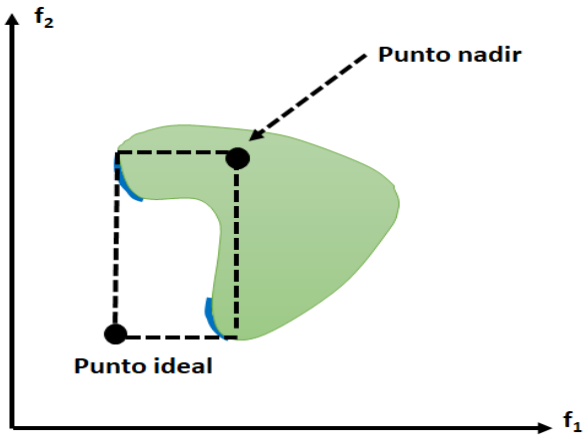


Figura 5.4: Representación gráfica del punto ideal y punto nadir.

1. ***Métodos sin preferencias.*** Esta clase de métodos no asume ningún tipo de información acerca de la importancia de los objetivos. Usualmente, se utiliza una heurística para encontrar una solución única y óptima.

2. ***Métodos a priori.*** En este enfoque, el encargado de la toma de decisiones proporciona información de sus preferencias antes del proceso de optimización. Existen diferentes métodos para la formulación de las preferencias, algunos métodos sugieren combinar las diferentes funciones objetivo de acuerdo a alguna función de utilidad v, con el propósito de obtener, solamente, una función objetivo a ser optimizada.

3. ***Métodos a posteriori.*** En esta categoría, el proceso de búsqueda determina un conjunto de soluciones óptimas de Pareto. Este conjunto de soluciones ayuda al tomador de decisiones a tener un conocimiento completo del frente de Pareto. De cierta manera, el frente de Pareto constituye la adquisición de conocimiento acerca del problema que se esté resolviendo. Posteriormente, se tendrá que seleccionar una solución del conjunto de soluciones proporcionadas por el algoritmo de búsqueda.

4. ***Métodos interactivos.*** En esta clase de métodos, existe una interacción progresiva entre el algoritmo de búsqueda y el tomador de decisiones. A partir de la información obtenida durante la resolución del problema, el que toma las decisiones define sus preferencias de una manera comprensible, por ejemplo, la selección de un punto de referencia, una dirección de referencia, de acuerdo a una tasa de retorno, entre otras. El algoritmo de búsqueda toma en cuenta la definición de las preferencias en la resolución del problema y el proceso itera durante un determinado número de etapas hasta obtener un frente de Pareto que representa un compromiso aceptable para el que toma las decisiones.

5.2.3. ALGORITMOS PARA OPTIMIZACIÓN MULTI-OBJETIVO

En la literatura especializada se pueden encontrar un gran número de algoritmos que resuelven problemas multi-objetivo. En esta sección se presenta una clasificación de dichos

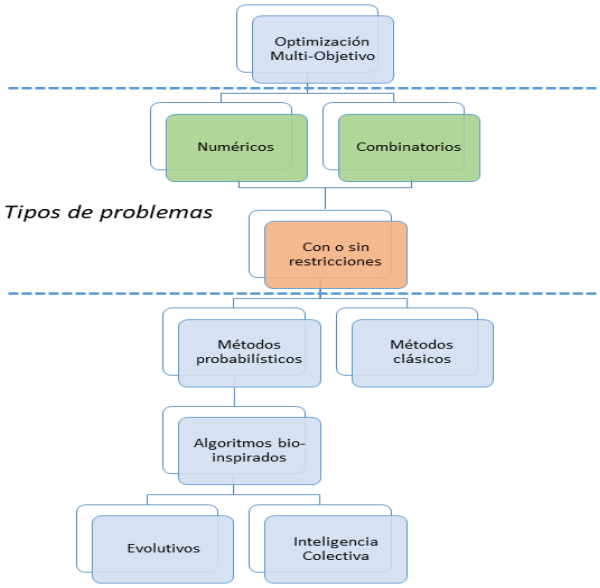


Figura 5.5: Taxonomía de métodos de optimización multi-objetivo.

algoritmos. Los problemas que se resuelven en optimización multi-objetivo se dividen en dos categorías: a) problemas numéricos, donde las soluciones son codificadas con variables reales, también conocidas como variables continuas y b) problemas combinatorios en los que las soluciones son codificadas mediante el uso de variables discretas. Estas dos categorías de problemas pueden presentar o carecer de algún tipo de restricción. A partir de esta división se desglosan dos clases de métodos de optimización multi-objetivo que son: a) los métodos clásicos y b) los métodos probabilísticos, ver Figura 5.5.

Antes de describir las dos clases de métodos multi-objetivo, se presentan a continuación, tres aspectos importantes que se deben de considerar en el proceso de búsqueda de cualquier algoritmo multi-objetivo.

- *Asignación de aptitud.* Representa una tarea importante que guía la búsqueda del algoritmo hacia soluciones óptimas de Pareto. En general, la asignación de aptitud se puede llevar a cabo mediante agregación, basada en criterios y basada en el frente de Pareto.

- *Preservación de la diversidad.* El principal objetivo de este aspecto, es generar un conjunto diverso de soluciones de Pareto en el espacio objetivo.

- *Elitismo.* En términos generales, el elitismo consiste en conservar las mejores soluciones generadas durante la búsqueda.

MÉTODOS CLÁSICOS

Los métodos clásicos se caracterizan por transformar un problema multi-objetivo en un problema mono-objetivo o en un conjunto de problemas mono-objetivo. Esta clase de métodos requieren que el tomador de decisiones posea conocimiento del problema que se esté resolviendo. Algunos métodos representativos se describen a continuación:

- ***Método de suma ponderada.*** Este método es uno de los primeros y mayormente usados para la generación de soluciones óptimas de Pareto. La idea general de este método es asociar un peso w_i a cada función objetivo f_i y minimizar la suma ponderada de los M objetivos, por tal motivo, se dice que este método transforma un problema multi-objetivo en uno mono-objetivo, mediante la combinación de varias funciones objetivo f_i en una función objetivo única f, tal como se expresa en la Ecuación 5.8. Una de las principales desventajas del método de suma ponderada, es la dificultad de encontrar soluciones en regiones no-convexas del frente de Pareto [191].

$$\min_{x \in \mathbb{X}} \sum_{i=1}^{M} w_i f_i(\mathbf{x}) \tag{5.8}$$

- ***Método de la restricción ϵ.*** Este método surge para tratar la desventaja de trabajar con frentes no convexos que presenta el método de suma ponderada. El método de la restricción ϵ consiste en mantener un solo objetivo y restringir el resto de los objetivos a un valor ϵ. En este caso, un problema multi-objetivo es convertido a un problema mono-objetivo. Una posible desventaja de este método es la selección del valor de ϵ, ya que se tiene que asegurar que dicho valor se encuentre dentro del valor mínimo o máximo de las funciones objetivo individuales [189].

- ***Programación por metas.*** En este método se definen puntos de referencia $\mathbf{z_i}$ (niveles de aspiración) por cada función objetivo, que son denominados como metas. Posteriormente, se establece un orden de prioridad entre los objetivos. Para resolver el problema de optimización es necesario incorporar las variables de desviación, dos por cada objetivo, una de ellas es con signo negativo que indica la cantidad que se requiere para llegar a la meta propuesta, mientras que la otra variable tiene un signo positivo que indica la cantidad que se ha excedido respecto a la meta establecida. También presenta problemas para identificar soluciones factibles en regiones no convexas del frente de Pareto [6].

- ***Método lexicográfico.*** Este método asume que los objetivos pueden ser ordenados por importancia. Dicho método consiste en ir resolviendo una serie de problemas mono-objetivo secuencialmente, iniciando por el objetivo que tiene la prioridad más alta hasta el objetivo con la prioridad menor. En cada iteración, el valor óptimo encontrado para cada objetivo es considerado como restricción en los problemas subsecuentes. A través de este procedimiento se puede ir preservando el valor óptimo de los objetivos con mayor prioridad. Los inconvenientes que puede presentar este método es resolver múltiples problemas para obtener una solución además de que los sub-problemas pueden ser difíciles de resolver debido a las restricciones que se van incorporando [6].

Como se puede ver, algunos métodos clásicos de optimización multi-objetivo necesitan de parámetros o conocimiento del rango de los objetivos para ser evaluados, además de ser ejecutados múltiples veces para encontrar un conjunto de soluciones en el frente de Pareto. Ante las limitaciones de los métodos clásicos surgen los métodos alternativos que se describen en la siguiente sección.

MÉTODOS ALTERNATIVOS

Los métodos alternativos surgen ante la necesidad de enfrentar las desventajas que poseen los métodos clásicos, ya que la mayoría de ellos requieren llevar a cabo una transformación del problema multi-objetivo en una serie de sub-problemas mono-objetivo, establecer puntos metas y/o seleccionar límites (inferior y superior) por cada objetivo para poder lograr una solución competitiva. Dentro de la categoría de los métodos alternativos, se sitúan los métodos bio-inpirados, los cuales se basan en emular el comportamiento de sistemas naturales para el desarrollo y diseño de meta-heurísticas para la búsqueda de soluciones de un determinado problema. Para la optimización multi-objetivo, se desglosan dos sub-categorías a partir de los métodos bio-inspirados, que son: a) los Algoritmos Evolutivos Multi-objetivo (MOEAs, *Multi-objective Evolutionary Algorithms*) y b) algoritmos de inteligencia Colectiva para optimización multi-objetivo.

a) **Algoritmos Evolutivos Multi-objetivo.** Este grupo de algoritmos conjugan los conceptos básicos de dominancia con las características generales de los EAs. Los MOEAs son capaces de tratar con espacios no continuos, no convexos y/o no lineales, así como con problemas donde no se conocen explicitamente sus funciones objetivo [35, 97]. En la literatura especializada es posible encontrar diferentes algoritmos de este tipo. Sin embargo, la primera iniciativa de algoritmo evolutivo para resolver problemas multi-objetivo fue el algoritmo VEGA (*Vector Evaluated Genetic Algorithm*) propuesto por Shaffer [156]. Algunos algoritmos representativos de esta categoría son: MOGA (*Multi-Objective Genetic Algorithm*) [55], NPGA (*Niched Pareto Genetic Algorithm*) [71], PESA (*Pareto-envelope Based Selection Algorithm*) [30] y PESA-II [31], SPEA (*Strength Pareto Evolutionary Algorithm*) [195] y su versión mejorada SPEA2 [196], PAES (*Pareto Archived Evolution Strategy*) [85], NSGA (*Non-dominated Sorting Genetic Algorithm*) [164] y NSGA-II [37] y MOEA/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*) [192].

b) ***Inteligencia Colectiva para optimización multi-objetivo.*** Los algoritmos de Inteligencia Colectiva tratan de emular el comportamiento de un cúmulo de agentes descentralizados y auto-organizados. El término cúmulo es utilizado para hacer referencia a cualquier colección restringida de agentes o individuos interactuantes, por ejemplo cúmulo de hormigas, pájaros o abejas [106]. De acuerdo con Millonas [102], existen cinco principios que debe seguir un cúmulo para ser útil en cualquier problema de optimización:

1. Proximidad. El cúmulo debe ser capaz de hacer cálculos simples de espacio y tiempo, para comprender y conceptualizar la información de forma rápida.
2. Calidad. El cúmulo deberá responder a los factores de calidad del entorno, como la calidad de los alimentos o la seguridad del lugar.
3. Respuesta diversa. El cúmulo de debe asignar todos sus recursos en una región estrecha. La distribución debe diseñarse de manera que cada agente esté protegido frente a cambios ambientales.
4. Estabilidad. El cúmulo no debe cambiar su modo de comportamiento ante cada cambio ambiental.

5. Adaptabilidad. El cúmulo debe poder cambiar su comportamiento cuando la inversión de energía vale el costo computacional.

Estos principios pueden ser aplicados tanto para optimización mono-objetivo como para multi-objetivo. Dentro de la categoría de los algoritmos de Inteligencia Colectiva para optimización multi-objetivo, se pueden mencionar algunos de los algoritmos más populares como son: MOPSO (*Multi-objective Particle Swarm Optimization*) [143], MOACO(*Multi-Objective Ant Colony Optimization*) [133], MOBC (*Multi-objective Optimization Bee Colony*) [98].

Recientemente se han propuesto nuevos algoritmos de Inteligencia Colectiva para resolver problemas multi-objetivo: MOFA (*Multi-Objective Firefly Algorithm*) [190], MOCSO (*Multi-Objective Cat Swarm Optimization*) [12], MOBA (*Multi-Objective Bat Algorithm*) [172], MOGWO (*Multi-Objective Grey Wolf Optimizer*) [104], MOALO (*Multi-Objective Ant Lion Optimizer*) [105], MOMFO (*Multi-Objective Moth Flame Optimization*) [177] y MODA (*Multi-Objective Dragonfly Algorithm*) [103].

III

Experimentos y Resultados

6

Hacia la selección del modelo completo en bases de datos temporales mediante micro-Evolución Diferencial

> Lo importante en la ciencia no es tanto obtener nuevos datos, sino descubrir nuevas formas de pensar sobre ellos.
>
> —William Lawrence Bragg

En este capítulo se describe un enfoque propuesto para dar solución al problema de FMS en bases de datos temporales a través del conocido algoritmo de Evolución Diferencial (DE, *Differential Evolution*) [166], en su versión micro.

6.1. Antecedentes

DE es uno de los EAs, cuyo desempeño en la resolución de problemas de diferentes dominios ha sido notable [113]. DE es clasificado dentro de las metaheurísticas poblacionales, ya que a partir de un conjunto de soluciones, usualmente generadas de forma aleatoria, van evolucionando durante un determinado número de ciclos. Dentro de la literatura relacionada es posible encontrar una gama especial de algoritmos poblacionales que se caracterizan por trabajar con poblaciones pequeñas, estos algoritmos son conocidos como micro algoritmos evolutivos, denotados por el prefijo μ [121]. Se dice que los μ-EAs fueron creados para solucionar problemas muy simples y con propósitos educativos. Sin embargo se han hecho diversos esfuerzos por usarlos en aplicaciones más completas y exigentes. Por ejemplo Koppen [86], propuso el uso de un μ-GA para resolver problemas de procesamiento de imágenes. Por otro lado se ha propuesto el uso de μ-PSO para tratar con problemas de alta dimensionalidad [74] o problemas multi-objetivo [20]. Versiones de μ-DE se han propuesto para encontrar el umbral de una imagen durante su pre-procesamiento [134]. De acuerdo con Caraffini

[23], los algoritmos con poblaciones pequeñas se caracterizan por llevar a cabo más explotación que los algoritmos con poblaciones más grandes. Por lo tanto, los μ-EAs son capaces de encontrar rápidamente soluciones prometedoras en las primeras etapas de proceso de optimización. De acuerdo a ésto, en este capítulo se presenta una propuesta donde se utiliza un μ-DE para resolver el problema de FMS en bases de datos temporales donde la evaluación de la función de aptitud es computacionalmente costosa de calcular. Cuatro diferentes variantes de DE son evaluadas con el μ-DE utilizado.

6.2. PROPUESTA: μ-DE PARA LA SELECCIÓN DEL MODELO ($\mu - DEMS$)

μ-DEMS es un enfoque basado en el comportamiento estándar del tradicional algoritmo de DE propuesto por Storn y Price [166]. El algoritmo estándar de DE consta de tres operadores: mutación, cruza y selección, además de un factor de escalamiento (F), la tasa de cruza (CR) y el tamaño de la población (NP), que son los tres principales parámetros de control del algoritmo. A continuación se describen las etapas del DE estándar:

- **Creación de la población inicial**. Se crea una población inicial de vectores $NP_g = [\mathbf{x}_{1,g}, \mathbf{x}_{2,g}, ..., \mathbf{x}_{NP,g}]$ en la generación g, donde cada $\mathbf{x}_{i,g} = [x^1_{i,g}, ..., x^D_{i,g}]^T$ tiene D elementos que son generados aleatoriamente entre los límites inferior y superior predefinidos $L_j \leq x_i \leq U_j$.

- **Operación de mutación**. Por cada vector $\mathbf{x}_{i,g}$ en la población (conocido como vector objetivo), se crea un nuevo vector denominado como vector mutante $\mathbf{v}_{i,g}$ mediante la Ecuación 6.1.

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_3,g} + F \cdot (\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}) \tag{6.1}$$

  Donde tres vectores denotados como r_1, r_2 and r_3 son seleccionados al azar de la población actual como $r_1 \neq r_2 \neq r_3 \neq i$.
  $\mathbf{x}_{r_3,g}$ es el vector base, cuya posición define la ubicación de inicio para definir una dirección de búsqueda $(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g})$. El factor $F > 0$ es un número real que escala el vector de diferencia $(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g})$ que define la dirección de búsqueda a partir del vector base.

- **Operación de cruza**. Tras la mutación se realiza una operación de recombinación sobre cada vector objetivo $\mathbf{x}_{i,g}$ y su vector mutante, para generar un vector descendiente $\mathbf{u}_{i,g}$, que es generado a partir de la Ecuación 6.2.

$$u^j_{i,g} = \begin{cases} v^j_{i,g} & \text{if } rand_j(0,1) \leq CR \text{ or } j = j_{rand} \\ x^j_{i,g} & \text{otherwise} \end{cases} \tag{6.2}$$

  donde $j = 1, ..., D$, $CR \in [0,1]$ es un valor definido por el usuario, dicho valor controla el parecido entre el vector descendiente respecto al vector mutante, $rand_j$ genera un número aleatorio entre 0 y 1 con una distribución uniforme y j_{rand} es un número entero elegido al azar $\in [1, \ldots, D]$ para garantizar que no se duplique el vector objetivo.

- **Selección**. Finalmente, el operador de selección decide de acuerdo a la función de aptitud, si el vector descendiente $\mathbf{u}_{i,g}$ es elegido y reemplaza al vector objetivo $\mathbf{x}_{i,g}$, o por el contrario, el vector descendiente es rechazado y prevalece el vector objetivo para la

siguiente generación. Suponiendo minimización, el operador de selección se rige bajo la expresión presentada en la Ecuación 6.3.

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases} \tag{6.3}$$

En esta propuesta hace uso de un μ-DE, donde la estructura y operaciones siguen siendo las mismas que el DE clásico. Sin embargo μ-DE hace uso de una población pequeña y un mecanismo de reinicio para actualizar la población que converge rápidamente. El Algoritmo 8 muestra los pasos del μ-DE, propuesto en[121], que es utilizado como algoritmo de búsqueda en la solución del problema de FMS en bases de datos temporales.

Algoritmo 8 μ-DE

Entrada: $NP \in [3,6]$ (Tamaño de población), $CR \in [0,1]$ (Tasa de cruza), $F > 0$ (Factor de escalamiento), $N \in \mathbb{N}$ (Número de soluciones a reiniciar), $R \in \mathbb{N}$ (Generación a reemplazar).

1: Set $G = MaxEval/NP$% Número de generaciones
2: Set $Cont = 1$
3: Generar de forma aleatoria una población inicial $NP_0 = [\mathbf{x}_{1,0}, \mathbf{x}_{2,0}, \ldots, \mathbf{x}_{NP,0}]^T$
4: **for** $g = 1$ to G **do**
5: **if** Cont==R **then**
6: Reinicialización de N peores individuos
7: Cont = 1
8: **end if**
9: **for** i=1 to NP **do**
10: Set $r_1 \neq r_2 \neq r_3, \in [1, NP], jrand \in [1, D]$ aleatoriamente
11: **for** j=1 to D **do**
12: **if** $(rand_j[0,1] < CR$ or $j == jrand)$ **then**
13: $u^j_{i,g} = x^j_{r_3,g} + F \cdot (x^j_{r_1,g} - x^j_{r_2,g})$
14: **else**
15: $u^j_{i,g} = x^j_{i,g}$
16: **end if**
17: **end for**
18: **if** $(f(\mathbf{u}_{i,g}) < f(\mathbf{x}_i, g))$ **then**
19: $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$
20: **else**
21: $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$
22: **end if**
23: **end for**
24: $Cont = Cont + 1$
25: $g = g + 1$
26: **end for**

El algoritmo que se propone en este trabajo es μ-DEMS, que toma como base el algoritmo presentado en el Algoritmo 8. Diversad variantes son evaluadas para resolver una instancia del problema de FMS presentada en la Tabla 6.1. Donde dos métodos de suavizados, tres métodos de representación de series de tiempo y el clasificador k-NN con cuatro diferentes medidas de similitud son incluidos en el grupo de métodos disponibles para sugerir un modelo completo en una base de datos determinada.

6.2.1. CODIFICACIÓN DE LA SOLUCIÓN

Cada solución candidata es codificada mediante un vector de once variables como el que se muestra en la Figura 6.1. La variable x_1 puede tomar solo dos posibles valores (1 o 2) que identifica el método de suavizado seleccionado. Las variables x_2 a x_4 representan los valores

Tabla 6.1: Descripción de métodos incluidos en el espacio de búsqueda explorado por $\mu - DEMS$. TLS significa longitud de serie de tiempo, S: suavizado, R: representación y C: clasificación.

ID	Método	Tipo	Hiperparámetros $\in \mathbb{R}$	Descripción
1	SG	S	k $\in [1, TSL]$, f $\in [1, 5]$	Orden polinomial , Tamaño de ventana.
2	Promedio Móvil	S	span $\in [2, TSL]$	Tamaño de ventana.
1	SAX	R	w $\in [2, TSL/2]$, a $\in [2, 20]$	# de segmentos, Tamaño de alfabeto.
2	PAA	R	ns $\in [2, TSL/2]$	# de segmentos.
3	PCA	R	red $\in [0.2 , 0.9]$	Tamaño de reducción.
1	KNN-ED	C	k $\in [1, 15]$	# de vecinos.
2	KNN-DTW	C	k $\in [1, 15]$, r $\in [1, TSL \cdot 0.1]$	# de vecinos, Tamaño ventana.
3	KNN-MD	C	k $\in [1, 15]$	# de vecinos.
4	KNN-CD	C	k $\in [1, 15]$	# de vecinos.

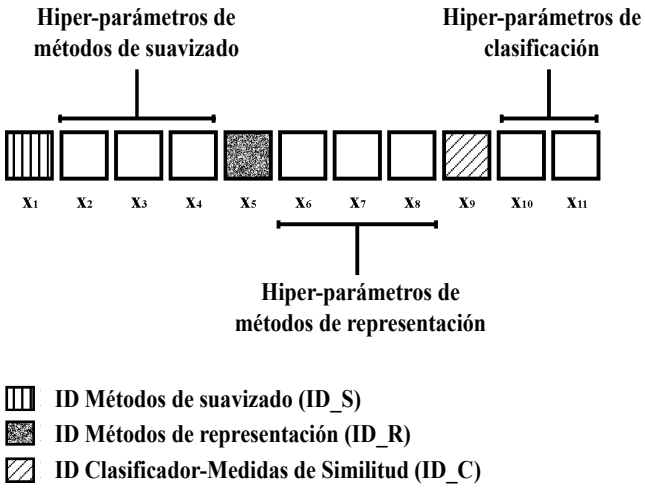


Figura 6.1: Codificación de una solución en μ-DEMS.

que pueden tomar los hiperparámetros de los métodos de suavizado (x_2 y x_3 pertenecen a ID_S = 1, mientras que x_4 pertenece a ID_S = 2). La variable x_5 codifica el identificador del método de representación seleccionado (ID_R) y puede tomar tres diferentes valores. Las variables x_6, x_7 y x_8 codifican los hiperparámetros de los métodos de representación, donde x_6 es una variable compartida por los métodos de representación ID_R = 1 y ID_R = 2, debido a que el hiperparámetro denominado como número de segmentos es ocupado tanto por el método SAX como por PAA, y está definido en el mismo rango. La variable x_9 representa el identificador de la combinación del clasificador k-NN con una medida de similitud diferente. Las variables x_{10} y x_{11} codifican los hiperparámetros restantes que corresponden a la tarea de clasificación.

El algoritmo estándar de DE codifica las soluciones en un esquema de representación de números reales. Por lo tanto, cada variable de un vector solución en μ-DEMS es inicializado con un valor aleatorio distribuido uniformemente dentro de un rango que establece los

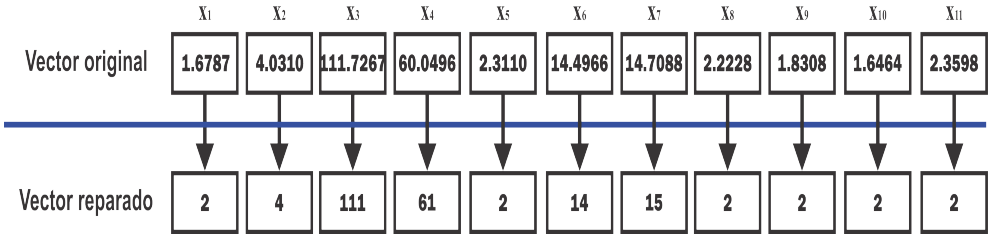


Figura 6.2: Decodificación de una solución en μ-DEMS.

límites de cada hiperparámetro (ver Tabla 6.1). En la Ecuación 6.4 se expresa la forma de inicializar cada variable del vector solución en μ-DEMS, donde L representa el límite inferior de la variable, U el límite superior y $rand_j$ representa un número aleatorio.

$$x_{i,0}^j = b_L^j + rand_j(0,1) \cdot (b_U^j - b_L^j) \tag{6.4}$$

6.2.2. DECODIFICACIÓN DE LA SOLUCIÓN

Para poder evaluar una solución dentro de μ-DEMS es necesario llevar a cabo un proceso de decodificación, que consiste en transformar un vector de valores reales a valores discretos como se muestra en la Figura 6.2.

La mayoría de las variables del vector son transformadas en el valor discreto más cercano usando una función de redondeo, donde a una variable real se le asigna el entero más cercano, si la primera cifra decimal es mayor o igual a 5 el redondeo se hace por exceso, es decir, el valor se redondea desde cero al entero de mayor magnitud, si no se toma exclusivamente la parte entera. Dentro del vector codificado como en la Figura 6.1, existen algunas variables que requieren obligatoriamente un valor impar. Para estos casos se hace uso de las funciones de *piso()* y *techo()* para localizar el valor impar más cercano.

6.2.3. FUNCIÓN DE APTITUD

La calidad de las soluciones generadas por μ-DEMS es evaluada mediante la tasa de error (ER) que se expresa en la Ecuación 6.5.

$$ER = \frac{a}{b} \tag{6.5}$$

donde a representa la porción de instancias de la base de datos temporal que fueron clasificados incorrectamente mientras que b es el número total de instancias en dicha base de datos. Con el objetivo de validar los modelos y evitar el sobre-ajuste (*overfitting*), se hace uso del método de validación cruzada de k iteraciones [69, 141], ver Ecuación 6.6.

$$CVER = \frac{1}{k}\sum_{i=1}^{k}(\frac{a}{b})_i \tag{6.6}$$

Decidir el valor de k para la validación cruzada puede resultar otro problema de decisión. Sin embargo, dentro de la literatura relacionada en FMS se sugiere utilizar un valor de 5 [45].

6.2.4. FUNCIÓN DE RESTRICCIÓN DEL LÍMITE

Durante la etapa de mutación dentro de μ-DEMS, así como en otros algoritmos de optimización, puede darse el caso de que algunas variables del vector solución se salgan de los límites establecidos para dichas variables. Por lo tanto, dentro de la literatura se sugiere el uso de algún método para el manejo de restricciones de límite. En este trabajo se usa el método de reflexión [145] que se encarga de mantener dentro de los límites a las vectores intermedios que violan las restricciones del límite, en la Ecuación 6.7 se resume el método de reflexión.

$$u_{i,g}^{j} = \begin{cases} 2 \cdot b_L^j - u_{i,g}^j & \text{si} u_{i,g}^j < b_L^j \\ 2 \cdot b_U^j - u_{i,g}^j & \text{si} u_{i,g}^j > b_U^j \end{cases} \tag{6.7}$$

6.2.5. VARIANTES

Existen diversas variantes de DE, con μ-DEMS se compara el comportamiento de cuatro variantes que funcionan bajo el procedimiento presentado en el Algoritmo 8. Los únicos componentes que varían son el tipo de cruza y el tipo de vector base utilizado en el cálculo del vector mutante (pasos 12-16 en el Algoritmo 8). Las variantes se nombran siguiendo la nomenclatura $DE/x/y/z$, donde DE son las siglas en inglés para referirse al algoritmo de evolución diferencial, x indica la forma en que se selecciona al vector base que interviene en el cálculo de valores de la mutación, y es el número de vectores de diferencia y z el tipo de operador de cruza [128].

A través del operador de cruza se crea un descendiente, el cual se componente de la mezcla del vector actual y el vector generado por la mutación. Usualmente en DE se utilizan dos opciones de cruza que son: la cruza binomial (bin) y la exponencial (exp). Para ambas opciones de cruza, el proceso de mezcla está controlado por una probabilidad de cruza, generalmente indicada por CR. La principal diferencia entre estas dos opciones de cruza es que para la cruza binomial cada variable del vector descendiente es tomado en cada momento de uno de los dos vectores padres en función de CR. Por otro lado, en la cruza exponencial cada variable del vector descendiente se toma del primer vector padre hasta que un número aleatorio supera el valor de CR, a partir de este punto, todas las variables del vector descendiente serán tomadas del segundo vector padre [100]. En la Tabla 6.2 se detallan las variantes evaluadas con μ-DEMS.

6.3. EXPERIMENTOS Y RESULTADOS

En esta sección se describen los experimentos llevados a cabo con el objetivo de evaluar el funcionamiento de μ-DEMS. El diseño experimental está enfocado en comparar el rendimiento de μ-DEMS usando cuatro diferentes variantes de evolución diferencial evaluadas en conjuntos de datos temporales.

6.3.1. CONFIGURACIÓN EXPERIMENTAL

Para esta etapa de experimentos se ha usado un sub-conjunto de bases de datos temporales que se derivan de un importante benchmark de bases de datos de series de tiempo [25], que han sido ampliamente utilizados en diversos estudios relacionados con minería de datos temporal [73, 123, 180]. En la Tabla 6.3 se presentan las características de las bases de datos usadas para los experimentos.

Tabla 6.2: Variantes de DE usadas en los experimentos, $rand_j \in [0,1]$ es un número real aleatorio, $j_{rand} \in [1,D]$ es un número entero aleatorio . $\vec{x}_{best,g}$ es el mejor vector de la generación g.

Nomenclatura	Variante
rand/1/bin	$u^j_{i,g} = \begin{cases} x^j_{r_3,g} + F \cdot (x^j_{r_1,g} - x^j_{r_2,g}) & \text{if } rand_j[0,1] < CR \text{ or } j = j_{rand} \\ x^j_{i,g} & \text{otherwise} \end{cases}$
rand/1/exp	$u^j_{i,g} = \begin{cases} x^j_{r_3,g} + F \cdot (x^j_{r_1,g} - x^j_{r_2,g}) & \text{while } rand_j[0,1] < CR \text{ or } j = j_{rand} \\ x^j_{i,g} & \text{otherwise} \end{cases}$
best/1/bin	$u^j_{i,g} = \begin{cases} x^j_{best,g} + F \cdot (x^j_{r_1,g} - x^j_{r_2,g}) & \text{if } rand_j[0,1] < CR \text{ or } j = j_{rand} \\ x^j_{i,g} & \text{otherwise} \end{cases}$
best/1/exp	$u^j_{i,g} = \begin{cases} x^j_{best,g} + F \cdot (x^j_{r_1,g} - x^j_{r_2,g}) & \text{while } rand_j[0,1] < CR \text{ or } j = j_{rand} \\ x^j_{i,g} & \text{otherwise} \end{cases}$

Tabla 6.3: Características de las bases de datos de series de tiempo utilizadas en experimentos con μ-DEMS.

No.	**Nombre**	**# Clases**	**# Instancias**	**Longitud de serie de tiempo**
1.	Beef	5	60	470
2.	Coffee	2	56	286
3.	ECG200	2	200	96
4.	OliveOil	4	60	570
5.	FaceFour	4	112	350
6.	GunPoint	2	200	150

Los parámetros utilizados por μ-DEMS, independientemente de la variante, se presentan en la Tabla 6.5, esta configuración se basa en los valores sugeridos por Viveros Jiménez et al. [179], quienes comparan diferentes algoritmos en su versión micro para optimización numérica. El único valor que fue ajustado fue la frecuencia de reemplazo (R), ya que Viveros Jiménez et al. [179] sugiere reemplazar cada 100 generaciones y en este trabajo se usa 10 con el objetivo de no perder diversidad.

Todos los experimentos se llevaron a cabo en equipos de cómputo con la configuración que se muestra en la Tabla 8.2.

Tabla 6.4: Configuración del equipo de cómputo utilizado en cada experimento.

Sistema Operativo	Windows 7 Pro
CPU	Intel® Xeon® E3-1200 v3
RAM	16 GB

Debido a lo costoso que resulta calcular la función de aptitud y siguiendo las sugerencias mencionadas por Escalante [45], cinco ejecuciones independientes fueron llevadas a cabo por cada variante de μ-DEMS. En cada ejecución el criterio de paro utilizado fue el de 1500 evaluaciones. Los resultados de los experimentos se dividen en tres partes: (1) la comparación estadística de los resultados numéricos, (2) la convergencia de las variantes y (3) el análisis de la mejor solución obtenida.

Tabla 6.5: Configuración de parámetros utilizada por μ-DEMS.

Parámetros	Valor
NP	6
CR	0.1
F	0.9
N	2
R	10

6.3.2. EXPERIMENTO 1: COMPARACIÓN ESTADÍSTICA DE RESULTADOS NUMÉRICOS

A partir de las ejecuciones realizadas por cada variante de μ-DEMS en cada base de datos, se obtienen los resultados numéricos que se presentan en la Tabla 6.6, donde por cada base de datos se muestra el mejor, la media, el peor y la desviación estándar de CVER obtenido por cada variante del algoritmo. La prueba no-paramétrica denominada como Kruskal-Wallis con un 95% de confianza fue aplicada para llevar a cabo la comparación estadística de los valores finales reportados por cada variante de μ-DEMS. Con base en la prueba estadística no se observó diferencia significativa entre las variantes propuestas. Sin embargo, se puede notar que los valores de CVER que fueron obtenidos por la variante *best/1/bin*, son los más bajos en la mayoría de las bases de datos utilizadas. En un contexto global, se puede decir que la variante *rand/1/bin* fue la variante menos competitiva.

Es importante mencionar que un CVER bajo representa que el modelo tiene un nivel alto de predicción mientras que un CVER alto significa una mala predicción.

Para mejorar la validación estadística como se sugiere en [39], se aplicó la prueba post hoc de Bonferroni, donde se hizo uso de los mejores valores y los valores de las medianas para comparar el rendimiento general de las cuatro variantes de μ-DEMS. La Figura 6.3 muestra los resultados de la prueba de Bonferroni, donde el eje x representa el intervalo de confianza de rangos medios (proporcionados por Bonferroni) y el eje y muestra el nombre de las cuatro variantes involucradas en la prueba. De acuerdo con la prueba post hoc, se confirma que no existe diferencia significativa entre las variantes, ya que gráficamente en la Figura 6.3 se observa un traslape entre los intervalos de confianza de cada variante. Tanto en el valor de las medianas como de los mejores valores se puede ver que la variante *best/1/bin* tiene los rangos medios más bajos en comparación con el resto de las variantes. Si se consideran los mejores valores, la variante rand/1/bin resultó tener el peor rendimiento general, mientras que por el lado de los valores de la mediana, la variante *best/1/exp* es la que tiene los peores valores.

A partir del análisis estadístico se puede concluir que el rendimiento de las cuatro variantes de μ-DEMS es muy similar, sin embargo la variante que destaca por encontrar un CVER menor es la variante *best/1/bin*, en la que el mejor vector es utilizado como vector base para el operador de cruza.

6.3.3. EXPERIMENTO 2: ANÁLISIS DE CONVERGENCIA

Complemento al análisis estadístico, se presenta el análisis de convergencia de cada una de las variantes de μ-DEMS para cada base de datos utilizada. Para graficar la convergencia de cada variante se tomó la ejecución localizada en la mediana de las cinco ejecuciones independientes hechas por cada variante en cada base de datos. En las Figuras 6.4- 6.6 se

Tabla 6.6: Resultados estadísticos de CVER (B: mejor, M: media, W: peor, SD: Desviación estándar, p: p-valor) obtenidos por cada variante de μ-DEMS. Los valores en **negritas** representan los mejores valores encontrados.

Base de datos	Stad.	rand/1/bin	rand/1/exp	best/1/bin	best/1/exp
Beef	**B**	0.3667	0.3667	**0.3333**	0.3667
	M	0.3800	0.3900	**0.3700**	0.3833
	W	0.4000	0.4167	0.4000	0.4000
	SD	**0.0139**	0.0224	0.0298	0.0167
	p	0.6751			
Coffee	**B**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
	M	**0.0000**	0.0100	0.0067	0.0100
	W	0.0000	0.0333	0.0167	0.0167
	SD	**0.0000**	0.0149	0.0091	0.0091
	p	0.2945			
ECG200	**B**	0.0650	**0.0600**	**0.0600**	**0.0600**
	M	0.0670	0.0630	**0.0610**	0.0650
	W	0.0700	0.0650	0.0650	0.0700
	SD	0.0027	0.0027	**0.0022**	0.0050
	p	0.0637			
OliveOil	**B**	0.0667	**0.0500**	0.0667	0.0833
	M	0.0670	0.0630	**0.0610**	0.0650
	W	0.0700	0.0650	0.0650	0.0700
	SD	0.0027	0.0027	**0.0022**	0.0050
	p	0.4670			
FaceFour	**B**	0.0265	0.0265	**0.0261**	**0.0261**
	M	0.0353	0.0338	**0.0261**	0.0265
	W	0.0704	0.0447	0.0261	0.0269
	SD	0.0196	0.0076	**0.0000**	0.0003
	p	0.1021			
GunPoint	**B**	0.0050	0.0100	**0.0000**	0.0050
	M	0.0100	0.0100	**0.0030**	0.0100
	W	0.0150	0.0100	0.0050	0.0150
	SD	0.0035	**0.0000**	0.0027	0.0071
	p	0.0712			

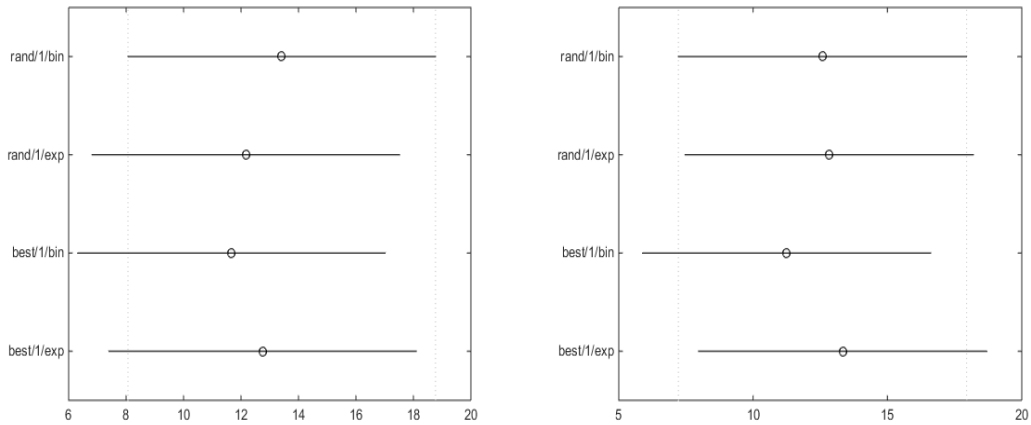


Figura 6.3: Prueba post-hoc de Bonferroni utilizando (a) los mejores valores y (b) los valores de las medianas del conjunto de bases de datos.

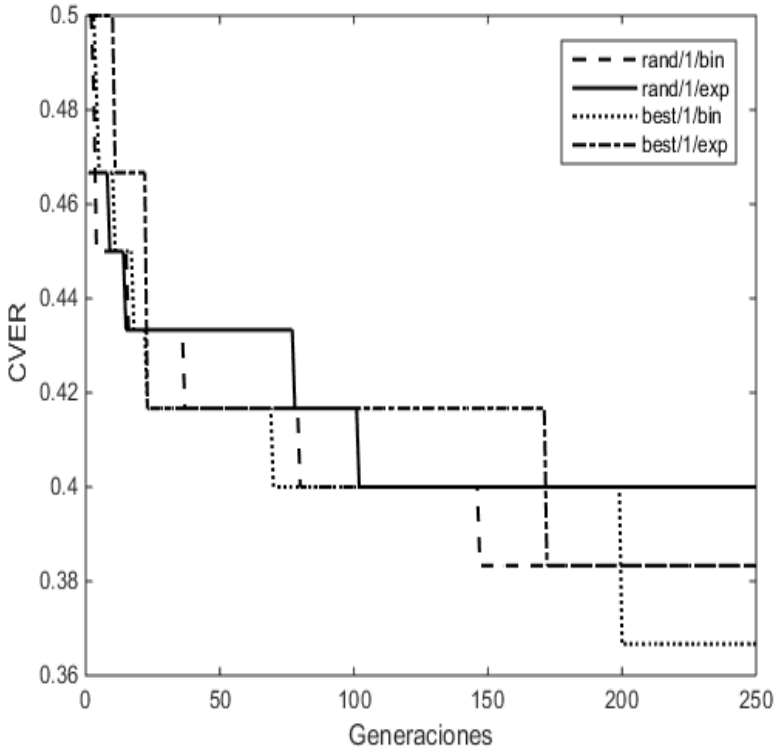
muestran las gráficas de convergencia de las cuatro variantes por cada base de datos.

A partir de las gráficas de convergencia, es interesante ver como la variante *best/1/bin* es capaz de alcanzar resultados más competitivos evitando óptimos locales en las bases de datos Beef, ECG200 y GunPoint. En las bases de datos OliveOil y FaceFour es posible ver cómo la variante *best/1/bin* también logra alcanzar un resultado competitivo además de lograr converger más rápido, en comparación con *rand/1/exp* y *best/1/exp*. En el caso de la base de datos Coffee, la variante rand/1/bin fue capaz de converger más rápido hacia un mejor resultado. Es importante mencionar que la variante rand/1/exp presenta una notable mejora numérica en la función de evaluación durante las primeras generaciones en las bases de datos ECG200, OliveOil, FaceFour y GunPoint. Sin embargo, en el transcurso de las generaciones queda atrapada en soluciones óptimas locales.

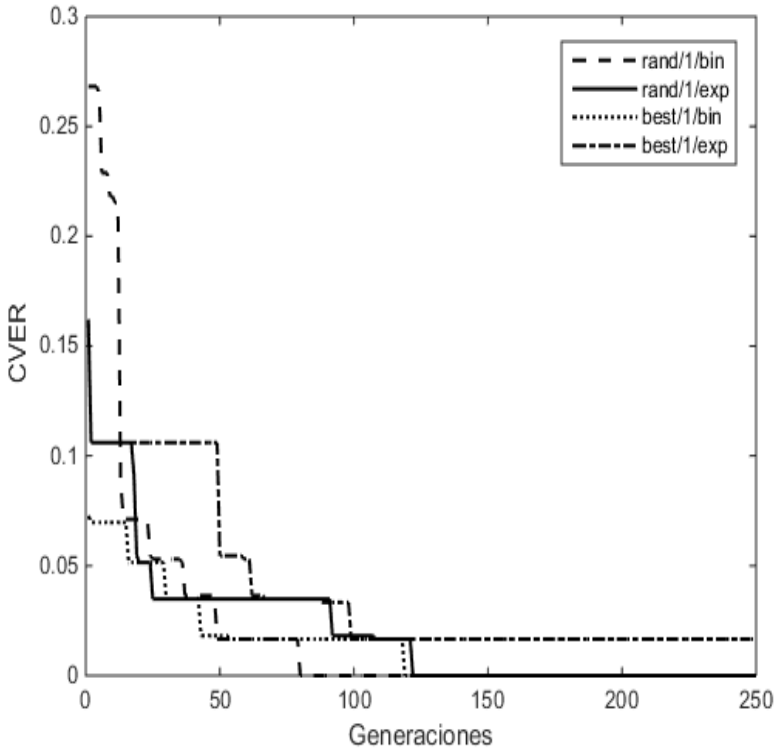
Para esta etapa experimental, donde se consideró el sub-conjunto de métodos presentados en la Tabla 6.1 como opciones para la formación de modelos completos, se sugiere que μ-DEMS con la variante *best/1/bin* representa la mejor opción de herramienta de búsqueda que tiene la capacidad de evitar óptimos locales. Por otro lado, si se desea obtener una solución rápida en un menor número de evaluaciones, se sugiere el uso de μ-DEMS con la variante *rand/1/exp* con la alta posibilidad de converger hacia óptimos locales.

6.3.4. EXPERIMENTO 3. ANÁLISIS DE LA SOLUCIÓN FINAL

En esta parte de los experimentos, se presenta el análisis de las soluciones finales sugeridas por cada una de las variantes de μ-DEMS para cada una de las bases de datos. La Tabla 6.7 resume los mejores modelos encontrados por cada variante, la primera columna muestra el nombre de la base de datos involucrada, la segunda columna muestra el nombre de la variante utilizada, en la tercera columna se detalla el modelo final sugerido, en la cuarta columna se muestra el valor de CVER obtenido para dicho modelo y en la quinta columna se muestra el tiempo promedio, en minutos, de la ejecución de una corrida de cada variante.

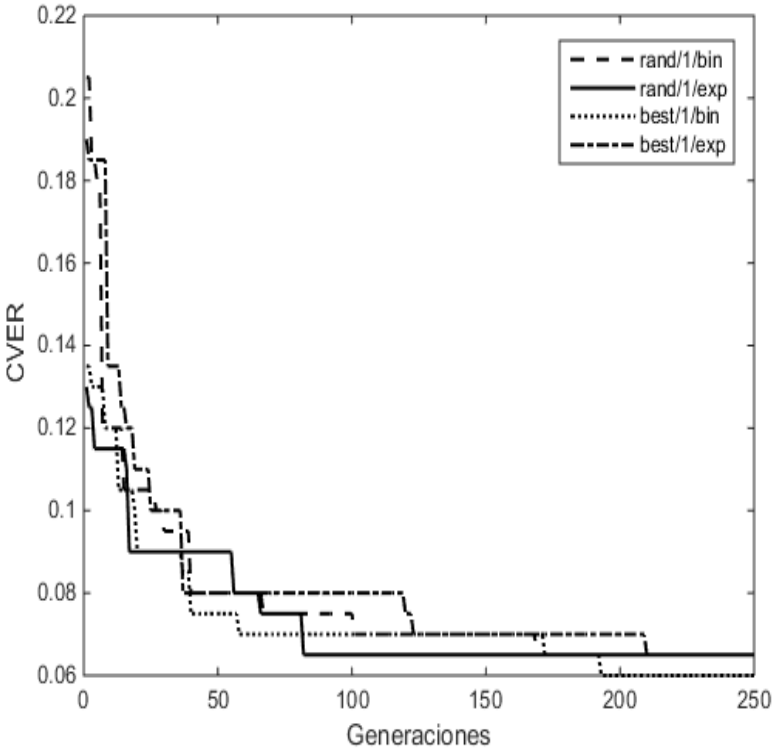


(1) Beef

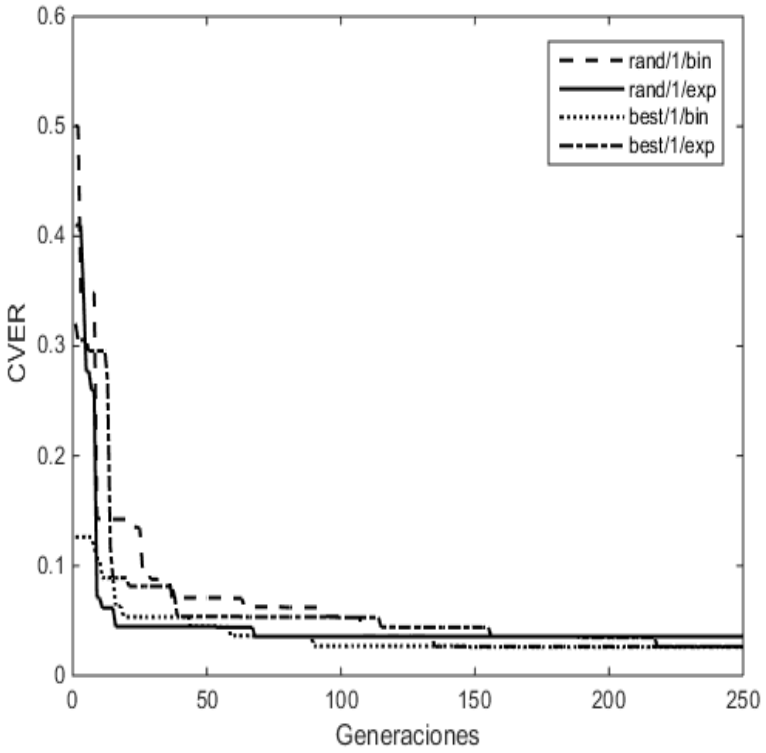


(2) Coffee

Figura 6.4: Gráficas de convergencia para las bases de datos Beef y Coffee.

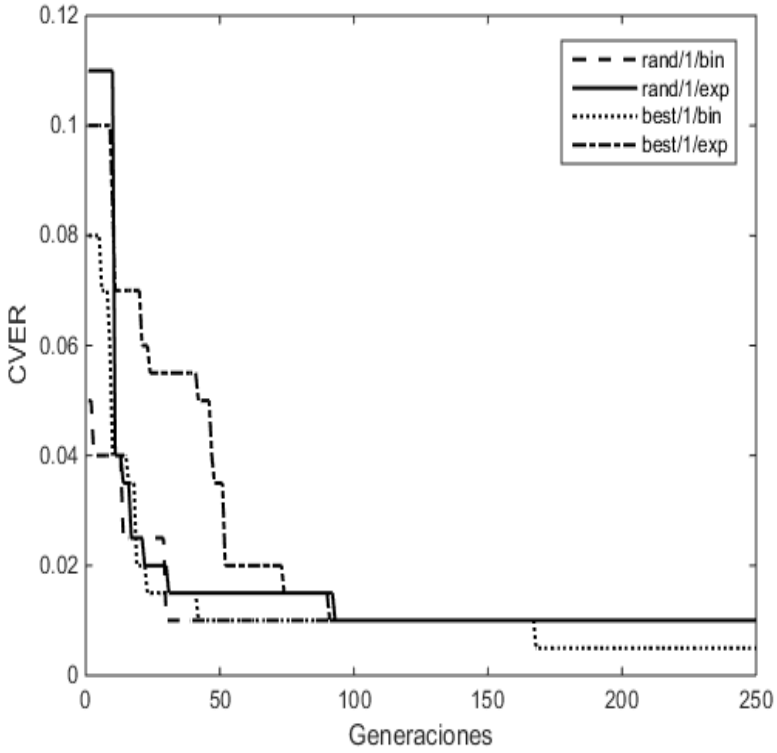


(3) ECG200

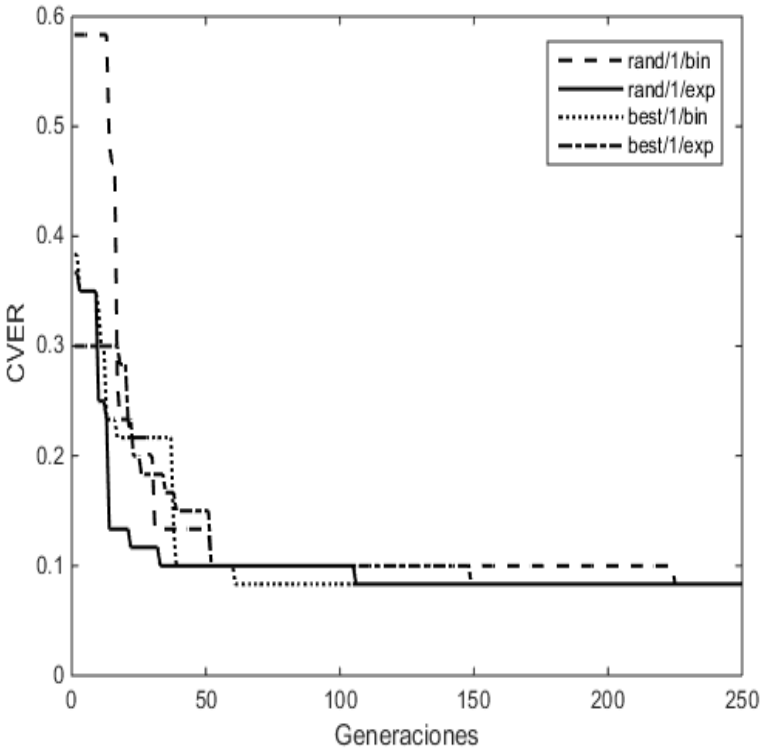


(4) Face Four

Figura 6.5: Gráficas de convergencia para las bases de datos ECG200 y FaceFour.



(5) Gun Point



(6) OliveOil

Figura 6.6: Gráficas de convergencia para las bases de datos Gun Point y OlieOil.

A pesar de la variedad de soluciones sugeridas, se pueden observar algunas similitudes interesantes. En relación al proceso de suavizado de las series de tiempo, el método SG, que corresponde al filtro de Savitzky-Golay, fue el algoritmo más seleccionado. Se destaca que para el caso del hiperparámetro relacionado con el tamaño de la ventana, se sugiere utilizar un tamaño que oscila entre un 5 al 20 % respecto de la longitud de la serie de tiempo.

Respecto a la representación de series de tiempo, los métodos PCA y SAX fueron los métodos de representación más populares. En el caso del método PCA se puede considerar que sin importar las características intrínsecas de las series de tiempo, el hiperparámetro encargado del porcentaje de la reducción de dimensionalidad de la serie se ajusta a un 60 %, es decir, que es la capacidad promedio de reducción en la mayoría de las bases de datos donde se utiliza este método. El método SAX fue sugerido principalmente para dos bases de datos en particular, Coffee y FaceFour que se caracterizan por tener series de tiempo de longitud intermedia. En relación al proceso de clasificación, el clasificador k-NN con distancia euclidiana fue la combinación mayormente preferida, donde se observa que el valor del hiperparámetro k, usualmente es sugerido en un rango de 1 a 3 vecinos.

Hablando de los tiempos de ejecución, se observa que varían considerablemente debido a diferentes aspectos como son: las características propias de las bases de datos (cantidad de clases, cantidad de instancias de entrenamiento, longitud de las series de tiempo), la variedad de métodos evaluados y sus hiperparámetros relacionados. Considerando los tiempos promedios que se presentan en la quinta columna de la Tabla 6.7, se puede notar que la ejecución con un menor costo (119 minutos) fue obtenida por la variante *rand/1/exp* en el contexto de la base de datos Coffee, mientras que la ejecución más cara (6870 minutos) fue realizada por la variante *best/1/bin* con la base de datos GunPoint.

Finalmente, a partir de los mejores modelos encontrados para cada una de las variantes en cada base de datos, en las Figuras 6.7 y 6.8 se muestran dos ejemplos gráficos del modelo sugerido para las bases de datos FaceFour y Beef. En estos ejemplos se muestra el comportamiento promedio de los datos originales, posteriormente se muestra el resultado al ser aplicado el método de suavizado y finalmente se observa el resultado de los datos reducidos.

El resto de las representaciones gráficas de las soluciones pueden ser consultadas en el Anexo 10.

6.4. CONCLUSIONES

En este capítulo se presentó la propuesta del algoritmo denominado como μ-DEMS, a partir del cual se realizó una comparación empírica de cuatro variantes de evolución diferencial: *rand/1/bin*, *rand/1/exp*, *best/1/bin* y *best/1/exp* para resolver una instancia del problema de selección del modelo completo en bases de datos temporales. La comparación se centró en analizar el efecto de la cruza y el tipo de vector base cuando se resuelve el problema de FMS en bases de datos temporales. Cada variante de μ-DEMS fue evaluada en seis bases de datos temporales, donde el análisis experimental se dividió en el análisis estadístico de los resultados numéricos, el comportamiento de convergencia y el análisis del tipo de solución final encontrada.

La evaluación general sugiere que μ-DEMS representa una opción viable para encontrar modelos completos. Por otro lado, se encontró que el uso del mejor vector de la población actual como vector base junto con una cruza binaria es una buena opción para alcanzar me-

Tabla 6.7: Modelos sugeridos por cada variante de μ-DEMS.

Base de datos	Variante	Modelo	CVER	Tiempo (min.)
Beef	**rand/1/bin**	Smoothing: MA{span(105)}, Representación: PCA{red(0.6)}, Clasificación: KNN-MD{k(6)}	0.3667	752.55
	rand/1/exp	Suavizado: SG{k(1),f(115)}, Representación: PCA{red(0.4)}, Clasificación: KNN-MD{k(7)}	0.3667	1046.23
	best/1/bin	Suavizado: SG{k(2),f(153)}, Representación: PCA{red(0.8)}, Clasificación: KNN-MD{k(8)}	0.3333	752.97
	best/1/exp	Suavizado: MA{span(33)}, Representación: PAA{ns(190)}, Clasificación: KNN-MD{k(9)}	0.3667	1089.15
Coffee	**rand/1/bin**	Suavizado: SG{k(2),f(11)}, Representación: PAA{ns(103)}, Clasificación: KNN-CD{k(1)}	0.0000	174.67
	rand/1/exp	Suavizado: SG{k(5),f(33)}, Representación: SAX{w(86),a(18)}, Clasificación: KNN-ED1{k(13)}	0.0000	119.04
	best/1/bin	Suavizado: SG{k(3),f(25)}, Representación: SAX{w(93),a(18)}, Clasificación: KNN-ED1{k(4)}	0.0000	190.78
	best/1/exp	Suavizado: SG{k(2),f(23)}, Representación: SAX{w(139),a(20)}, Clasificación: KNN-ED1{k(10)}	0.0000	107.20
ECG200	**rand/1/bin**	Suavizado: SG{k(3),f(5)}, Representación: PCA{red(0.5)}, Clasificación: KNN-MD{k(2)}	0.0650	260.35
	rand/1/exp	Suavizado: SG{k(5),f(5)}, Representación: PCA{red(0.4)}, Clasificación: KNN-MD{k(2)}	0.0600	281.97
	best/1/bin	Suavizado: SG{k(2),f(3)}, Representación: PCA{red(0.7)}, Clasificación: KNN-DTW1{k(1),r(8)}	0.0600	306.41
	best/1/exp	Suavizado: SG{k(4),f(9)}, Representación: PCA{red(0.7)}, Clasificación: KNN-MD{k(3)}	0.0600	169.75
OliveOil	**rand/1/bin**	Suavizado: MA{span(15)}, Representación: PCA{red(0.8)}, Clasificación: KNN-MD{k(1)}	0.0667	637.96
	rand/1/exp	Suavizado: SG{k(4),f(39)}, Representación: PCA{red(0.3)}, Clasificación: KNN-MD{k(1)}	0.0500	700.18
	best/1/bin	Suavizado: MA{span(15)}, Representación: PCA{red(0.9)}, Clasificación: KNN-MD{k(2)}	0.0667	739.35
	best/1/exp	Suavizado: SG{k(2),f(19)}, Representación: PCA{red(0.8)}, Clasificación: KNN-MD{k(3)}	0.0833	813.28
FaceFour	**rand/1/bin**	Suavizado: SG{k(4),f(23)}, Representación: PCA{red(0.6)}, Clasificación: KNN-MD{k(1)}	0.0265	1136.19
	rand/1/exp	Suavizado: SG{k(3),f(9)}, Representación: SAX{w(93),a(19)}, Clasificación: KNN-ED1{k(1)}	0.0265	551.81
	best/1/bin	Suavizado: SG{k(5),f(23)}, Representación: SAX{w(146),a(9)}, Clasificación: KNN-ED1{k(1)}	0.0261	1672.13
	best/1/exp	Suavizado: SG{k(3),f(13)}, Representación: PCA{red(0.4)}, Clasificación: KNN-MD{k(1)}	0.0261	882.48
GunPoint	**rand/1/bin**	Suavizado: MA{span(11)}, Representación: PAA{ns(48)}, Clasificación: KNN-DTW2{k(1),r(3)}	0.0050	1820.59
	rand/1/exp	Suavizado: SG{k(3),f(19)}, Representación: PAA{ns(74)}, Clasificación: KNN-DTW2{k(1),r(14)}	0.0000	3518.24
	best/1/bin	Suavizado: SG{k(2),f(15)}, Representación: PAA{ns(74)}, Clasificación: KNN-DTW2{k(1),r(15)}	0.0000	6870.46
	best/1/exp	Suavizado: SG{k(4),f(25)}, Representación: PAA{ns(75)}, Clasificación: KNN-DTW2{k(3),r(14)}	0.0050	4105.02

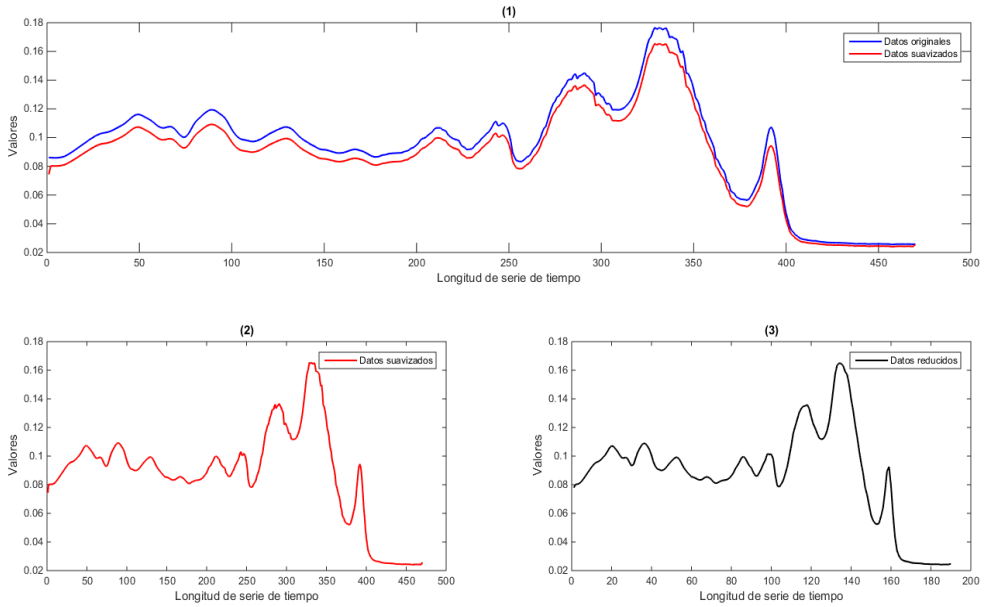


Figura 6.7: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

jores resultados numéricos. Por el contrario, si se tiene una base de datos que requiera un mayor tiempo computacional para la búsqueda y evaluación de sus modelos, se recomienda que en lugar de usar el mejor vector base puede elegirse uno al azar con una cruza binaria para obtener resultados competitivos en un menor número de evaluaciones.

A partir del análisis del comportamiento de convergencia, se observó que en un rango de 150 a 200 generaciones era posible lograr resultados competitivos a través del uso de las variantes *rand*, es decir, aquellas variantes donde el vector base utiliza un vector de la población actual seleccionado al azar. Además, si el número de generaciones excedía a 200, se notó que la variante *best/1/bin* era capaz de lograr mejores resultados finales.

Finalmente, en el análisis de las soluciones finales encontrados se muestra que los métodos SG, SAX, PCA y KNN con distancia euclidiana fueron los métodos mayormente sugeridos por las variantes de μ-DEMS para llevar a cabo las tareas de suavizado, representación de series de tiempo y clasificación respectivamente. Algunos hallazgos importantes fueron encontrados a partir del análisis de los modelos finales, por ejemplo:

- Para el uso del método de suavizado SG, el valor del hiperparámetro correspondiente a la ventana puede tomar valores en un rango del 5 al 20 % respecto a la longitud de la serie de tiempo que se esté procesando.

- El método PCA, tiene una capacidad promedio de reducción de dimensionalidad de una serie de tiempo del 60 %.

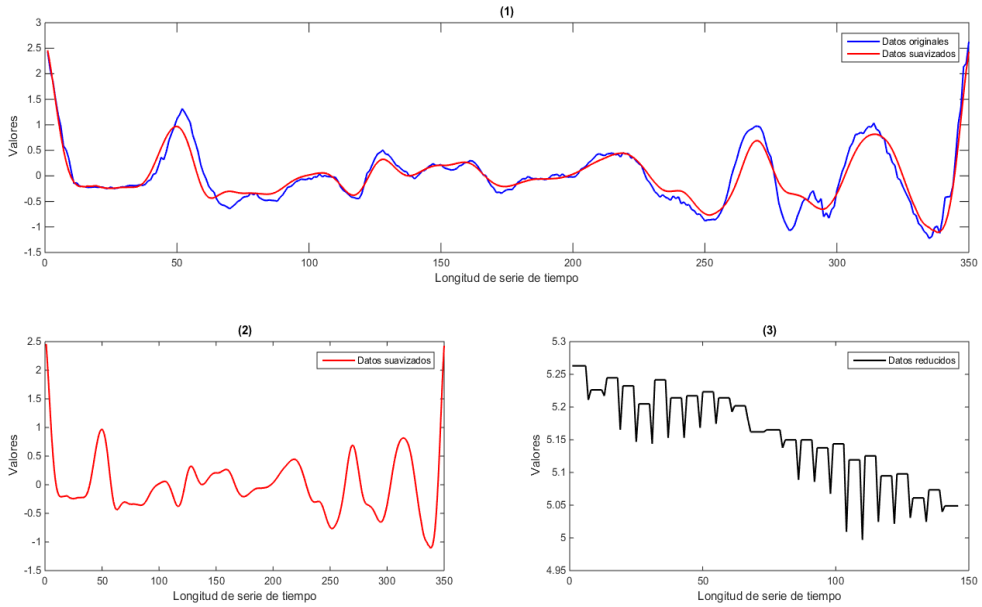


Figura 6.8: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

- El valor más común para el hiperparámetro k del clasificador KNN oscila entre 1 a 3.

En el siguiente capítulo se presenta una extensión de la instancia del problema de FMS, donde se incorporan nuevos métodos de suavizado, se incorpora la tarea de reducción de instancias y se agregan nuevos métodos de clasificación.

7

FMS EN BASES DE DATOS TEMPORALES: COMPARACIÓN DE METAHEURÍSTICAS BASADA EN POBLACIÓN Y BASADA EN UN SOLO PUNTO

> La ciencia es un cementerio de ideas muertas, aunque de ellas puede salir la vida.
>
> —Miguel de Unamuno

7.1. ANTECEDENTES

En este capítulo se presenta una extensión del análisis descrito en el Capítulo 6. Se incorporan nuevos métodos de suavizado, se agregan dos métodos para la tarea de reducción de datos a nivel de tuplas, así como la incorporación de nuevos métodos de clasificación. A partir de esta nueva instancia del problema de FMS, se comparan dos enfoques de búsqueda, el primero basado en metaheurísticas poblacionales (PM, *Population-based Metaheuristic*) y el segundo en metaheurísticas de un solo punto de búsqueda (SM, *Single-point-based search Metaheuristic*).

El enfoque PM funciona bajo el concepto del algoritmo μ-DEMS presentado en el capítulo anterior, en el que cuatro diferentes variantes son evaluadas. En este capítulo, dichas variantes son denominadas como P-DEMS1, P-DEMS2, P-DEMS3 y P-DEMS4, las cuales usan una codificación con números reales de las soluciones. Respecto al enfoque SM, se presentan dos versiones basadas en una simple búsqueda local (LS, *Local Search*). La primera versión usa una codificación binaria y es denominada como S-LSMS1, mientras que la segunda versión utiliza una codificación mixta y es denominada como S-LSMS2.

7.2. DESCRIPCIÓN DE LA PROPUESTA

La metodología que se adoptó para la evaluación de los enfoques PM y SM se presenta en la Figura 7.1. Los datos de entrada se dividen en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento se utiliza en el proceso de búsqueda para evaluar cada modelo candidato, donde cada modelo está compuesto por la combinación de un método de suavizado, de representación de series de tiempo, de reducción de numerosidad y de un clasificador con los hiper-parámetros relacionados a cada método. Al final del proceso de búsqueda, se obtiene el mejor modelo que se evalúa en el conjunto de prueba. A continuación se describen las codificaciones utilizadas por las soluciones, la función de aptitud y los enfoques de búsqueda.

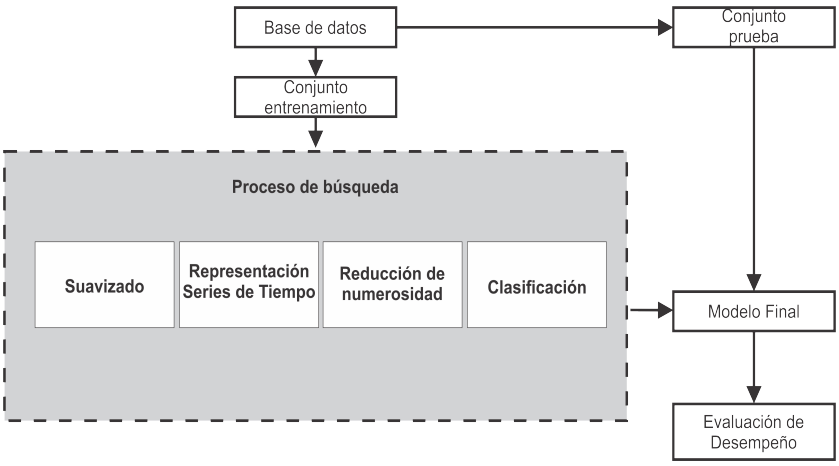


Figura 7.1: Metodología general del enfoque de FMS para bases de datos temporales.

7.2.1. CODIFICACIÓN DE LA SOLUCIÓN

CODIFICACIÓN REAL

En el enfoque PM, cada solución potencial es codificada como un vector de números reales que es formado como se muestra en la Ecuación 7.1. Donde j representa cada posición dentro de un vector en particular, $x_{j,s} \in [1,4]$, $x_{j,r} \in [1,3]$, $x_{j,n} \in [1,2]$ y $x_{j,c} \in [1,6]$ representan el identificador (ID) del método seleccionado para las tareas de suavizado, representación de series de tiempo, reducción de numerosidad y clasificación respectivamente. $h_{j,1,\ldots,ns}$, $h_{j,1,\ldots,nr}$, $h_{j,1,\ldots,nn}$ y $h_{j,1,\ldots,nc}$ codifica cada conjunto de hiper-parámetros correspondientes a cada método seleccionado, donde ns, nr, nn y nc representan el número total de hiper-parámetros que poseen diferentes rangos, de los límites (límite inferior, lb y límite superior, ub) por cada subconjunto de métodos de suavizado, representación, reducción de numerosidad y clasificación. Por lo tanto, la dimensión D de cada vector solución es calculada mediante la Ecuación 7.2.

En este capítulo, cada solución candidata generada por el enfoque PM posee 24 dimensiones (variables) que toman valores de acuerdo a los límites de cada hiper-parámetro presentados en la Tabla 7.1. La Figura 7.2 muestra una representación gráfica del vector solución.

$$\mathbf{x}_i = [x_{j,s}, h_{j,1,\ldots,ns}, x_{j,r}, h_{j,1,\ldots,nr}, x_{j,n}, h_{j,1,\ldots,nn}, x_{j,c}, h_{j,1,\ldots,nc}] \tag{7.1}$$

Tabla 7.1: Descripción de los métodos considerados y los hiperparámetros utilizados. TLS significa la longitud de la serie de tiempo.

ID	Método	Tipo	Hiperparámetros $\in \mathbb{R}$	Descripción
1	SG - Filtro Savitzky-Golay	S	k [1,5] f [1,TSL]	Orden polinomial Tamaño ventana
2	MA - Promedio Móvil	S	span [2,TSL]	Tamaño de ventana
3	LRw - Regresión Local (lowess)	S	%span [0.1,1]	Porcentaje de tamaño ventana
4	LRe - Regresión Local (loess)	S	%span [0.1,1]	Porcentaje de tamaño ventana
1	SAX - Symbolic Aggregate Approximation	R	nseg [2,TSL/2] a [2,20]	Número de segmentos Alfabeto
2	PAA - Piecewise Aggregate Approximation	R	nseg [2,TSL/2]	Número de segmentos
3	PCA - Análisis de Componentes Principales	R	%red [0.20,0.90]	Porcentaje de reducción
1	INSIGTH	I	%ins [0.5,1]	Porcentaje de instancias
2	MC1 - Monte Carlo 1	I	%ins [0.5,1] nitera [1,500]	Porcentaje de instancias Número de iteraciones
1	KNN-ED - K-vecinos más cercanos - Distancia Euclidiana	C	k [1,7]	Número de vecinos
2	KNN-LBDTW - K-vecinos más cercanos - Lower Bounding Dynamic Time Warping	C	k [1,7] r [0.0,0.05]	Número de vecinos Porcentaje de warping
3	NB - Naive Bayes	C	tk [1,4]	Tipo de kernel, 1= normal, 2 = box, 3 = epanechnikov y 4 = triangle
4	TD - Árbol de decisión	C	tsc [1,3] maxCat [2,20]	Criterio de división, 1 = gdi, 2 = twoing y 3 = deviance Máximo de niveles
5	AB - AdaBoost	C	nTD [2,500]	Número de aprendices
6	SVM - Support Vector Machine	C	-t [0,3] -d [1,50] -g [1.0,100.0] -r [0.0,100.00] -c [1.0,1000.0]	Tipo de kernel, 0 = lineal, 1 = polinomial, 2 = radial y 3 = sigmoidal Grado de la función kernel Gamma en función kernel Coef0 en función kernel Cost

$$D = s + ns + r + nr + n + nn + c + nc \tag{7.2}$$

CODIFICACIÓN BINARIA

La codificación binaria es el esquema tradicional para la representación de soluciones usado por los algoritmos genéticos. Algunas de las razones por las que se ha utilizado la representación binaria se remonta al trabajo de Holland [70], que comparó dos representaciones diferentes (binaria y decimal) que tuvieran aproximadamente la misma capacidad de acarreo de información.

Holland argumentó que la representación binaria da pie a un grado más elevado de paralelismo implícito porque permite mayor número de esquemas que la representación decimal. Un esquema se puede definir como una plantilla que describe un sub-conjunto de cadenas que comparten ciertas similitudes en algunas posiciones a lo largo de su longitud [65].

Por lo tanto, el hecho de contar con más esquemas favorece la diversidad e incrementa la probabilidad de formar mejores soluciones que vayan mejorando en cada generación.

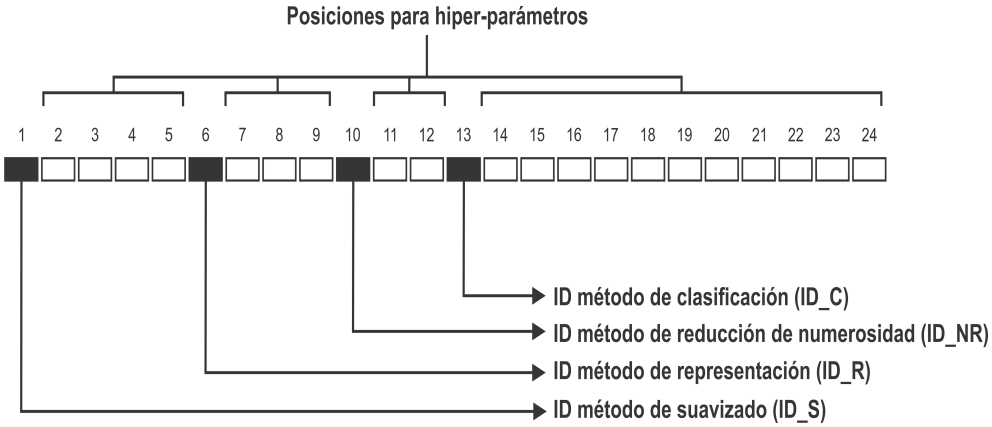


Figura 7.2: Representación gráfica de un vector solución usado en la codificación real y mixta.

La representación binaria es utilizada para evaluar el enfoque SM, donde cada vector solución $\mathbf{x}_i$ está formado por cadenas binarias. Cada cadena binaria posee una longitud en particular l_j, calculada a partir de la Ecuación 7.3, donde se consideran los límites de cada variable (hiper-parámetro) así como un valor de precisión p. Para esta instancia del problema, el valor de p fue establecido a 0 cuando se requería codificar valores discretos, mientras que para codificar valores reales se usó un valor de $p = 4$. La longitud total del vector svl, es el resultado de la concatenación de cada cadena que corresponde a una determinada variable, la Ecuación 7.4 muestra la forma de calcular la longitud total de un vector solución. En la Figura 7.3 se muestra un ejemplo de una solución codificada en representación binaria.

$$l_j = \text{int}[\log_2[(ub_j - lb_j) \cdot 10^p] + 0{,}9] \tag{7.3}$$

$$svl = \sum_{j=1}^{D} l_j \tag{7.4}$$

Para el cálculo de la función de aptitud de cada solución, se requiere de un proceso de decodificación. Dicho proceso es aplicado por cada cadena que forma el vector completo a través de la Ecuación 7.5.

$$x_{ij} = lb_j + \frac{x_{int}(ub_j - lb_j)}{2^{l_j} - 1} \tag{7.5}$$

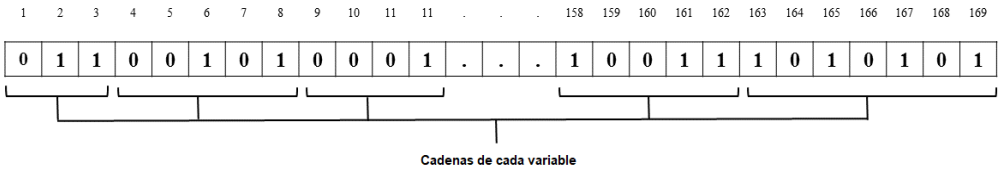


Figura 7.3: Representación gráfica de la codificación binaria.

CODIFICACIÓN MIXTA

La codificación mixta consiste en un vector D-dimensional, donde D se calcula mediante la Ecuación 7.2. Cada variable del vector se genera dentro de sus límites establecidos, ver Tabla 7.1. En esta codificación, se permiten tanto valores reales como discretos.

7.2.2. FUNCIÓN DE APTITUD

El problema de FMS es tratado como un problema de optimización global (mono-objetivo). Tanto en el enfoque PM como en el SM, la tasa de error (ver Ecuación 6.5) de un modelo es considerada como la función de aptitud f_x. Al igual que en el capítulo anterior, se adopta el método de validación cruzada (ver Ecuación 6.6) con el objetivo de hacer frente al sobreajuste de los modelos. Por lo tanto, la función de aptitud puede expresarse como en la Ecuación 7.6.

$$f(x) = CVER \tag{7.6}$$

7.2.3. METAHEURÍSTICAS PARA EL PROCESO DE BÚSQUEDA

En esta sección se describen los dos enfoques que se van a comparar, que son el basado en una metaheurística poblacional y el basado en una metaheurística de un solo punto de búsqueda.

ENFOQUE PM

El enfoque poblacional está basado en la propuesta presentada en el Capítulo 6, donde se propone el uso de la versión micro del algoritmo de evolución diferencial, ver Algoritmo 8. Se evalúan las cuatro variantes que son: *rand/1/bin*, *rand/1/exp*, *best/1/bin* y *best/1/exp*.

ENFOQUE SM

El enfoque SM esta basado en la idea básica de la búsqueda local, que es considerada como el método metaheurístico más antiguo y simple [169]. En cada iteración de LS, una solución s es reemplazada por una solución vecina siempre que se mejore la función de aptitud; de lo contrario, la solución original es conservada. Alguno de los criterios de paro es detener la búsqueda cuando todas las soluciones vecinas resultan ser peores que la solución actual. En este trabajo se estableció como criterio de paro un número máximo de iteraciones.

Algoritmo 9 LS

Entrada: N_k (Tamaño vecindario), $iter$ (Máximo número de iteraciones)
1: Set $i = 0$
2: s_0; */* Generar una solución inicial */*
3: **while** $i < iter$ **do**
4: $s = s_0$
5: Generar $(N(s))$; */* Generar vecinos candidatos/*
6: **if** existe un mejor vecino **then**
7: $s = s\prime$; */* Seleccionar un mejor vecino $s\prime \in N(s)$ */*
8: **end if**
9: **end while**
10: $i = i + 1$
Salida: Solución final encontrada(óptimo local)

Dos aspectos importantes que definen a la LS es la forma en que se generan las soluciones vecinas (estocástica o determinista) y la estrategia para seleccionar la solución vecina. En la

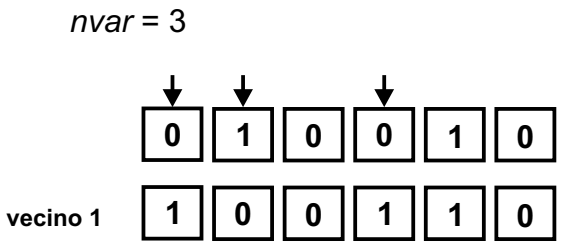


Figura 7.4: Ejemplo de vecino generado por el operador de LS en codificación binaria.

literatura, se han propuesto diversas estrategias que pueden ser aplicadas para la selección de la solución vecina, a continuación se describen brevemente algunas de estas estrategias [169].

- **Estrategia de la mejora más competitiva**. En esta estrategia, se selecciona el mejor vecino (es decir, el mejor vecino que mejora la función de aptitud). El vecindario es evaluado de forma determinista. Por lo tanto, la exploración se lleva a cabo de forma exhaustiva y se tratan todos los posibles movimientos de una solución con el objetivo de seleccionar la mejor solución vecina.
- **Estrategia de la primera mejora**. En esta estrategia, se elige el primer vecino que sea mejor que la solución actual. Por lo tanto, si se encuentra un vecino que mejora es seleccionado de forma inmediata. En el mejor de los casos, esta estrategia implica una evaluación parcial del vecindario.
- **Estrategia aleatoria**. En esta estrategia, se aplica una selección aleatoria a los vecinos que logre mejorar la solución actual.

En esta sección se propone un operador para la generación estocástica de soluciones vecinas, considerando dos enfoques de codificación (binaria y mixta). Para la selección de la solución vecina que pasa a la siguiente iteración, se propone el uso de la estrategia de la mejora más competitiva. En el paso cinco del Algoritmo 9 se señala el llamado a dicho operador. Se proponen dos versiones de LS, basadas en el Algoritmo 9, donde el operador del vecindario varía ligeramente dependiendo del tipo de codificación utilizada para representar a una solución.

Si se utiliza la codificación binaria, se seleccionan $nvar$ posiciones aleatoriamente, donde $nvar \in [1, svl]$. Las posiciones seleccionadas son cambiadas mediante una función de intercambio, es decir, 0 por 1 o viceversa. Cuando todas las $nvar$ posiciones han sido modificadas, se puede decir que se tiene una nueva solución vecina, ver ejemplo en la Figura 7.4.

Por otro lado, se plantea una versión donde la solución actual utiliza una representación mixta. Para dicha versión también se seleccionan de manera aleatoria $nvar$ posiciones de la solución actual. Sin embargo, es esta versión cada posición es cambiada por un nuevo valor que se encuentre en los límites de las variables correspondientes, ver ejemplo en la Figura 7.5.

7.3. EXPERIMENTOS Y RESULTADOS

En esta sección se describen los experimentos llevados a cabo en los que se compara el comportamiento de las metaheurísticas anteriormente descritas (basadas en la población y

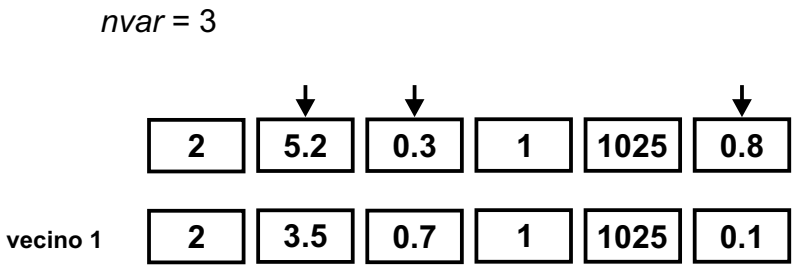


Figura 7.5: Ejemplos de vecino generado por el operador de LS en codificación mixta.

basadas en un único punto), para resolver el problema de FMS en bases de datos temporales. Se usaron diez bases de datos de series de tiempo, las cuales se describen en la Tabla 7.2.

Las características de las bases de datos que se presentan son: el nombre de la base de datos (columna 2), el número de clases (columna 3), el número de instancias que posee el conjunto de entrenamiento (columna 4), el número de instancias del conjunto de prueba (columna 5), la longitud de las series de tiempo (columna 6) y el tipo (columna 7).

En las columnas de la 8 a la 10, se incorporan tres características que son el resultado de un análisis visual presentado en [140]. A partir de este análisis se obtiene la característica de la separación entre clases (columna 8) que determina la facilidad de separar series de tiempo por clases, donde un valor de 1 significa no separable y un valor de 3 significa fácilmente separable. El nivel de ruido (columna 9) también es analizado, donde se establece un criterio de 1 a 5 en el que 1 significa un nivel bajo de ruido mientras que 5 significa un nivel alto de ruido en las series de tiempo. La similitud entre el conjunto de entrenamiento y el de prueba también es analizada, para esta característica se establece un rango de 1 a 3, donde 1 representa una baja similitud y 3 significa una alta similitud entre los conjuntos de datos.

De acuerdo con el análisis de estas tres últimas características en las bases de datos, se espera obtener un mejor resultado en aquellas bases de datos donde la separación entre clases es alta, el nivel de ruido es bajo y existe una alta similitud entre los conjuntos de entrenamiento y de prueba.

7.3.1. CONFIGURACIÓN EXPERIMENTAL

Cada enfoque metaheurístico fue evaluado en diez bases de datos, descritas en la Tabla 7.2. El conjunto de bases de datos fue seleccionado de acuerdo a su dimensionalidad vertical y horizontal, factibles de ser procesadas para obtener resultados en un tiempo de ejecución considerable. Se llevaron a cabo cinco ejecuciones independientes de las versiones del enfoque PM y de las versiones del enfoque SM. Se usaron 3000 evaluaciones como criterio de paro. La configuración usada por cada enfoque se describe en la Tabla 7.3.

Tabla 7.2: Descripción de bases de datos. CS significa separación entre clases, NL significa nivel de ruido y SBS significa similitud entre el conjunto de entrenamiento y prueba.

No.	Nombre	No. de clases	Tamaño conjunto entrenamiento	Tamaño conjunto de prueba	Longitud de serie de tiempo	Tipo	CS	NL	SBS
1.	Beef	5	30	30	470	Spectro [2]	2	1	3
2.	CBF	3	30	900	128	Simulated [151]	2	2	2
3.	Coffee	2	28	28	286	Spectro [17]	2	1	3
4.	ECG200	2	100	100	96	ECG [117]	3	2	3
5.	FaceFour	4	24	88	350	Image [25]	3	1	2
6.	Gun_Point	2	50	150	150	Motion [138]	2	1	1
7.	Lightning-2	2	60	61	637	Sensor [42]	2	3	1
8.	Lightning-7	7	70	73	319	Sensor [42]	1	4	1
9.	OliveOil	4	30	30	570	Spectro [171]	2	4	3
10.	Trace	4	100	100	275	Sensor [148]	2	2	1

Tabla 7.3: Configuración de parámetros de las metaheurísticas. I significa iteraciones, NP es el tamaño de la población, CR es la taza de cruza, F es el factor de escalamiento, N es el número de soluciones a reiniciar, R es la frecuencia de reemplazos y N_k es el tamaño del vecindario.

Metaheurísticas	Algoritmo	Configuración de parámetros
PM	μ-DE versions	I = 500, NP = 6, CR = 0.1, F = 0:9, N = 2, R = 10
SM	LS versions	I = 500, Nk =6

Todos los experimentos se llevaron a cabo en equipos de cómputo con la configuración que se muestra en la Tabla 7.4.

Tabla 7.4: Configuración del equipo de cómputo utilizado en cada experimento.

Sistema Operativo	Windows 7 Pro
CPU	Intel® Xeon® E3-1200 v3
RAM	16 GB

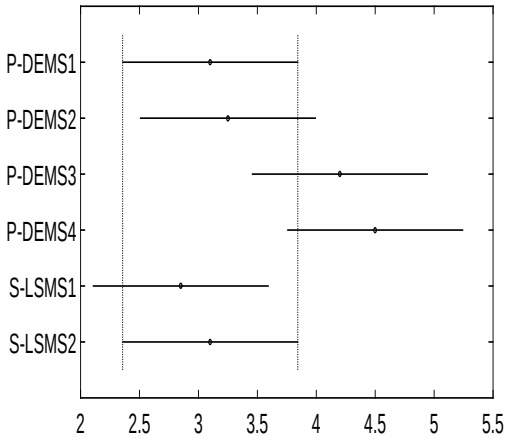
7.3.2. EXPERIMENTO 1. RESULTADOS ESTADÍSTICOS

A partir de las cinco ejecuciones realizadas por cada enfoque, en la Tabla 7.5 se muestran los resultados finales de CVER obtenidos por las seis metaheurísticas. Los valores que se reportan corresponden al promedio de los modelos finales (mejor modelo obtenido en cada ejecución de una metaheurística) evaluados en el conjunto de prueba de cada base de datos.

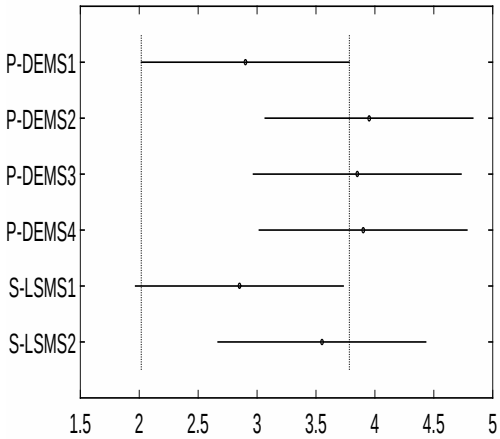
Se llevó a cabo una comparación múltiple no-paramétrica mediante la prueba de Friedman [61]. Dado que las metaheurísticas fueron evaluadas a través de los mismos conjuntos de datos, la prueba de Friedman considera que la hipótesis nula que se prueba es que todas las versiones de metaheurísticas obtienen resultados similares con diferencias no significativas. Por lo tanto, clasifica las metaheurísticas para cada problema por separado, donde la metaheurística que muestra un mejor rendimiento debería tener el rango 1, la segunda mejor rango 2, etc., como se presenta en la Tabla 7.6. Cuando se presentan vínculos o empates, se calculan los rangos promedios.

Con las seis metaheurísticas comparadas y las diez bases de datos utilizadas, el *p-valor* calculado mediante la prueba de Friedman fue de 0.183, aceptando así la hipótesis nula. Por lo tanto, no se encontraron diferencias significativas entre las metaheurísticas comparadas usando la prueba de Friedman. Sin embargo, de acuerdo con el rango promedio que se muestra en la Tabla 7.6, la metaheurística denominada como S-LSMS1 (SM con codificación binaria) fue la mejor ponderada en la mayoría de las bases de datos, seguida por P-DEMS1 (PM basada en μ-DE, donde el vector base es seleccionado al azar y se usa una cruza binomial).

Con el objetivo de validar los resultados estadísticos, se aplicó la prueba post hoc de Tukey basada en los resultados de la prueba de Friedman, donde se utilizaron los mejores valores y los valores de las medianas obtenidos de las cinco ejecuciones independientes de cada metaheurística para el conjunto de las diez bases de datos (resumidos en la Tabla 7.5). Los resultados de la prueba post hoc de Tukey se muestran en la Figura 7.6, donde el eje x presenta el intervalo de confianza en los rangos medios (proporcionado por la prueba de Friedman)



(a)



(b)

Figura 7.6: Prueba post hoc de Tukey utilizando (1) los mejores valores y (2) los valores en las medianas sobre todo el conjunto de bases de datos.

y el eje y muestra el nombre de cada metaheurística comparada. Utilizando los mejores valores, la prueba arrojó un *p-valor* de 0.005, mientras que para los valores de las medianas el *p-valor* fue de 0.250. En el caso de la Figura 7.6(a), se observa una diferencia significativa entre S-LSMS1 y P-DEMS4, siendo S-LSMS1 la metaheurística que reporta los más bajos valores de CVER. En la Figura 7.6(b) no se observan diferencias significativas entre las metaheurísticas comparadas.

Finalmente, como parte del análisis de resultados numéricos se realizó una comparación a pares para determinar cuál de las metaheurísticas muestra un rendimiento diferente en comparación con una metaherística de control seleccionada. La metaheurística que fue considerada como de control fue S-LSMS1, debido a que fue la mejor ponderada por la mayoría de las bases de datos en la prueba de Friedman. Se aplicó la prueba de suma de rangos de Wilcoxon con un 95 % de confianza, para dicha prueba se utilizaron los resultados numéricos obtenidos por cada metaheurística en cada una de las bases de datos. La Tabla 7.6 muestra los resultados numéricos de la comparación por pares, donde las metaheurísticas están ordenadas de acuerdo con el rango promedio proporcionado por la prueba de Friedman.

Los resultados presentados en la Tabla 7.7 muestran que la metaheurística S-LSMS1 fue capaz de obtener resultados más competitivos en la mayoría de las bases de datos en comparación con las otras metaheurísticas. S-LSMS1 superó significativamente a P-DEMS1 en dos bases de datos que son Lightning-7 y Trace, mientras que S-LSMS1 fue superada por P-DEMS1 en las bases de datos Coffee y GunPoint y en seis de las bases de datos el comportamiento entre S-LSMS1 y P-DEMS1 fue similar ya que no se presentaron diferencias significativas. En la comparación entre S-LSMS1 y P-DEMS2, S-LSMS1 logró superar a P-DEMS2 en cuatro bases de datos (Beef, ECG200, Lightning-7 y Trace), mientras que P-DEMS2 fue mejor solo en la base de datos Coffee y en 5 de las 10 fueron iguales. Cuando se compararon S-LSMS1 y S-LSMS2, S-LSMS1 fue significativamente mejor en Beef y Lightning-7 y fue peor en la base de datos Coffee, mientras que en siete de las diez bases de datos no presentaron di-

Tabla 7.5: Mejores valores y medianas de cada metaheurística. Donde B: significa mejor valor y M: significa valor de la mediana.

Base de datos	Stat.	P-DEMS1	P-DEMS2	P-DEMS3	P-DEMS4	S-LSMS1	S-LSMS2
Beef	B	**0.000**	0.033	**0.000**	0.100	**0.000**	0.033
	M	0.000	0.100	0.000	0.167	0.000	0.367
CBF	B	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	M	0.000	0.000	0.000	0.000	0.000	0.049
Coffee	B	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	M	0.000	0.000	0.000	0.000	0.307	0.000
ECG200	B	**0.000**	**0.000**	1.000	1.000	**0.000**	**0.000**
	M	0.000	1.000	1.000	1.000	0.000	0.000
FaceFour	B	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	M	0.000	0.000	0.000	0.000	0.000	0.000
Gun_Point	B	**0.000**	**0.000**	0.493	0.493	**0.000**	**0.000**
	M	0.000	0.493	0.493	0.493	0.480	0.087
Lightning-2	B	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	M	0.000	0.000	0.000	0.000	0.000	0.000
Lightning-7	B	0.737	0.729	0.766	0.739	**0.000**	**0.000**
	M	0.766	0.768	0.794	0.766	0.027	0.095
OliveOil	B	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	M	0.000	0.000	0.000	0.000	0.000	0.200
Trace	B	**0.000**	**0.000**	1.000	1.000	**0.000**	**0.000**
	M	1.000	1.000	1.000	1.000	0.000	0.000

ferencias significativas. S-LSMS1 superó a P-DEMS4 en cuatro bases de datos (Beef, ECG200, Lightning-7 y Trace) y fue superada solo en la base de datos Coffee mientras que en las otras cinco bases de datos no se presentaron diferencias significativas. Finalmente, P-DEMS3 fue superada por S-LSMS1 en tres bases de datos (ECG200, Lightning-7 y Trace) y logró superar a S-LSMS1 en Coffee. En resumen, se puede observar que S-LSMS1 fue capaz de obtener valores competitivos en ocho de diez bases de datos, dichas bases de datos fueron Beef, CBF, ECG200, FaceFour, Lightning-2, Lightning-7, OliveOil y Trace.

A partir del análisis estadístico de los resultados numéricos, se puede decir que entre las dos versiones de las metaheurísticas basadas en un solo punto, la versión donde se utilizó una codificación binaria (S-LSMS1) fue la que reportó mejores resultados numéricos, especialmente en bases de datos que poseen una alta dimensionalidad, el nivel de ruido es alto y el número de clases es mayor a dos. En el caso de las versiones de las metaheurísticas poblacionales, la que logra reportar resultados competitivos en comparación con el resto de las versiones poblacionales es P-DEMS1 que usa la codificación real y utiliza la variante *rand/1/bin*. Es importante remarcar que P-DEMS1 reporta resultados competitivos en bases de datos binarias y donde el nivel de ruido es moderado o bajo.

7.3.3. EXPERIMENTO 2. ANÁLISIS DE CONVERGENCIA

Para llevar a cabo el análisis de convergencia de las metaheurísticas, se consideró el comportamiento promedio de las cinco ejecuciones independientes de cada metaheurística. En las Figuras 7.7-7.11 se muestran las gráficas de convergencia de las bases de datos que fueron

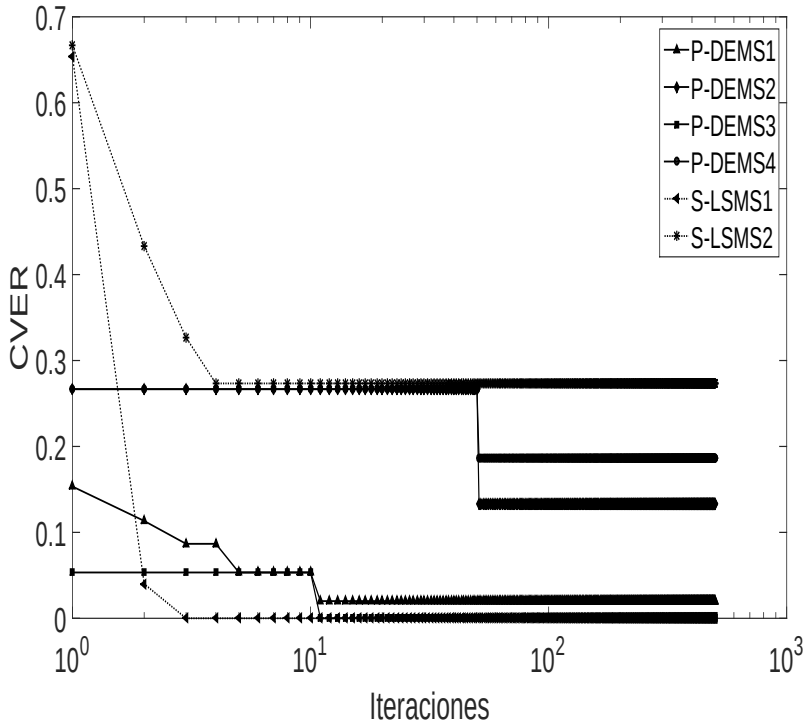
Tabla 7.6: Comparación múltiple del rendimiento promedio de las seis versiones de metaheurísticas por cada base de datos. Los valores a la derecha del símbolo ± representa la desviación estándar y los valores entre paréntesis representan los rangos calculados por la prueba de Friedman. Los valores en **negritas** representan los valores más bajos encontrados o el mejor ranking.

Base de datos	**P-DEMS1**	**P-DEMS2**	**P-DEMS3**	**P-DEMS4**	**S-LSMS1**	**S-LSMS2**
Beef	0.053±0.102 (3)	0.087±0.038 (4)	**0.000±0.000 (1.5)**	0.160±0.060 (5)	**0.000±0.000 (1.5)**	0.367±0.227 (6)
CBF	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	0.030±0.027 (6)
Coffee	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	0.268±0.157 (6)	0.000+0.000 (3)
ECG200	**0.000±0.000 (2)**	0.800±0.447 (4)	1.000±0.000 (5.5)	1.000±0.000 (5.5)	**0.000±0.000 (2)**	**0.000±0.000 (2)**
FaceFour	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**	**0.000±0.000 (3.5)**
Gun_Point	**0.000±0.000 (1)**	0.395±0.221 (4)	0.493±0.000 (5.5)	0.493±0.000 (5.5)	0.388±0.217 (3)	0.212±0.253 (2)
Lightning-2	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	**0.000±0.000 (3)**	0.069±0.154 (6)
Lightning-7	0.766±0.035 (5)	0.762±0.028 (4)	0.786±0.019 (6)	0.761±0.012 (3)	**0.019±0.019 (1)**	0.082±0.046 (2)
OliveOil	0.013±0.030 (2.5)	0.033±0.047 (5)	0.027±0.043 (4)	0.013±0.018 (2.5)	**0.000±0.000 (1)**	0.133±0.122 (6)
Trace	0.800±0.447 (3.5)	0.800±0.447 (3.5)	1.000±0.000 (5.5)	1.000±0.000 (5.5)	**0.000±0.000 (1.5)**	**0.000±0.000 (1.5)**
Rango promedio	2.950	3.700	4.050	3.950	**2.550**	3.800

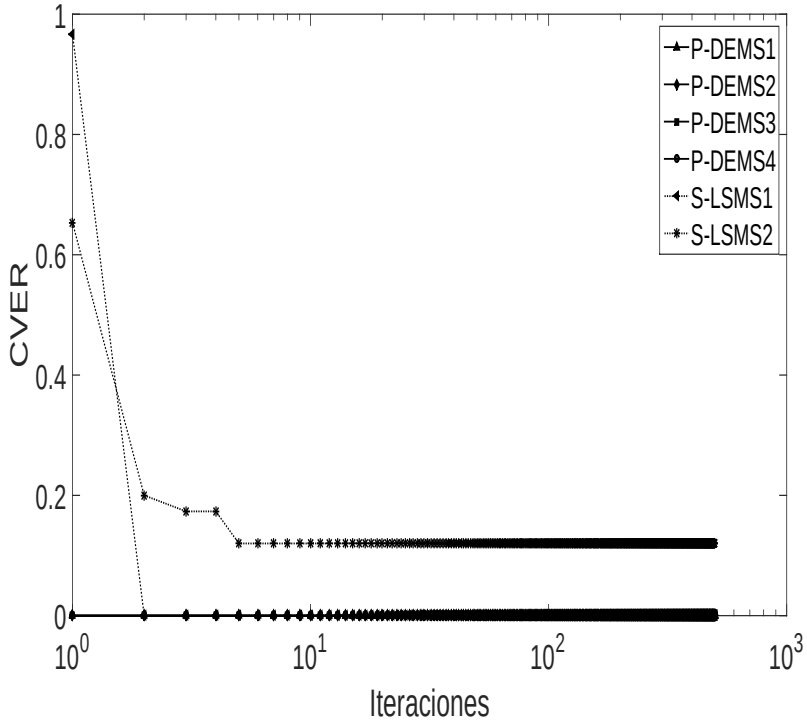
Tabla 7.7: Comparación por pares entre la metaheurística de control S-LSMS1 y el resto de las metaheurísticas. (-) significa que se presentó una diferencia significativa a favor de la metaheurística de control. (+) implica que hubo una diferencia significativa a favor de la metaheurística comparada. (=) significa que no presentó diferencia significativa entre las metaheurísticas comparadas. Los valores en **negritas** representan los mejores valores encontrados.

Database	**S-LSMS1** 1° (Control)	**P-DEMS1** 2°	**P-DEMS2** 3°	**S-LSMS2** 4°	**P-DEMS4** 5°	**P-DEMS3** 6°
Beef	**0.000±0.000**	0.053±0.102 (=)	0.087±0.038 (-)	0.367±0.227 (-)	0.160±0.060 (-)	**0.000±0.000** (=)
CBF	**0.000±0.000**	**0.000±0.000 (=)**	**0.000±0.000** (=)	0.030±0.027 (=)	**0.000±0.000** (=)	0.000±0.000 (=)
Coffee	0.268±0.157	**0.000±0.000** (+)	**0.000±0.000** (+)	**0.000±0.000** (+)	**0.000±0.000** (+)	**0.000±0.000** (+)
ECG200	**0.000±0.000**	**0.000±0.000** (=)	0.800±0.447 (-)	**0.000±0.000** (=)	1.000±0.000 (-)	1.000±0.000 (-)
FaceFour	**0.000±0.000**	**0.000±0.000** (=)	**0.000±0.000** (=)	**0.000±0.000** (=)	**0.000±0.000** (=)	**0.000±0.000** (=)
GunPoint	0.388±0.217	**0.000±0.000** (+)	0.395±0.221 (=)	0.212±0.253 (=)	0.493±0.000 (=)	0.493±0.000 (=)
Lightning-2	**0.000±0.000**	**0.000±0.000** (=)	**0.000±0.000** (=)	0.069±0.154 (=)	**0.000±0.000** (=)	**0.000±0.000** (=)
Lightning-7	**0.019±0.019**	0.766±0.035 (-)	0.762±0.028 (-)	0.082±0.046 (-)	0.761±0.012 (-)	0.786±0.019 (-)
OliveOil	**0.000±0.000**	0.013±0.030 (=)	0.033±0.047 (=)	0.133±0.122 (=)	0.013±0.018 (=)	0.027±0.043 (=)
Trace	**0.000±0.000**	0.800±0.447 (-)	0.800±0.447 (-)	**0.000±0.000** (=)	1.000±0.000 (-)	1.000±0.000 (-)
Número de (-)		2	4	2	4	3
Número de (+)		2	1	1	1	1
Número de (=)		6	5	7	5	6

consideradas. El eje x representa el número de iteraciones realizadas por cada metaheurística y el eje y muestra el valor de la función objetivo obtenido en cada iteración. El eje x es presentado en una escala logarítmica para una mejor visión de los resultados. A partir de las gráficas de convergencia, se puede observar que para el caso del enfoque poblacional se sugiere el uso de P-DEMS1. Sin embargo, en algunos casos como la base de datos Trace, P-DEMS1 quedó atrapada en un óptimo local. Respecto a los enfoques SM, la opción de S-LSMS2 (que usa representación mixta) logró una convergencia más rápida respecto a S-LSMS1 (que usa representación binaria), pero se observó un mayor estancamiento en óptimos locales en bases de datos como Beef, CBF, Lightning-2 o OliveOil, mientras que S-LSMS2 logra encontrar valores más prometedores conforme avanza el número de iteraciones. Un hallazgo importante fue que P-DEMS1 fue capaz de obtener una mejora significativa en el valor de la función de aptitud en iteraciones tempranas, es decir, antes de las 100 iteraciones, en la mayoría de las bases de datos consideradas en la experimentación. Sin embargo S-LSMS1 fue capaz de conseguir resultados competitivos al final del proceso de búsqueda.

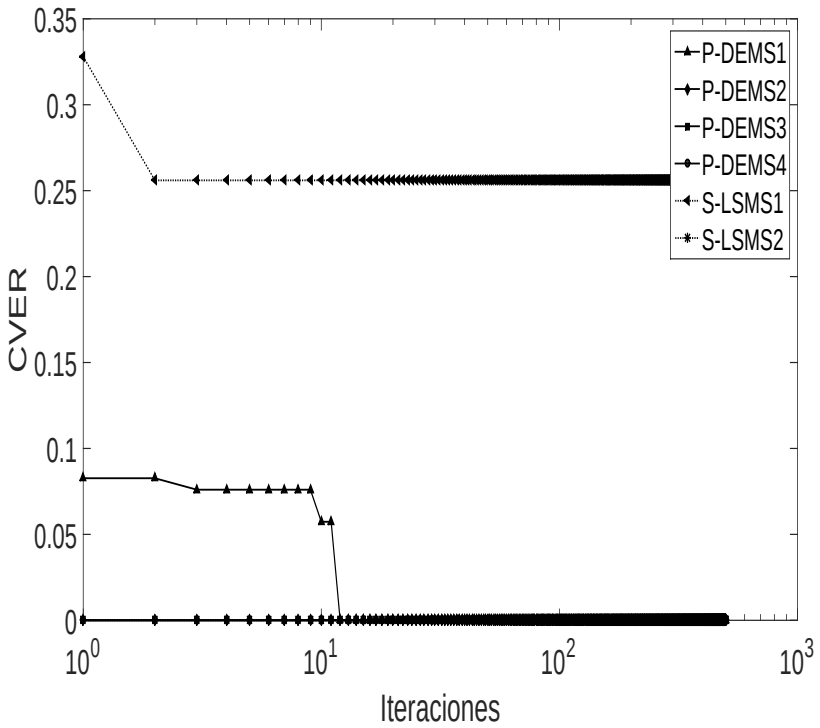


Beef

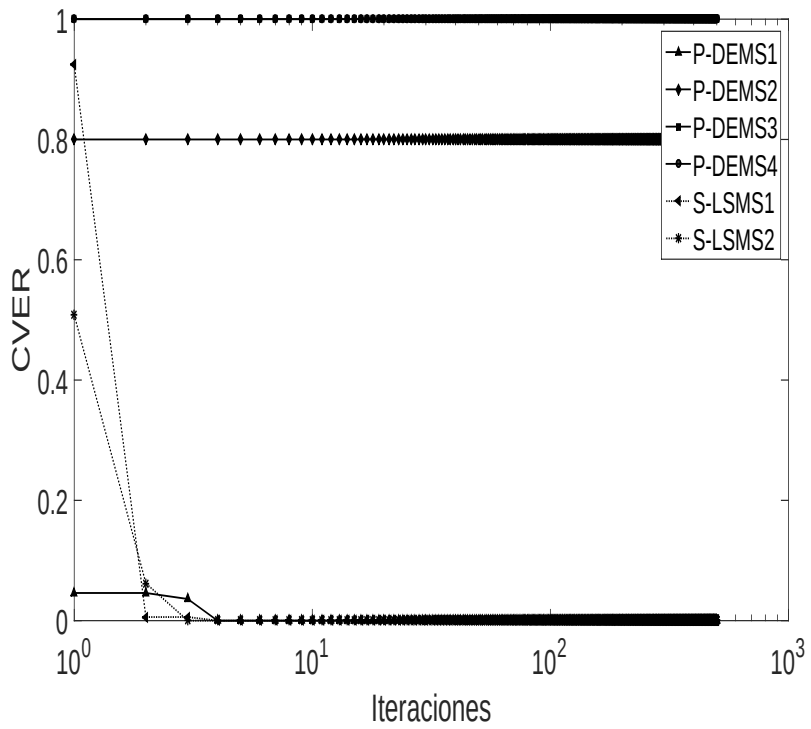


CBF

Figura 7.7: Gráficas de convergencia para las bases de datos Beef y CBF.

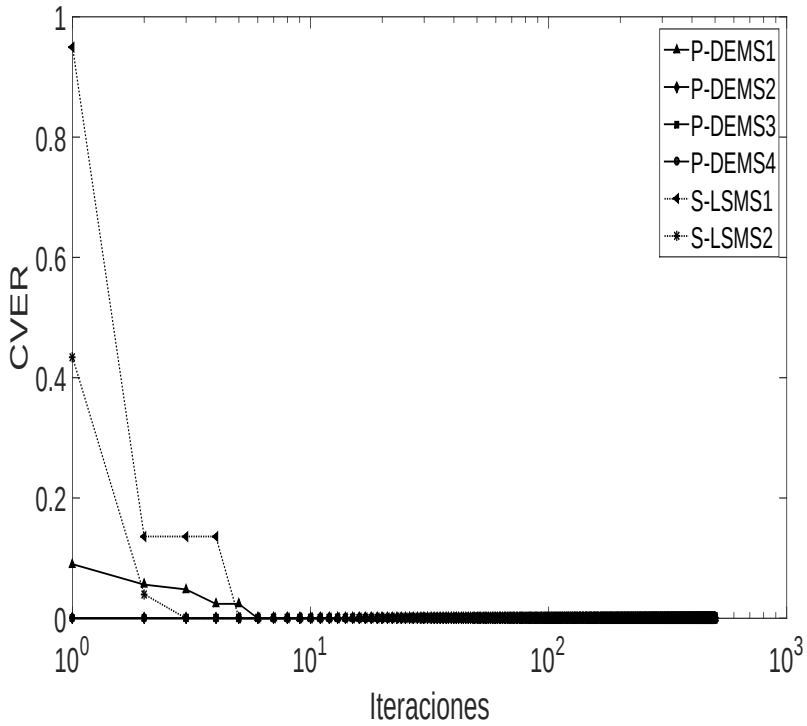


Coffee

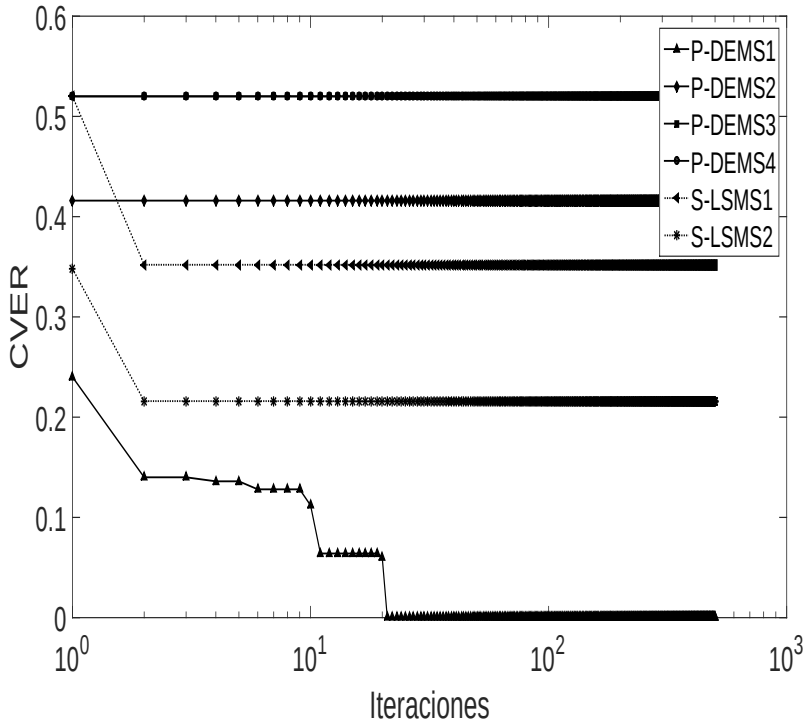


ECG200

Figura 7.8: Gráficas de convergencia para las bases de datos Coffee y ECG200.

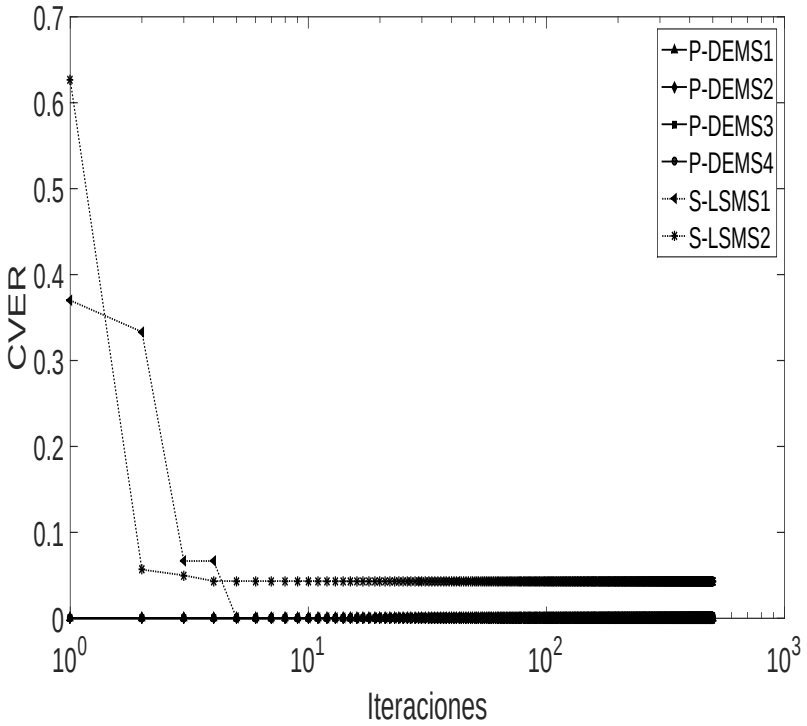


FaceFour

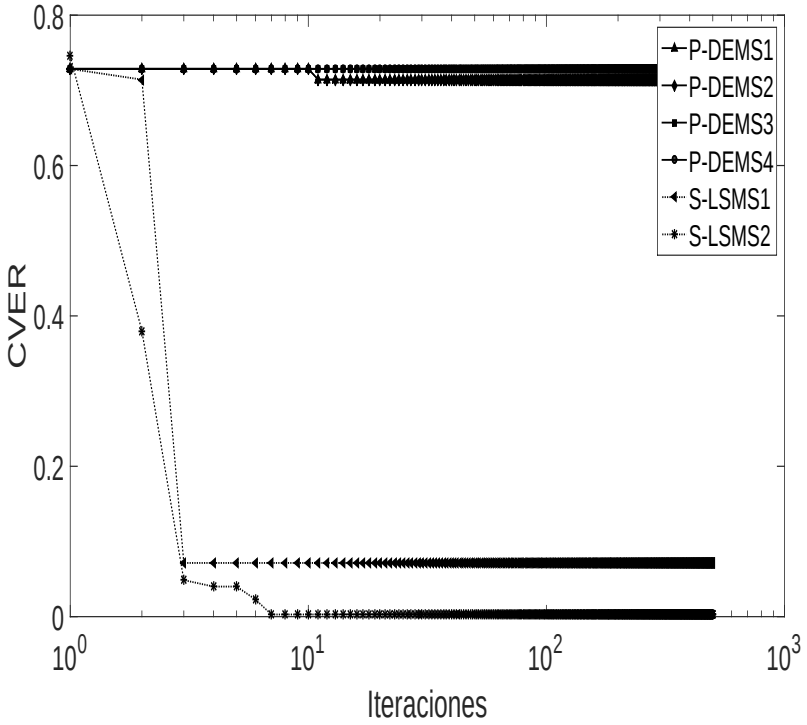


GunPoint

Figura 7.9: Gráficas de convergencia para las bases de datos FaceFour y GunPoint.

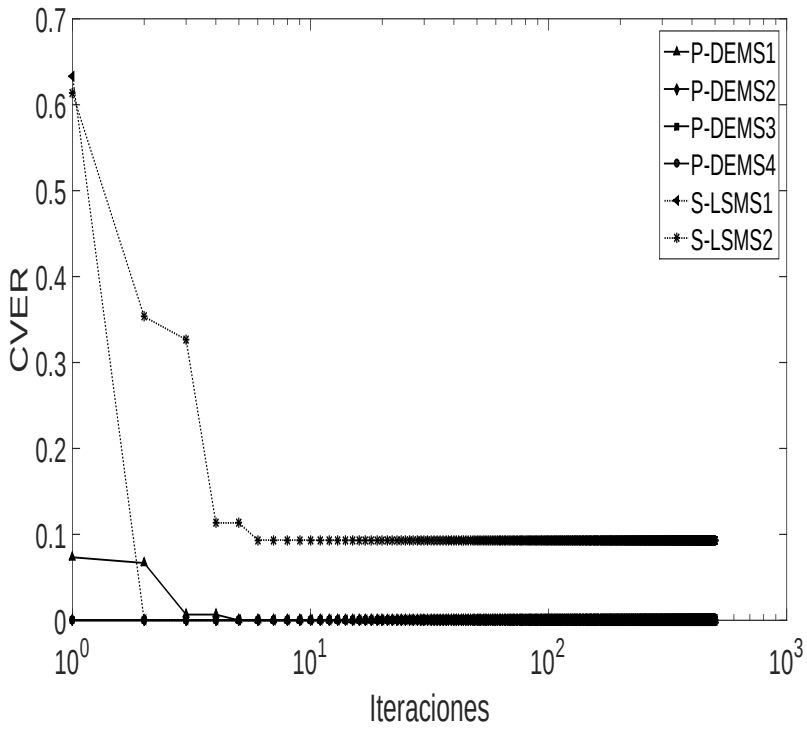


Lightning2

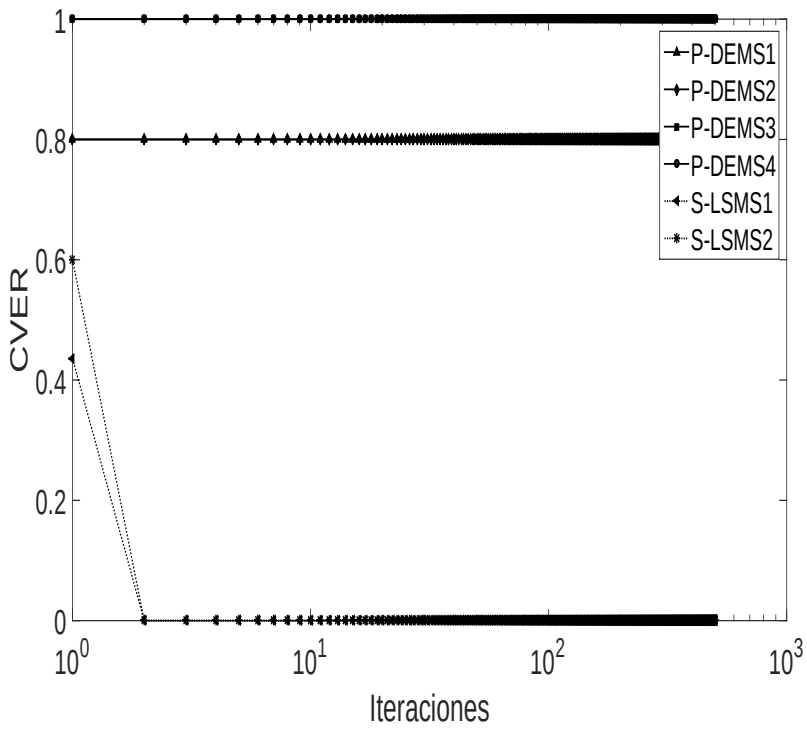


Lightning7

Figura 7.10: Gráficas de convergencia para las bases de datos Lightning2 y Lightning7.



OliveOil



Trace

Figura 7.11: Gráficas de convergencia para las bases de datos OliveOil y Trace.

7.3.4. EXPERIMENTO 3. ANÁLISIS DE DIVERSIDAD EN EL ENFOQUE PM

En esta sección se muestra un breve análisis de la diversidad de la población en los enfoques PM. La medida de diversidad utilizada en estos experimentos está basada en el trabajo de Morrison, et al. [110]. Dicha medida se centra en la extensión del concepto del momento de la inercia donde se pretende realizar la medición de la distribución de la masa en espacios que pueden ser clasificados como espacios de alta dimensionalidad para la medición de la diversidad de la población. Esta medida de diversidad calcula el centroide $C = (c_1, c_2, c_3, ..., c_n)$ de los puntos o vectores P en un espacio de dimensión N, donde cada c_i es calculado a través de la Ecuación 7.7.

$$c_i = \frac{\sum_{j=1}^{j=P} x_{ij}}{P} \tag{7.7}$$

donde $x_{ij} \in \mathbb{R}$ representa el i-ésimo valor de un vector. Esta medida es aplicada por cada vector de la población actual y es definida por la Ecuación 7.8. Posteriormente, se calcula la desviación estándar σ del conjunto I.

$$I = \sum_{i=1}^{i=N} \sum_{j=1}^{j=P} (x_{ij} - c_i)^2 \tag{7.8}$$

Durante cada ejecución independiente de cada variante del algoritmo P-DEMS1, por cada base de datos, se fue midiendo la diversidad de la soluciones. Como resultado de esta medición se obtuvo la grafica que se muestra en la Figura 7.12 que muestra el comportamiento global de la diversidad de las soluciones para cada una de las variantes de P-DEMS.

Se puede observar que la variante P-DEMS1, cuyo vector base es seleccionado de forma aleatoria y hace uso de la cruza binomial, reporta mejores valores en la medida de diversidad. Dado que la medida de diversidad está basada en la distribución de las soluciones, los resultados mostrados se pueden traducir como: a mayores valores de la medida de diversidad existe una mayor área entre las soluciones y son menos parecidas unas entre otras.

Por otro lado, la variante P-DEMS3, que usa el esquema *best/1/bin*, reporta los peores valores en la medida de diversidad a nivel global. Por lo tanto se puede decir que en la búsqueda de modelos completos donde están involucradas bases de datos temporales se prefiere el esquema *rand/1/bin* para obtener una mejor diversidad de las soluciones durante la búsqueda.

7.3.5. EXPERIMENTO 4. ANÁLISIS DE LOS MODELOS FINALES

A partir de la ejecución de los enfoques propuestos, en la Tabla 7.8 y la Tabla 7.9 se presentan los mejores modelos encontrados por cada enfoque y por cada una de las bases de datos. En la tercera columna se detalla el modelo obtenido, la cuarta y quinta columna muestran el CVER durante el entrenamiento y prueba respectivamente. La sexta columna muestra el tiempo de ejecución, en minutos, que se requirió para obtener el modelo durante una ejecución, mientras que la última columna muestra el tiempo de ejecución cuando el modelo es evaluado en la base de datos de prueba de cada base de datos.

Se puede observar que el método de suavizado mayormente sugerido fue el de promedio móvil. Respecto a la representación de las series de tiempo o reducción de dimensionalidad, el método PAA fue el más sugerido por los enfoques de búsqueda. En relación con la reducción pero en términos de instancias, denominado como reducción de numerosidad, el

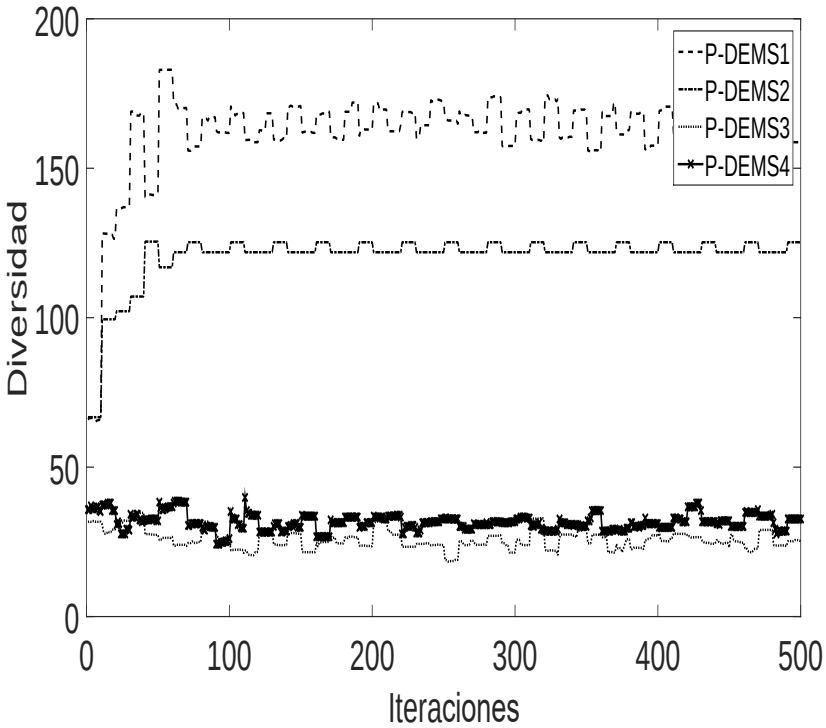


Figura 7.12: Gráfica de diversidad global, donde se promedió cada una de las iteraciones de todas las bases de datos.

método INSIGHT fue el método más popular. Referente a la tarea de clasificación, el árbol de decisión y AdaBooost fueron los métodos más sugeridos.

A partir de los modelos finales, se puede observar que existen diferentes opciones con valores similares en CVER para determinadas bases de datos. Estas bases de datos fueron identificadas como problemas multi-modales, es decir, que pueden presentar múltiples opciones de soluciones. Tal es el caso de Beef, Coffee, Gun_Point y Lightning-7, las cuales muestran más diversificación en la estructura de sus respectivos modelos finales y en el valor de sus hiper-parámetros.

El tiempo de ejecución de los modelos varía considerablemente debido a las características de las bases de datos y los métodos seleccionados para llevar a cabo una sub-tarea específica. En general, P-DEMS3 muestra el tiempo de ejecución más bajo, mientras que S-LSMS1 fue el enfoque más costoso. Para el caso de P-DEMS3 resulta ser el enfoque poblacional más barato, computacionalmente hablando, pero se caracteriza por tener una baja diversidad y tiende a quedarse en óptimos locales tal y como se puede ver en las gráficas de convergencia anteriormente presentadas. Por otro lado, S-LSMS1 es el enfoque basado en una solución, que usa una codificación binaria, que resulta ser el enfoque más costoso. Sin embargo, S-LSMS1 reporta los valores más bajos para CVER durante la etapa de entrenamiento y resultados competitivos en la fase de prueba. La representación gráfica de cada modelo final se puede visualizar en al Anexo 11.

Tabla 7.8: Mejores modelos finales de las bases de datos Beef, CBF, Coffee, ECG200 y FaceFour obtenidos por cada meta-heurística.

Base de datos	Enfoque	Modelo	CVER Entrenamiento	CVER Prueba	(m) Tiempo Entrenamiento	(m) Tiempo Prueba
Beef	P-DEMS1	'S:SG{k=4,f=375},R:PAA{nseg=103},NR:MC1{%ins=0.99961,nitera=85},C:ADABOOST{nTree=403}'	0.0000	0.0000	1617.2000	0.1164
	P-DEMS2	'S:MA{span=241},R:PAA{nseg=75},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=114}'	0.1333	0.1333	693.2000	0.0769
	P-DEMS3	'S:MA{span=45},R:SAX{nseg=55,alphabet=7},NR:INSIGTH{%ins=0.98017},C:ADABOOST{nTree=423}'	0.0000	0.0000	764.0000	0.1195
	P-DEMS4	'S:MA{span=241},R:PAA{nseg=75},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=242}'	0.1333	0.2333	816.4000	0.1388
	S-LSMS1	'S:LRe{%span=0.95622},R:PAA{nseg=45},NR:MC1{%ins=0.8254,nitera=194},C:ADABOOST{nTree=369}'	0.0000	0.0000	737.0000	0.1901
	S-LSMS2	'S:SG{k=5,f=445},R:PCA{%red=0.87636},NR:INSIGTH{%ins=0.89898},C:ADABOOST{nTree=180}'	0.1333	0.0333	2293.8000	0.2033
CBF	P-DEMS1	'S:LRe{%span=0.89734},R:PCA{%red=0.38038},NR:MC1{%ins=0.80504,nitera=25},C:TREE{typeSC=3(deviance),maxCat=14}'	0.0000	0.0000	365.6000	24.7179
	P-DEMS2	'S:MA{span=67},R:PAA{nseg=21},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=436}'	0.0000	0.0000	609.2000	27.1500
	P-DEMS3	'S:MA{span=51},R:PAA{nseg=49},NR:MC1{%ins=0.83219,nitera=388},C:TREE{typeSC=2(twoing),maxCat=8}'	0.0000	0.0000	335.6000	29.7145
	P-DEMS4	'S:MA{span=67},R:PAA{nseg=21},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=436}'	0.0000	0.0000	634.2000	27.7813
	S-LSMS1	'S:LRw{%span=0.43313},R:PAA{nseg=32},NR:INSIGTH{%ins=0.5873},C:ADABOOST{nTree=330}'	0.0000	0.0000	465.2000	4.3729
	S-LSMS2	'S:SG{k=5,f=113},R:SAX{nseg=63,alphabet=7},NR:INSIGTH{%ins=0.89898},C:TREE{typeSC=3(deviance),maxCat=10}'	0.0000	0.0000	295.4000	0.2547
Coffee	P-DEMS1	'S:SG{k=1,f=65},R:SAX{nseg=104,alphabet=17},NR:INSIGTH{%ins=0.99285},C:TREE{typeSC=2(twoing),maxCat=18}'	0.0000	0.0000	489.4000	0.0156
	P-DEMS2	'S:MA{span=73},R:PAA{nseg=137},NR:INSIGTH{%ins=0.74371},C:TREE{typeSC=3(deviance),maxCat=12}'	0.0000	0.0000	25.6000	0.0006
	P-DEMS3	'S:MA{span=73},R:PAA{nseg=137},NR:INSIGTH{%ins=0.74371},C:TREE{typeSC=3(deviance),maxCat=12}'	0.0000	0.0000	25.6000	0.0006
	P-DEMS4	'S:MA{span=73},R:PAA{nseg=137},NR:INSIGTH{%ins=0.74371},C:TREE{typeSC=3(deviance),maxCat=12}'	0.0000	0.0000	25.0000	0.0006
	S-LSMS1	'S:LRw{%span=0.36973},R:PCA{%red=0.51969},NR:MC1{%ins=0.53968,nitera=372},C:KNN-LBDTW{k=4,r=0.048141}'	0.3200	0.2733	190.2000	0.0393
	S-LSMS2	'S:SG{k=3,f=89},R:PAA{nseg=57},NR:INSIGTH{%ins=0.89898},C:TREE{typeSC=3(deviance),maxCat=15}'	0.0000	0.0000	124.2000	0.0013
ECG200	P-DEMS1	'S:SG{k=5,f=51},R:SAX{nseg=11,alphabet=11},NR:INSIGTH{%ins=0.98538},C:TREE{typeSC=3(deviance),maxCat=12}'	0.0000	0.0000	1031.6000	0.0950
	P-DEMS2	'S:MA{span=25},R:PAA{nseg=46},NR:INSIGTH{%ins=0.74371},C:TREE{typeSC=3(deviance),maxCat=12}'	0.0000	0.0000	44.4000	0.0027
	P-DEMS3	'S:MA{span=75},R:PAA{nseg=8},NR:INSIGTH{%ins=0.78967},C:NB{typeK=4(triangle)}'	1.0000	1.0000	45.0000	0.0029
	P-DEMS4	'S:MA{span=75},R:PAA{nseg=8},NR:INSIGTH{%ins=0.78967},C:NB{typeK=4(triangle)}'	1.0000	1.0000	45.0000	0.0029
	S-LSMS1	'S:LRw{%span=0.91243},R:PAA{nseg=23},NR:MC1{%ins=0.70635,nitera=103},C:TREE{typeSC=1(gdi),maxCat=14}'	0.0000	0.0000	1984.0000	0.3617
	S-LSMS2	'S:SG{k=1,f=87},R:SAX{nseg=7,alphabet=17},NR:INSIGTH{%ins=0.95157},C:TREE{typeSC=3(deviance),maxCat=9}'	0.0000	0.0000	819.0000	0.0295
FaceFour	P-DEMS1	'S:SG{k=1,f=325},R:PAA{nseg=2},NR:INSIGTH{%ins=0.82472},C:ADABOOST{nTree=55}'	0.0000	0.0000	1313.6000	0.2044
	P-DEMS2	'S:LRe{%span=0.11784},R:PAA{nseg=72},NR:INSIGTH{%ins=0.78492},C:ADABOOST{nTree=102}'	0.0000	0.0000	290.4000	0.0520
	P-DEMS3	'S:LRe{%span=0.11784},R:PAA{nseg=72},NR:INSIGTH{%ins=0.78492},C:ADABOOST{nTree=102}'	0.0000	0.0000	269.2000	0.0512
	P-DEMS4	'S:LRe{%span=0.11784},R:PAA{nseg=72},NR:INSIGTH{%ins=0.78492},C:ADABOOST{nTree=102}'	0.0000	0.0000	268.8000	0.0512
	S-LSMS1	'S:LRe{%span=0.50762},R:PAA{nseg=109},NR:MC1{%ins=0.63492,nitera=159},C:ADABOOST{nTree=443}'	0.0000	0.0000	873.6000	0.8100
	S-LSMS2	'S:LRe{%span=0.33232},R:SAX{nseg=105,alphabet=8},NR:INSIGTH{%ins=0.97784},C:ADABOOST{nTree=152}'	0.0000	0.0000	703.8000	0.2331

Tabla 7.9: Mejores modelos finales de las bases de datos Gun_Point, Lightning-2, Lightning-7, OliveOil y Trace.

Base de datos	Enfoque	Modelo	CVER Entrenamiento	CVER Prueba	(m) Tiempo Entrenamiento	(m) Tiempo Prueba
Gun_Point	P-DEMS1	'S:SG{k=3,f=131},R:SAX{nseg=44,alphabet=12},NR:INSIGTH{%ins=0.76904},C:TREE{typeSC=3(deviance),maxCat=14}'	0.0000	0.0000	272.0000	0.2588
	P-DEMS2	'S:LRe{%span=0.17202},R:SAX{nseg=18,alphabet=5},NR:INSIGTH{%ins=0.78438},C:TREE{typeSC=2(twoing),maxCat=6}'	0.0000	0.0000	208.2000	0.1852
	P-DEMS3	'S:SG{k=5,f=83},R:PCA{%red=0.79674},NR:MC1{%ins=0.66346,nitera=141},C:ADABOOST{nTree=458}'	0.5200	0.4933	214.6000	0.2253
	P-DEMS4	'S:SG{k=5,f=83},R:PCA{%red=0.79674},NR:MC1{%ins=0.66346,nitera=141},C:ADABOOST{nTree=458}'	0.5200	0.4933	202.0000	0.2247
	S-LSMS1	'S:LRw{%span=0.28211},R:PCA{%red=0.27717},NR:INSIGTH{%ins=0.70635},C:TREE{typeSC=3(deviance),maxCat=14}'	0.0000	0.0000	933.8000	0.0854
	S-LSMS2	'S:SG{k=4,f=109},R:PCA{%red=0.6856},NR:MC1{%ins=0.66141,nitera=421},C:TREE{typeSC=1(gdi),maxCat=20}'	0.0000	0.0000	1084.8000	0.7672
Lightning-2	P-DEMS1	'S:LRw{%span=0.28222},R:PCA{%red=0.69059},NR:INSIGTH{%ins=0.7557},C:TREE{typeSC=1(gdi),maxCat=6}'	0.0000	0.0000	1096.8000	0.1757
	P-DEMS2	'S:LRw{%span=0.28222},R:PCA{%red=0.69059},NR:INSIGTH{%ins=0.7557},C:TREE{typeSC=1(gdi),maxCat=6}'	0.0000	0.0000	1093.4000	0.1754
	P-DEMS3	'S:LRw{%span=0.28222},R:PCA{%red=0.69059},NR:INSIGTH{%ins=0.7557},C:TREE{typeSC=1(gdi),maxCat=6}'	0.0000	0.0000	1099.8000	0.1758
	P-DEMS4	'S:LRw{%span=0.28222},R:PCA{%red=0.69059},NR:INSIGTH{%ins=0.7557},C:TREE{typeSC=1(gdi),maxCat=6}'	0.0000	0.0000	1099.8000	0.1757
	S-LSMS1	'S:MA{span=39},R:SAX{nseg=270,alphabet=8},NR:INSIGTH{%ins=0.51587},C:TREE{typeSC=1(gdi),maxCat=15}'	0.0000	0.0000	10147.8000	0.0047
	S-LSMS2	'S:SG{k=5,f=377},R:PCA{%red=0.84105},NR:INSIGTH{%ins=0.9274},C:TREE{typeSC=2(twoing),maxCat=5}'	0.0000	0.0000	2386.0000	0.1307
Lightning-7	P-DEMS1	'S:MA{span=69},R:SAX{nseg=65,alphabet=5},NR:MC1{%ins=0.69909,nitera=424},C:NB{typeK=4(triangle)}'	0.6571	0.7371	231.2000	0.0660
	P-DEMS2	'S:LRw{%span=0.16923},R:PCA{%red=0.84005},NR:INSIGTH{%ins=0.89798},C:SVM{-t=2(radial),-g=45.7029,-c=324.839}'	0.7286	0.7943	221.6000	0.0654
	P-DEMS3	'S:LRw{%span=0.16923},R:PCA{%red=0.84005},NR:INSIGTH{%ins=0.89798},C:SVM{-t=2(radial),-g=45.7029,-c=324.839}'	0.7286	0.7943	202.4000	0.0175
	P-DEMS4	'S:LRw{%span=0.16923},R:PCA{%red=0.84005},NR:INSIGTH{%ins=0.89798},C:SVM{-t=2(radial),-g=45.7029,-c=324.839}'	0.7286	0.7676	200.6000	0.0176
	S-LSMS1	'S:SG{k=1,f=211},R:PCA{%red=0.64094},NR:INSIGTH{%ins=0.81746},C:ADABOOST{nTree=296}'	0.0714	0.0267	1501.0000	0.1151
	S-LSMS2	'S:MA{span=213},R:SAX{nseg=72,alphabet=16},NR:INSIGTH{%ins=0.89749},C:TREE{typeSC=3(deviance),maxCat=5}'	0.0000	0.0952	2575.6000	0.1608
OliveOil	P-DEMS1	'S:MA{span=331},R:PCA{%red=0.6962},NR:MC1{%ins=0.72457,nitera=136},C:ADABOOST{nTree=250}'	0.0000	0.0000	1418.0000	0.1899
	P-DEMS2	'S:MA{span=291},R:PAA{nseg=91},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=436}'	0.0000	0.0000	729.4000	0.1794
	P-DEMS3	'S:MA{span=291},R:PAA{nseg=91},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=436}'	0.0000	0.0000	578.8000	0.1755
	P-DEMS4	'S:MA{span=291},R:PAA{nseg=91},NR:MC1{%ins=0.81403,nitera=400},C:ADABOOST{nTree=436}'	0.0000	0.0000	776.6000	0.1778
	S-LSMS1	'S:LRw{%span=0.50883},R:PCA{%red=0.43701},NR:INSIGTH{%ins=0.89683},C:ADABOOST{nTree=330}'	0.0000	0.0000	1151.2000	0.2317
	S-LSMS2	'S:SG{k=5,f=191},R:SAX{nseg=235,alphabet=7},NR:INSIGTH{%ins=0.89898},C:ADABOOST{nTree=226}'	0.0000	0.0000	2194.6000	0.2602
Trace	P-DEMS1	'S:LRw{%span=0.68959},R:SAX{nseg=18,alphabet=11},NR:INSIGTH{%ins=0.79263},C:ADABOOST{nTree=275}'	0.0000	0.0000	235.4000	0.1084
	P-DEMS2	'S:MA{span=25},R:PAA{nseg=116},NR:INSIGTH{%ins=0.95765},C:ADABOOST{nTree=138}'	0.0000	0.0000	112.4000	0.0209
	P-DEMS3	'S:MA{span=213},R:PAA{nseg=19},NR:INSIGTH{%ins=0.78967},C:NB{typeK=4(triangle)}'	1.0000	1.0000	104.4000	0.0113
	P-DEMS4	'S:MA{span=213},R:PAA{nseg=19},NR:INSIGTH{%ins=0.78967},C:NB{typeK=4(triangle)}'	1.0000	1.0000	104.0000	0.0116
	S-LSMS1	'S:MA{span=103},R:PAA{nseg=18},NR:INSIGTH{%ins=0.92063},C:ADABOOST{nTree=328}'	0.0000	0.0000	3127.6000	0.3839
	S-LSMS2	'S:SG{k=1,f=253},R:SAX{nseg=18,alphabet=17},NR:INSIGTH{%ins=0.95157},C:TREE{typeSC=3(deviance),maxCat=9}'	0.0000	0.0000	1820.8000	0.0207

7.3.6. EXPERIMENTO 5. ANÁLISIS DE FRECUENCIA DEL USO DE LOS MÉTODOS

Con el fin de enriquecer el análisis de las soluciones que fueron mayormente seleccionadas, a continuación se presenta un análisis de la frecuencia de cada uno de los diferentes métodos que fueron considerados para construir y sugerir modelos completos, mediante el proceso de búsqueda realizado por las variantes del enfoque poblacional y el enfoque basado en un solo punto. La Figura 7.13 muestra la frecuencia promedio de selección de cada uno de los métodos. La frecuencia fue calculada a partir de las cinco ejecuciones independientes de cada metaheurística (P-DEMS y S-LSMS) sobre cada una de las diez bases de datos usadas para la experimentación. Durante el proceso de búsqueda, cada método tenía un contador que se iba incrementando cada vez que dicho método era seleccionado para formar una solución candidata de modelo completo. Los valores de la frecuencia de selección donde se usaron las variantes de P-DEMS se basaron en 601,200 modelos evaluados, mientras que las variantes de S-LSMS se basaron en 300,600 modelos. Respecto a las opciones de suavizado, S2, que corresponde al método de Promedio Móvil fue el método más sugerido por el enfoque PM, mientras que S1 (Sgolay) fue seleccionado por el enfoque SM. En el caso de los métodos para la representación de las series de tiempo, R2 (que corresponde al método PAA) fue el método más preferido por ambos enfoques de las metaherísticas. Para el tratamiento de la reducción de la numerosidad de las bases de datos, INSIGHT fue el método que se presentó con mayor frecuencia. En la etapa de clasificación, el método C5 que corresponde al ensamble fue el más frecuente en ambos enfoques de búsqueda.

7.4. CONCLUSIONES

En este capítulo se presentó una comparación entre dos enfoques de metaheurísticas para tratar con el problema de selección de modelo completo en bases de datos temporales. El primer enfoque se basa en la versión micro del algoritmo poblacional Evolución Diferencial, denominado en este trabajo como μ-DEMS, del cual se evaluaron cuatro variantes basadas en rand/1/bin (P-DEMS1), rand/1/exp (P-DEMS2), best/1/bin (P-DEMS3) y best/1/exp (P-DEMS4). El segundo enfoque se centra en evaluar el comportamiento de una de las metaheurísticas más simples basadas en una solución, el buscador local, del cual se generó una versión para codificación binaria (S-LSMS1) y otra versión para codificación mixta (s-LSMS2).

En total se evaluaron seis opciones de búsqueda de modelos completos, de las cuales 4 son las variantes de P-DEMS y dos de S-LSMS. Cada una de las variantes fue evaluada en diez bases de datos temporales diferentes que cuentan con un conjunto de datos para el entrenamiento y otro para la prueba. El conjunto de experimentos que se llevaron a cabo se dividió en cinco partes que son: el análisis estadístico de los resultados numéricos, el análisis de las gráficas de convergencia, un análisis de diversidad enfocada solo en las variantes poblacionales, el análisis de los modelos finales y el análisis de frecuencia de selección de los métodos involucrados para formar un modelo completo.

A partir de estos experimentos se listan a continuación algunas conclusiones y hallazgos importantes:

- El análisis estadístico sugiere que S-LSMS1 (variante con codificación binaria) es la mejor opción cuando se trabaja con bases de datos temporales que poseen una dimensionalidad alta, son ruidosas y el número de clases es mayor a dos. S-LSMS1 posee la ventaja de ser simple en su estructura además de ser libre de parámetros. Sin embargo

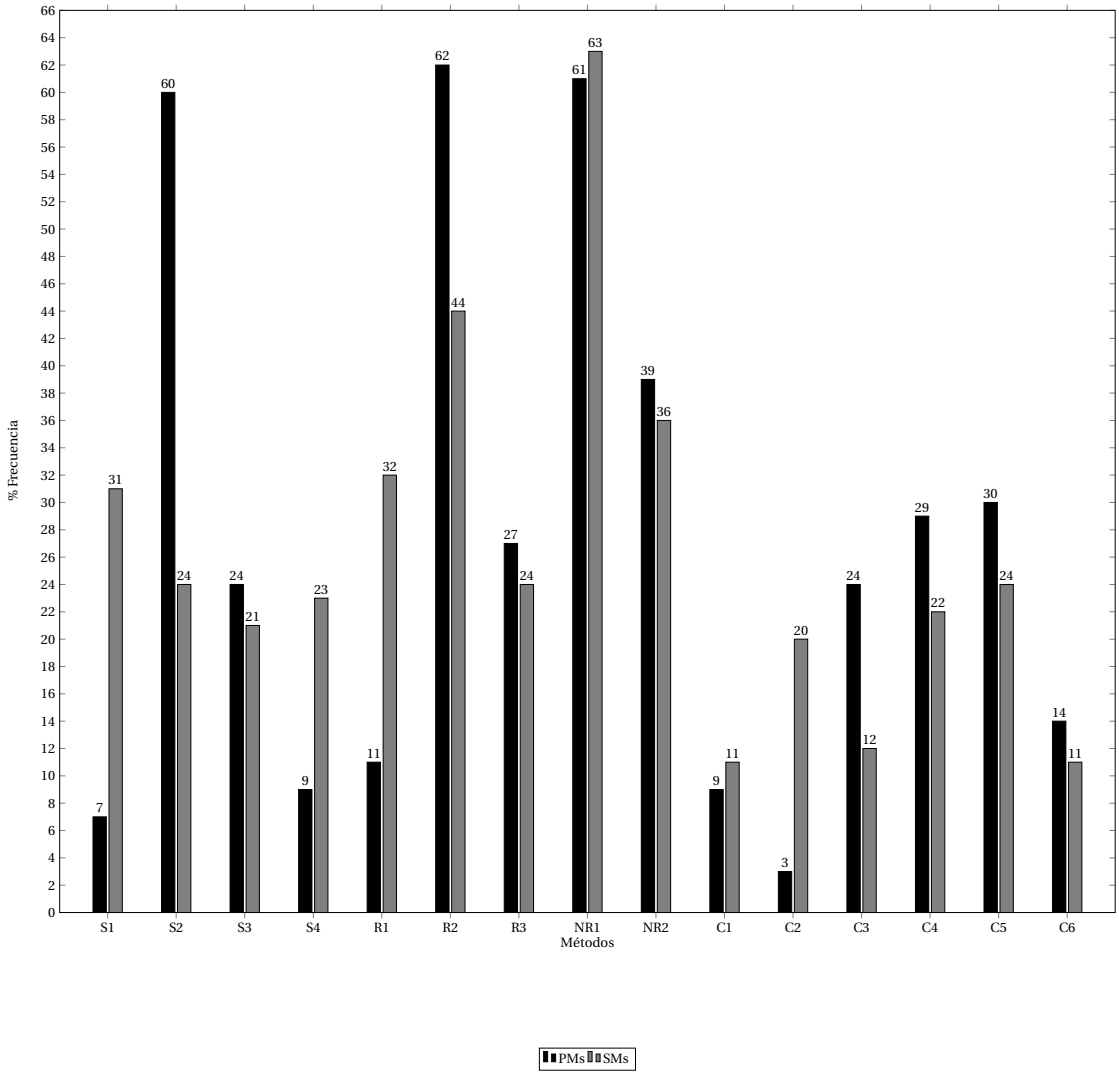


Figura 7.13: Análisis de frecuencia de los métodos que fueron considerados en el proceso de búsqueda de las metaherísticas. Donde S1-S4 representan los métodos de suavizado, R1-R3 corresponde a los métodos de representación de series de tiempo, NR1-NR2 son los métodos de reducción de numerosidad y C1-C6 representan los clasificadores.

tiene la desventaja de tener un costo computacional alto.

- Por otro lado, si la base de datos es dicotómica, el ruido es moderado y su dimensionalidad se encuentra por debajo de un valor aproximado de 350, entonces la variante poblacional P-DEMS1 que se basa en *rand/1/bin* resulta ser la mejor opción, además de lograr resultados competitivos antes de 100 iteraciones.

- Se logró observar que en presencia de un esquema 4/4/2, donde existen cuatro tareas diferentes con un número promedio de cuatro métodos alternativos por tarea, de los cuales se están optimizando por lo menos dos hiper-parámetros con diferentes límites de las variables, la variante *rand/1/bin* aporta una mejor diversidad de modelos completos.

- En relación a los modelos finales, se puede ver que para la mayoría de las bases de datos se encontró un modelo completo que contiene los métodos más simple para las tareas de suavizado, reducción de dimensionalidad y reducción de numerosidad. Estos métodos son Promedio móvil, PAA e INSIGHT respectivamente. Por el lado de la tarea de clasificación, Adaboost fue el método más frecuente.

- Un importante hallazgo fue encontrar diferentes configuraciones de modelos completos con un desempeño similar para una base de datos en particular. Ésto sugiere que algunas bases de datos temporales puedan verse como un problema multi-modal.

8

SELECCIÓN DEL MODELO COMPLETO MULTI-OBJETIVO EN BASES DE DATOS TEMPORALES: OPTIMIZANDO TIEMPO Y RENDIMIENTO

Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

—Isaac Newton

8.1. ANTECEDENTES

En este capítulo se presenta una extensión de la propuesta presentada en los Capítulo 6 y 7, dicha extensión consiste en tratar el problema de FMS desde un enfoque multi-objetivo. Dentro de la revisión de la literatura se pudo observar que existen algunas propuestas multi-objetivo para la solución del problema FMS, tal es el caso del uso de MOEA/D, donde se optimizan el error de clasificación durante el entrenamiento y la complejidad a través de la dimensión de Vapnik-Chervonenkis [147].

Otra propuesta es donde se muestra un enfoque multi-objetivo donde se utilizó el sesgo y la varianza como funciones objetivo para regular la complejidad de los modelos finales [146]. Recientemente, se propuso en la literatura un enfoque multi-objetivo donde se incorpora la reducción de instancias que enfoca la búsqueda en dos objetivos que son la tasa de error del modelo y la tasa de reducción del conjunto de entrenamiento [87, 139].

Sin embargo, se pudo observar que estos enfoques multi-objetivo están dedicados a tratar con bases de datos estáticas, en las cuales las variables (atributos) no son dependientes entre si y en conjunto no poseen una alta dimensionalidad. En la propuesta que se presenta en este capítulo se trabaja con bases de datos temporales que, dada su naturaleza continua y su alta dimensión a nivel columna, resulta costoso evaluar un modelo completo que implique

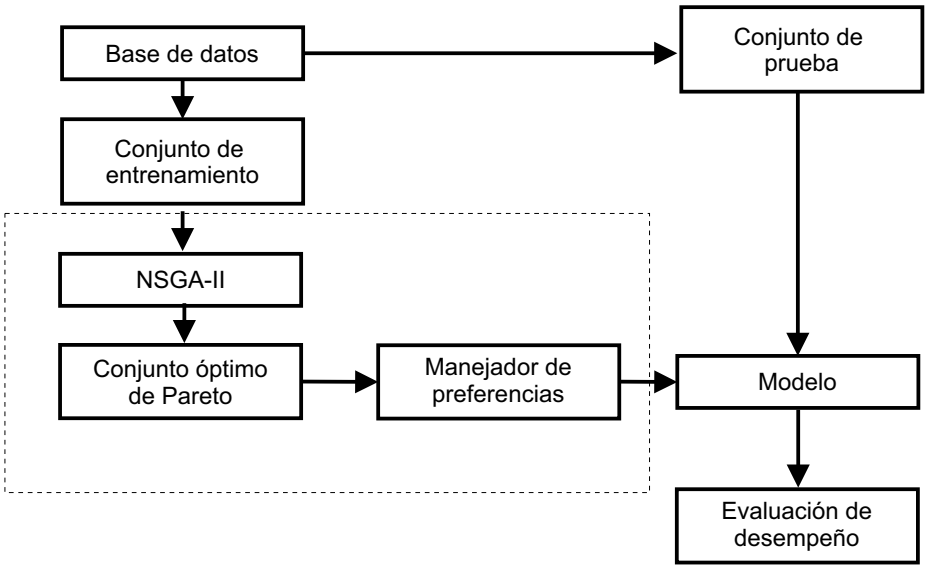


Figura 8.1: Representación gráfica del enfoque multi-objetivo N2SFMS.

múltiples sub-tareas. Por lo tanto, se propone un enfoque multi-objetivo que encuentre soluciones competitivas que posean un compromiso entre un error de clasificación y un tiempo de ejecución bajos.

8.2. DESCRIPCIÓN DE LA PROPUESTA

En esta sección se presenta el enfoque multi-objetivo para resolver el problema FMS en bases de datos temporales, dicha propuesta usa como herramienta de búsqueda al conocido algoritmo NSGA-II [37]. Las funciones objetivo que son consideradas durante la optimización son: a) el error de clasificación y b) el tiempo de ejecución que invierte cada solución sobre un conjunto de datos. En la Figura 8.1 se muestra gráficamente la propuesta multi-objetivo que es denominada como N2FMS.

Las bases de datos temporales que se utilizan para evaluar este enfoque se muestran en la Tabla 8.1. Cada base de datos se encuentra dividida en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento se utiliza durante el proceso de optimización de N2SFMS, donde cada solución que es evaluada representa un modelo completo que contiene una combinación de métodos con sus respectivos hiperparámetros para llevar a cabo las tareas de suavizado, reducción de dimensionalidad, reducción de numerosidad y clasificación de las series de tiempo. En la Tabla 7.1 se presentan los métodos e hiperparámetros que son considerados para sugerir un modelo completo.

Los modelos completos son evaluados en función del error de clasificación y el tiempo de ejecución que invierte cada solución. Cuando finaliza el proceso de optimización, se obtiene como resultado un frente de Pareto formado por un conjunto de soluciones candidatas.

A partir del frente de Pareto se procede a seleccionar una solución del frente a través de una estrategia para el manejo de preferencias. En este trabajo, se prueban cuatro estrategias para seleccionar una solución del frente y serán descritas más adelante. Una vez seleccionada la solución del frente, se procede a evaluar la calidad de la solución, pero esta vez se hace

sobre el conjunto de prueba de la base de datos.

Tabla 8.1: Descripción de las bases de datos usadas en los experimentos del enfoque multi-objetivo.

No.	Nombre	Clases	Entrenamiento	Prueba	Longitud de la serie de tiempo
1.	Beef	5	30	30	470
2.	CBF	3	30	900	128
3.	GunPoint	2	50	150	150
4.	Lightning-2	2	60	61	637
5.	Lightning-7	7	70	73	319
6.	OliveOil	4	30	30	570
7.	Synthetic_control	6	300	300	60
8.	Trace	4	100	100	275

8.2.1. NSGA-II PARA RESOLVER EL PROBLEMA FMS (N2FMS)

NSGA-II es un algoritmo evolutivo multi-objetivo propuesto por Kalyanmoy Deb y sus estudiantes. Es una versión mejorada del NSGA ya que utiliza un operador de apilamiento (crowding) en vez de nichos. A continuación se describe cómo trabaja el algoritmo NSGA-II que fue utilizado como herramienta de búsqueda de modelos completos para bases de datos temporales.

1. ***Población inicial***. Se genera aleatoriamente una población de individuos P_0 de tamaño N. Cada solución fue codificada de acuerdo a la forma presentada en el Capitulo 7 a través de la Ecuación 7.1, que corresponde a una codificación real.

2. ***Evaluar funciones objetivo***. Por cada individuo de la población se calculan los valores de aptitud de cada función objetivo. Para el caso de las soluciones que representan modelos completos, se consideró el error de clasificación con validación cruzada f_1 y el tiempo de ejecución acumulado f_2 que requirió cada solución para ser evaluada.

   La Ecuación 8.1 expresa el primer objetivo f_1, donde a representa la porción de instancias que fueron incorrectamente clasificadas y b el total de instancias de un determinado conjunto de datos.

$$f_1 = CVER = \frac{1}{k}\sum_{i=1}^{k}(\frac{a}{b})_i \tag{8.1}$$

   Por otra parte, la Ecuación 8.2 muestra cómo se calcula el segundo objetivo f_2, donde ts_i representa el tiempo de ejecución ocupado por el método de suavizado sugerido en el modelo, tr_i es el tiempo de ejecución ocupado por el método de representación, tnr_i el tiempo empleado para la reducción de numerosidad y tc_i el tiempo que se emplea para la tarea de clasificación.

$$f_2 = RT = \frac{1}{k}\sum_{i=1}^{k}(ts_i + tr_i + tnr_i + tc_i) \tag{8.2}$$

   En ambas funciones se ocupa el valor de k que representa el número de particiones generadas por la validación cruzada sobre el conjunto de entrenamiento.

3. ***Ordenamiento no-dominado***. Esta tarea consiste en ordenar la población con base en un rango de dominancia, donde los valores de las funciones objetivo de cada individuo son comparados contra los valores de los demás individuos de la población. Posteriormente los individuos son clasificados en diferentes frentes de acuerdo a sus rangos de dominancia.

4. ***Ordenamiento basado en distancia de crowding***. Después del ordenamiento basado en la dominancia, a cada individuo se le asigna un valor de distancia de crowding. Esta medida se encarga de medir la densidad de soluciones dentro de un vecindario en el espacio de las funciones objetivo, es decir, mide el espacio alrededor de una solución i que no está ocupado por otra solución en la población. Con el ordenamiento basado en esta medida se asegura que la población mantenga una buena diversidad de soluciones en el espacio de las funciones objetivo. La distancia de crowding d para cada solución i es calculada a través de la Ecuación 8.3.

$$d_i = \sum_{m=1}^{M} \left| \frac{f_m^{(I_{i+1}^m)} - f_m^{(I_{i-1}^m)}}{f_m^{max} - f_m^{min}} \right| \tag{8.3}$$

   Donde I^m representa un vector que indica la alternativa de solución vecina a la solución i, f_m^{max} y f_m^{min} son los valores máximos y mínimos sobre todo el espacio de la función objetivo m y M representa el número de funciones objetivo optimizadas.

5. ***Selección, cruza y mutación***. Para generar la próxima generación, son necesarias las operaciones de selección, cruza y mutación. Cada solución de la población cuenta con un rango de dominancia (r_i) asociado y una distancia de crowding (d_i). Para el proceso de selección se llevan a cabo torneos binarios donde participan N parejas que son generadas aleatoriamente de la población actual.

   Los ganadores de cada torneo son aquellas soluciones que tengan un r_i de mejor calidad. Si ambas soluciones tienen el mismo rango, entonces se preferirá la solución que porte un mayor grado de diversidad a la población en construcción, es decir, la que tenga una distancia d_i alta. Los vencedores de cada torneo son los únicos que podrán generar descendencia mediante los operadores de cruza y mutación.

   El NSGA-II clásico usa el operador de cruza binaria simulada (SBX, *Simulated Binary Crossover*) que simula la cruza binaria de un solo punto pero para espacios continuos. SBX genera dos nuevos descendientes c_1 y c_2 a partir de dos padres p_1 y p_2, cada descendiente es generado mediante la Ecuación 8.4.

$$\begin{cases} c_1 = 0{,}5[(1+\beta)p_1 + (1-\beta)p_2] \\ c_2 = 0{,}5[(1-\beta)p_1 + (1+\beta)p_2] \end{cases} \tag{8.4}$$

   Donde β se conoce como factor de dispersión, el cual puede tomar valores menores o igual a uno (si β es igual a uno los descendientes son una copia de los padres). β usa la distribución de probabilidad polinomial expresada en la Ecuación 8.5, donde η_n es el índice de la distribución que toma valores positivos, si los valores que toma

son cercanos a cero, entonces los descendientes serán menos parecidos a sus padres, mientras que con valores altos, los descendientes serán más parecidos a los padres.

$$\beta = \begin{cases} (2 \times rand)^{\frac{1}{1+\eta_c}} & \text{si } rand \leq 0{,}50 \\ \frac{1}{2-2\times rand}^{\frac{1}{1+\eta_c}} & \text{de otro modo} \end{cases} \tag{8.5}$$

El operador de mutación usa la distribución de probabilidad polinomial para perturbar una solución que formará parte de una nueva población [34]. Una solución mutada xm_i se genera a partir de la Ecuación 8.6, donde x_i representa el padre seleccionado. x_i^U y x_i^L representan el límite superior e inferior de la variable de decisión, respectivamente. $(x_i^U - x_i^L)$ es el valor máximo de perturbación permitido entre el padre y la solución mutada, y α_i es el factor de perturbación que puede ser calculado a través de la Ecuación 8.7.

$$xm_i = x_i + (x_i^U - x_i^L) \times \alpha_i \tag{8.6}$$

$$\alpha_i = \begin{cases} (2 \times rand)^{\frac{1}{1+\eta_m}-1} & \text{si } rand < 0{,}50 \\ 1 - (2 - 2 \times rand)^{\frac{1}{1+\eta_m}} & \text{si } rand \geq 0{,}50 \end{cases} \tag{8.7}$$

Donde $rand$ es un número aleatorio uniformemente distribuido entre $[0,1]$ y η_m representa el índice de distribución de mutación.

6. ***Reemplazo***. El reemplazo se realiza en una población intermedia que se forma de la unión de la población actual P_g y la población de descendientes Q_g. NSGA-II realiza un reemplazo elitista para seleccionar a los individuos que formarán la siguiente generación.

El Algoritmo 10 presenta de forma sintetizada los pasos que sigue el algoritmo NSGA-II, en este caso para encontrar soluciones candidatas que representan modelos completos para bases de datos temporales.

Algoritmo 10 N2FMS

Entrada: G (Número de generaciones), N (Tamaño de la población)

1: Generar aleatoriamente la población inicial de modelos completos P_0 mediante la codificación expresada en la Ecuación 7.1
2: Evaluar las funciones objetivo Ecuación 8.1 y Ecuación 8.2
3: P_0 = ordenamiento-no-dominado(P_0)
4: P_0 = distancia-crowding(P_0)
5: **for** g = 1 to G **do**
6: Crear población de descendientes Q_g de tamaño N a través de la selección por torneo binario, el operador de cruza SBX (Ecuaciones 8.4 y 8.5) y la mutación polinomial (Ecuaciones 8.6 y 8.7)
7: Evaluar funciones objetivo
8: Combinar población de padres y descendientes $R_g = P_g \cup Q_g$
9: R_g = ordenamiento-no-dominado(R_g)
10: P_g = reemplazo(R_g)
11: **end for**

8.2.2. ESTRATEGIAS PARA EL MANEJO DE PREFERENCIAS

Al finalizar el proceso de optimización multi-objetivo se tiene un conjunto de soluciones de las cuales se puede hacer una selección de la solución que mejor compromiso tenga entre los objetivos involucrados. Generalmente, la selección de la solución más satisfactoria se decide por el método de ponderación, donde se define una función de costos en términos de promedio ponderado en la que cada objetivo posee un peso basado en su importancia relativa. Sin embargo, puede resultar difícil establecer el peso adecuado a cada objetivo [136].

Por lo tanto, se han propuesto estrategias para facilitar la tarea de seleccionar una solución a partir de un conjunto de soluciones óptimas. Una de las estrategias mayormente utilizadas es la propuesta por Das [33], que propone seleccionar la solución que se encuentre a una mayor (problemas de minimización) distancia del frente de Pareto hacia una línea que conecta los dos extremos del frente.

Otra estrategia es establecer un punto de referencia [36, 184], el cual servirá para seleccionar la solución del frente que se encuentre más cerca de dicho punto de referencia. Bajo esta estrategia se proponen las siguientes opciones para seleccionar una solución del conjunto de soluciones que representan modelos completos para bases de datos temporales.

S-N2FMS (*Single-NSFMS*)

En esta estrategia, la selección de la solución se basa en la distancia mínima hacia un punto de referencia z_k^*, donde k representa el número de objetivos optimizados. En este caso, el punto de referencia es el punto (0,0), en el cual se establece el punto ideal para aquellos modelos completos donde el error de clasificación tienda a cero y el tiempo de ejecución sea lo más bajo posible. La medida de distancia que se utiliza es la distancia Euclidiana (ver Ecuación 3.9). En la Figura 8.2(a) se muestra un ejemplo de cómo funciona esta estrategia para seleccionar una solución del frente.

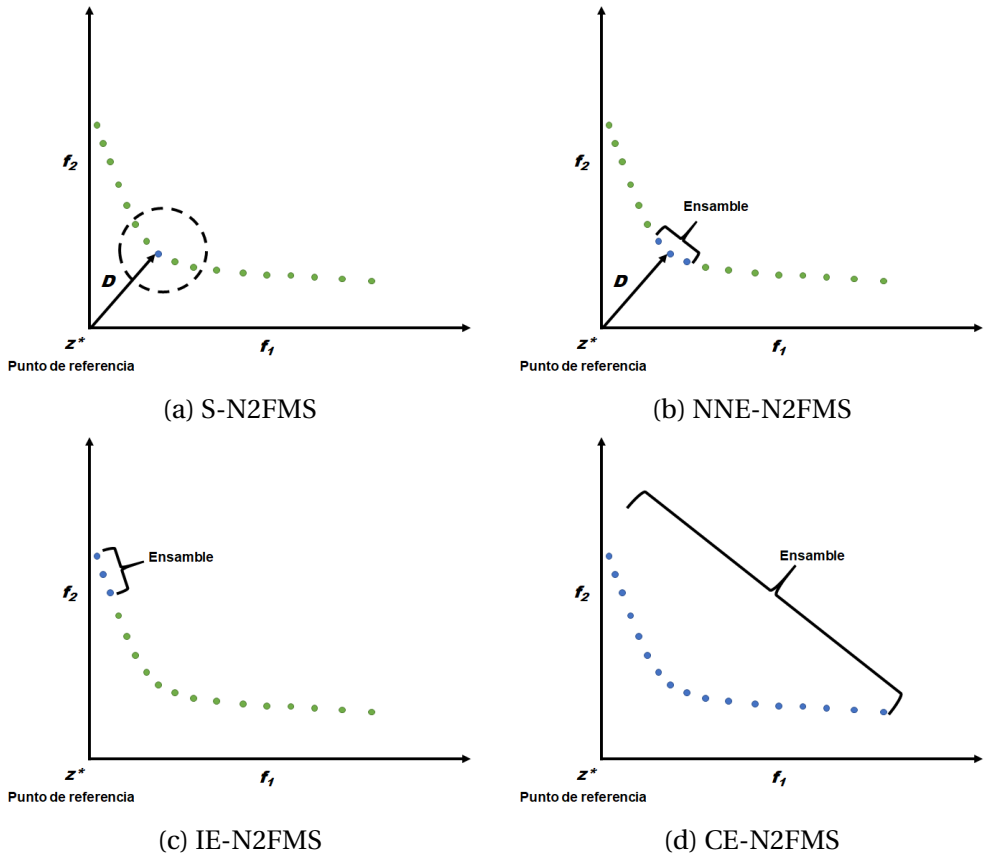


Figura 8.2: Representación gráfica de estrategias para el manejo de preferencias.

NNE-N2FMS (*Near Neighbors Ensemble-N2FMS*)

Para esta estrategia, se construye un ensamble basado en mayoría de votos [150]. Dicho ensamble es formado a partir de la solución más cercana al punto de referencia z_k^* y de sus dos vecinos más cercanos. En la Figura 8.2(b) se muestra un ejemplo de cómo funciona esta estrategia.

IE-N2FMS (*Increasing Ensemble-N2FMS*)

Esta estrategia ha sido denominada como ensamble incremental, debido a que se seleccionan las tres primeras soluciones, de menor a mayor considerando el valor que tengan en la función objetivo f_1. Para ésto, todas las soluciones que se encuentren en el frente son ordenadas de menor a mayor otorgando mayor preferencia al error de clasificación. En la Figura 8.2(c) se muestra un ejemplo de las soluciones seleccionadas de acuerdo a esta estrategia.

CE-N2FMS (*Complete Ensemble-N2FMS*)

Esta estrategia se basa en la construcción de un ensamble a partir de todas las soluciones disponibles en el frente de Pareto. La Figura 8.2(d) muestra un ejemplo representativo de esta estrategia.

Tabla 8.3: Configuración de parámetros usada por N2FMS. N es el tamaño de la población, G el número de generaciones, CR el porcentaje de cruza y MR el porcentaje de mutación.

Parámetro	Valor
N	20
G	150
CR	0.9
MR	0.1

8.3. EXPERIMENTOS Y RESULTADOS

Los experimentos de la propuesta multi-objetivo N2FMS están divididos en tres partes: (1) la comparación de los resultados numéricos de las cuatro estrategias para el manejo de preferencias que fueron descritas anteriormente, (2) la comparación entre el enfoque multi-objetivo y mono-objetivo y (3) el análisis de los frentes acumulados.

Para los experimentos se utilizaron las bases de datos descritas en la Tabla 8.1. La configuración de parámetros usada por el algoritmo N2FM es presentada en la Tabla 8.3. El algoritmo N2FM es una adaptación del algoritmo NSGA-II que fué implementado en el lenguaje MATLAB por Seshadri [158].

Todos los experimentos se llevaron a cabo en un equipo de cómputo con la configuración que se muestra en la Tabla 8.2.

Tabla 8.2: Configuración del equipo de cómputo utilizado en cada experimento.

Sistema Operativo	Windows 7 Pro
CPU	Intel® Xeon® E3-1200 v3
RAM	16 GB

8.3.1. EXPERIMENTO 1. COMPARACIÓN ENTRE ESTRATEGIAS PARA MANEJO DE PREFERENCIAS

Se realizaron cinco ejecuciones independientes del algoritmo N2FMS por cada base de datos, donde la condición de paro fue un máximo de 3000 evaluaciones o 150 generaciones. Por cada ejecución se obtuvo un frente de Pareto, del cual se seleccionó una solución a través de las cuatro estrategias de manejo de preferencias. Una vez que se selecciona una solución del frente, ésta es evaluada en el conjunto de prueba para medir su rendimiento. En la Tabla 8.4 se muestran los resultados numéricos obtenidos por las cuatro estrategias. Sobre el conjunto de resultados numéricos se realizó la prueba estadística no paramétrica de Krukal-Wallis con un 95 % de confianza. Derivado de esta prueba se muestran los *p-valor* por cada base de datos, que ayudan a evidenciar que solo en dos (Beef y CBF) de las ocho bases de datos no se presentó diferencia significativa entre las cuatros estrategias. Sin embargo, en las bases de datos Gun-Point, Lightning-2, Lightning-7, OliveOil y Trace presentan la tasa de error más baja a través de la estrategia S-N2FMS.

Se presentó una excepción en el caso de la base de datos Synthetic Control que muestra una tasa de error significativamente más baja con la estrategia NNE-N2FMS. Sin embargo, la estrategia que reporta mejores resultados es S-N2FMS.

Para reforzar la validación estadística, se realizó la prueba post-hoc de Bonferroni en la que se hizo uso de los mejores valores para comparar el rendimiento de las cuatro estrategias considerando un comportamiento generalizado sobre las ochos bases de datos, ver Figura 8.3. La Figura 8.3 muestra gráficamente los resultados de esta prueba estadística, donde el eje x muestra el rango medio del intervalo de confianza y en el eje y se presenta el nombre de cada una de las estrategias comparadas. Se puede ver claramente que la estrategia S-N2FMS presenta un mejor comportamiento generalizable al momento de seleccionar una solución del frente.

Tabla 8.4: Resultados numéricos del modelo construido a partir de las estrategias de N2FMS para cada base de datos. Los valores entre paréntesis representan la desviación estándar y los valores en **negritas** significa el mejor valor encontrado.

Base de datos	S-N2FMS	NNE-N2FMS	DE-N2FMS	CE-N2FMS
Beef	0.3333 (±0.20)	0.3333 (±0.10)	0.2200 (±0.14)	0.2733 (±0.08)
	p-valor: 1.07E-01			
CBF	0.1369 (±0.31)	0.2267 (±0.32)	0.1769 (±0.24)	0.2573 (±0.33)
	p-valor: 1.81E-01			
GunPoint	**0.1347 (±0.20)**	0.5320 (±0.08)	0.5333 (±0.07)	0.5227 (±0.08)
	p-valor: 6.34E-07			
Lightning-2	**0.0918 (±0.21)**	0.5018 (±0.01)	0.5013 (±0.01)	0.5013 (±0.03)
	p-valor: 1.23E-08			
Lightning-7	**0.0928 (±0.01)**	0.7621 (±0.02)	0.7621 (±0.02)	0.7650 (±0.24)
	p-valor: 3.26E-10			
OliveOil	**0.1000 (±0.11)**	0.1200 (±0.10)	0.2333 (±0.22)	0.2800 (±0.42)
	p-valor: 7.82E-03			
Synthetic Control	0.0480 (±0.07)	**0.0313 (±0.07)**	0.5307 (±0.42)	0.5313 (±0.22)
	p-valor: 1.17E-09			
Trace	**0.0160 (±0.04)**	0.5480 (±0.16)	0.6220 (±0.17)	0.6580 (±0.20)
	p-valor: 5.49E-13			

8.3.2. EXPERIMENTO 2. COMPARACIÓN ENTRE S-N2FMS Y μ-DEMS

A partir de los resultados obtenidos en el experimento anterior, se pudo observar que S-N2FMS fue la mejor estrategia para construir un modelo a partir de la selección de una solución del frente de Pareto. Por lo tanto, en esta parte de los experimentos se presenta una comparación entre S-N2FMS y μ-DEMS en su variante *rand/1/bin*, que fue la propuesta mono-objetivo que obtuvo los valores más bajos en error de clasificación para la mayoría de las bases de datos.

En este experimento, los algoritmos fueron evaluados bajo condiciones similares como el lenguaje de programación, equipo de cómputo, equipo de cómputo y número de evaluaciones. En la Tabla 8.5 se presentan los resultados obtenidos, donde CVER corresponde al valor promedio del error de clasificación de las cinco ejecuciones realizadas por cada base de datos, mientras que RT representa el promedio del tiempo de ejecución consumido por cada solución final. Como parte de las pruebas estadísticas de este experimento, se aplicó la prueba estadística no paramétrica de Wilcoxon, sobre las muestras pareadas de los valores de CVER, con un 95 % de confianza.

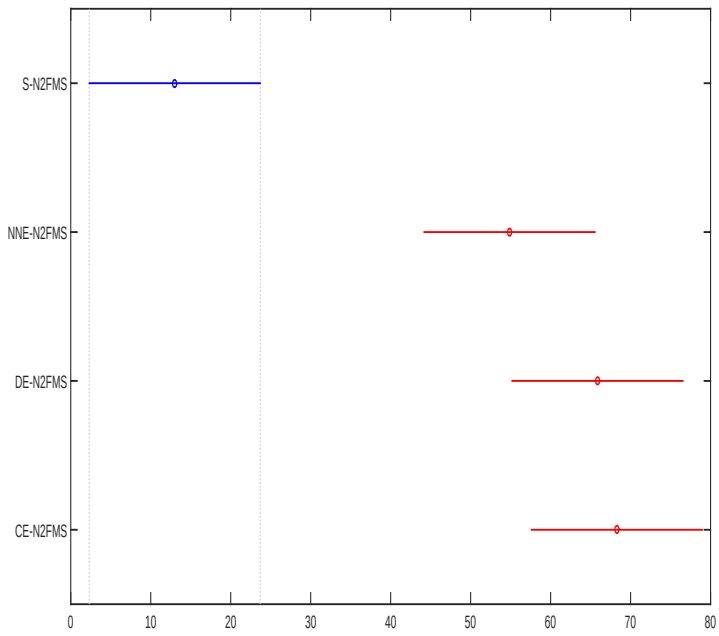


Figura 8.3: Prueba Post-hoc de Bonferroni, usando los mejores valores del conjunto de base de datos.

De acuerdo a la prueba estadística, S-N2FMS fue capaz de superar a μ-DEMS con una diferencia estadísticamente significativa en el valor de CVER solo en dos bases de datos (Lightning-7 y Trace). Por otro lado, $\mu - DEMS$ tuvo un mejor comportamiento, estadísticamente significativo, en las bases de datos Beef, CBF, GunPoint, Lightning-2, OliveOil y Synthetic control.

Respecto a RT, es muy notable que la mayoría de los valores reportados por S-N2FMS fueron considerablemente más pequeños, donde el valor más alto fue de 2.64 segundos. Mientras que en μ-DEMS, la mayoría de los valores de RT exceden los 3 segundos, siendo 1550.54 segundos el valor más alto en cuanto a tiempo de ejecución invertido por la mejor solución para una de las bases de datos.

Con este experimento se pudo observar que con la versión multi-objetivo, S-N2FMS, es posible encontrar soluciones con un costo computacional más bajo, es decir con un consumo de tiempo menor sin sacrificar tanto la calidad de la solución. Por otro lado, si se prefiere una solución más exacta sin importar el costo computacional, la versión mono-objetivo resulta ser la mejor opción.

8.3.3. EXPERIMENTO 3. ANÁLISIS DE LOS FRENTES ACUMULADOS DE PARETO

En esta parte de los experimentos se muestran los frentes acumulados de Pareto obtenidos de N2FMS por cada una de las bases de datos, ver de la Figura 8.4 a la Figura 8.7. Cada frente acumulado es resultado de los frentes obtenidos de cada ejecución independiente de N2FMS. Cada solución del frente acumulado es re-evaluada en el conjunto de prueba. A partir de los frentes acumulados se puede observar que N2FMS es capaz de proveer de so-

Tabla 8.5: Comparación entre S-N2FMS y μ-DEMS. Los valores entre paréntesis representan la desviación estándar y los valores en **negrita** representan los mejores valores encontrados.

Bases de datos	**S-N2FMS**			**μ-DEMS**[126]	
	CVER	RT		CVER	RT
Beef	0.3333 (±0.20)	**0.20**		**0.0667 (±0.07)**	7.40
CBF	0.1369 (±0.31)	**2.64**		**0.0000 (±0.00)**	1550.54
GunPoint	0.0827 (±0.12)	**0.10**		**0.0000 (±0.00)**	16.11
Lightning-2	0.0918 (±0.21)	**0.09**		**0.0000 (±0.00)**	10.47
Lightning-7	**0.0928 (±0.18)**	**0.19**		0.7587 (±0.03)	4.19
OliveOil	0.1000 (±0.11)	**0.08**		**0.0067 (±0.01)**	8.43
Synthetic_control	0.0480 (±0.07)	**0.30**		**0.0000 (±0.00)**	3.15
Trace	**0.0160 (±0.04)**	**0.18**		0.8000 (±0.45)	6.14

luciones baratas con un desempeño competitivo. Se puede observar que durante la etapa de entrenamiento los resultados pueden llegar a ser altamente optimistas y al momento de validar dichas soluciones en el conjunto de prueba se logra ver el comportamiento real sin sacrificar la calidad de las soluciones. Por lo tanto, N2FMS representa una opción competitiva para buscar modelos completos para bases de datos temporales que cumplan con el balance entre un desempeño competitivo y un bajo costo computacional.

8.4. CONCLUSIONES

En este capítulo se presentó un enfoque multi-objetivo, llamado N2FMS, para tratar con el problema de FMS en bases de datos temporales. El error de clasificación y la complejidad del modelo (en el contexto de costo computacional), fueron los dos objetivos optimizados. Se evaluaron ocho bases de datos temporales con diferentes características de dimensión, número de instancias y número de clases. Los experimentos realizados en este capítulo se integraron en tres partes que fueron: el análisis estadístico de los valores numéricos entre cuatro estrategias para la selección de una única solución del frente de Pareto, la comparación entre la mejor estrategias de selección contra la variante mono-objetivo basada en *rand/1/bin* y el análisis de los frentes acumulados. A partir de este conjunto de experimentos se presentan a continuación los hallazgos más relevantes:

- S-N2FM fue la mejor estrategia para seleccionar un modelo completo del conjunto de soluciones que forman el frente de Pareto. Por lo tanto, la solución que se sitúa en la rodilla del frente resulta ser el modelo más competitivo.
- A partir de la comparación entre enfoque multi-objetivo y el enfoque mono-objetivo, se recomienda que, si se desea encontrar una solución prometedora con un costo computacional bajo, se use el enfoque multi-objetivo N2FMS con la estrategia S-N2FMS. Por otro lado, si se desea una alta precisión en el desempeño del modelo sin importar el costo computacional, entonces se utilice el enfoque mono-objetivo basado en la variante *rand/1/bin*.
- Se pudo observar que el enfoque multi-objetivo es mejor y menos sensible que el enfoque mono-objetivo cuando trata con bases de datos que poseen clases no separables

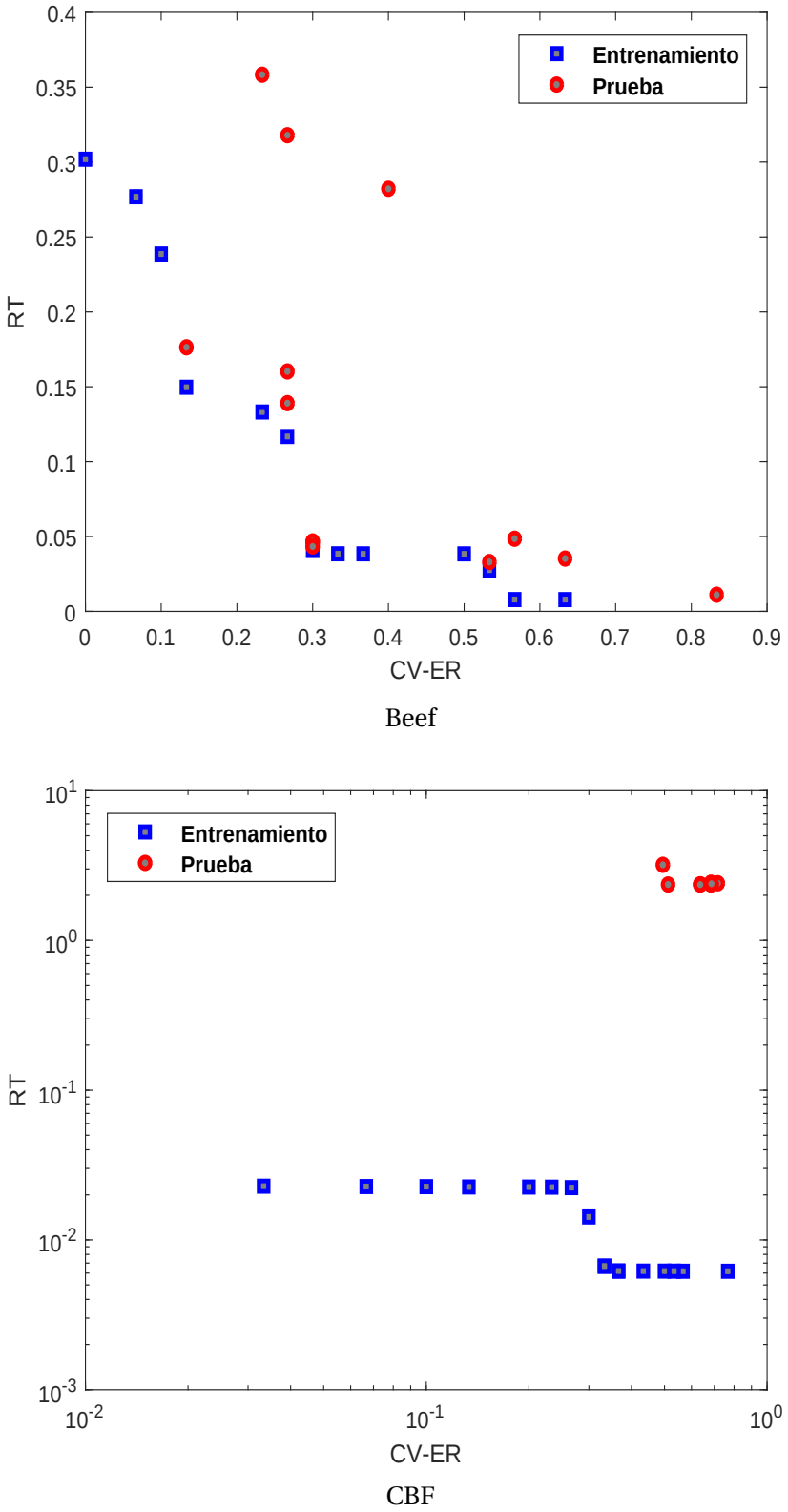
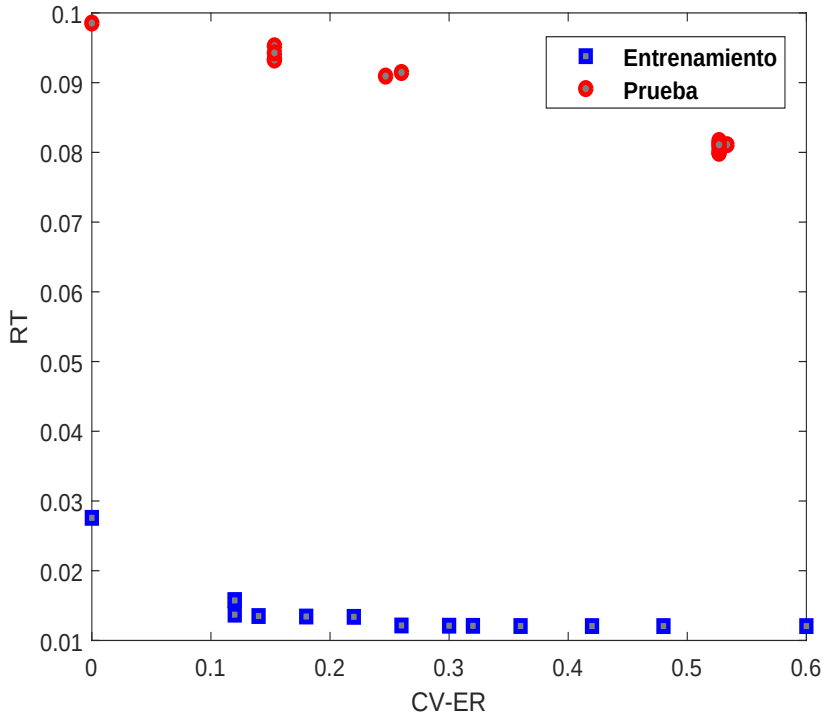
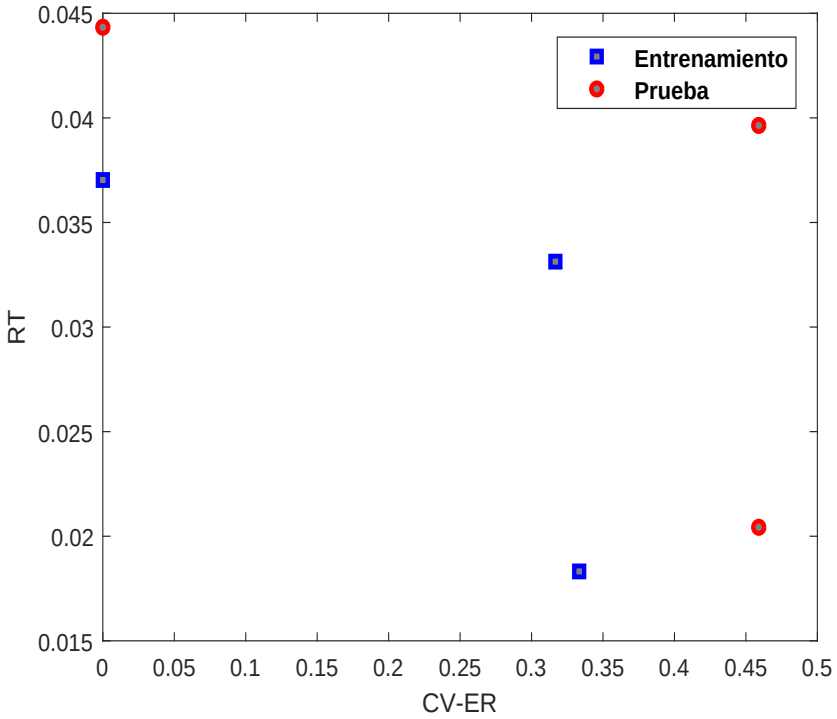


Figura 8.4: Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos Beef y CBF.

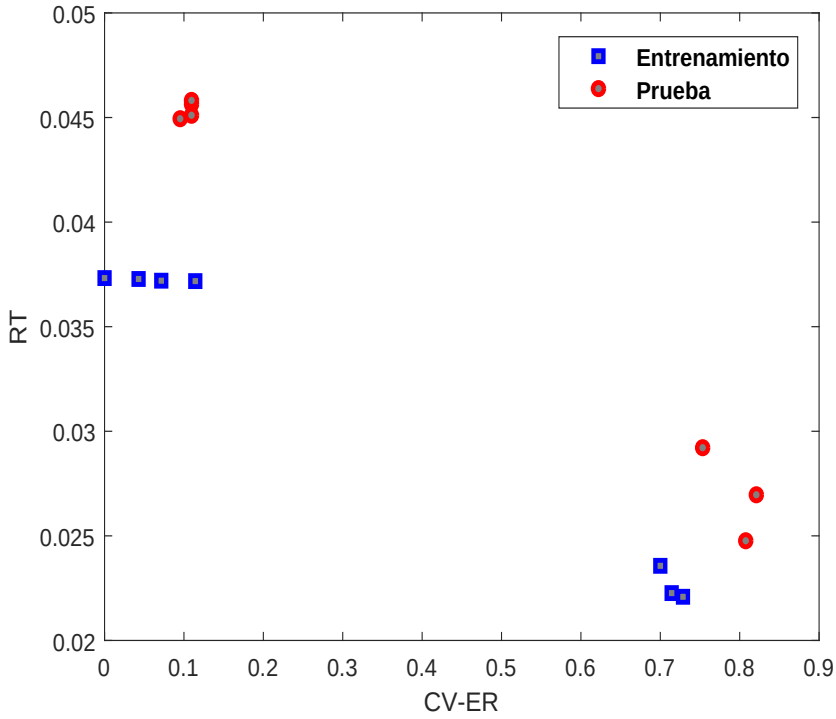


GunPoint

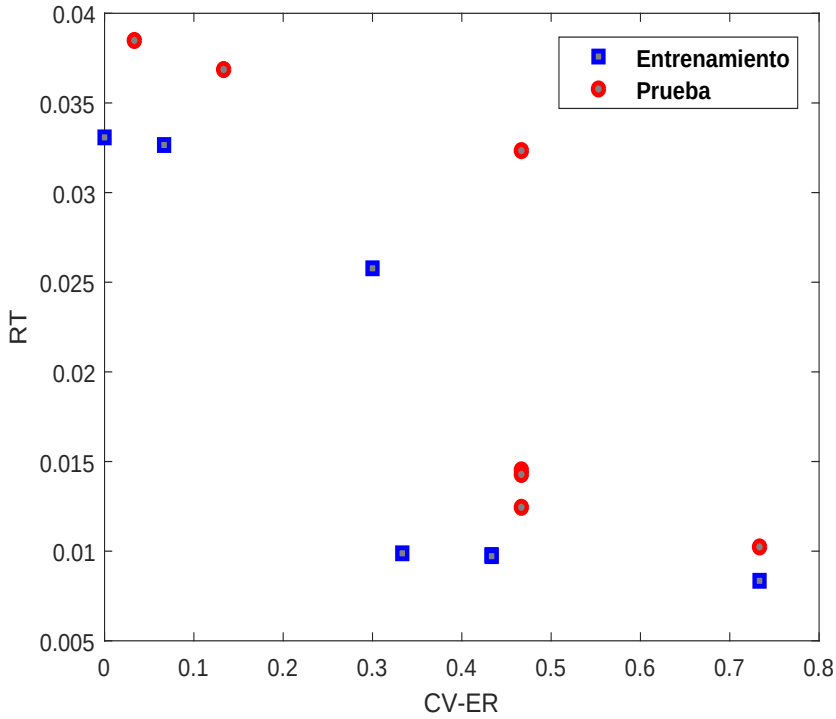


Lightning-2

Figura 8.5: Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos GunPoint y Lightning-2.

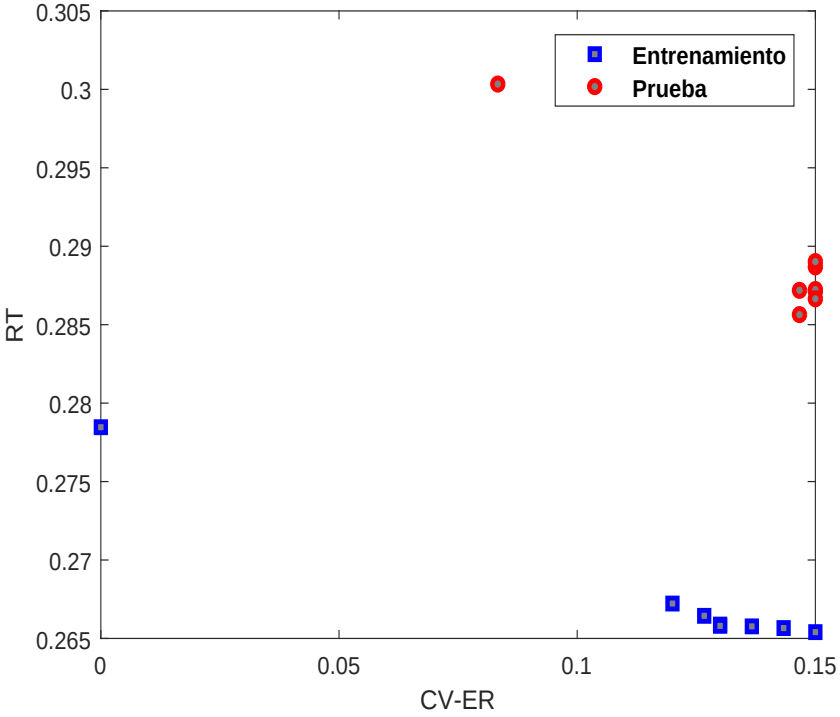


Lightning-7

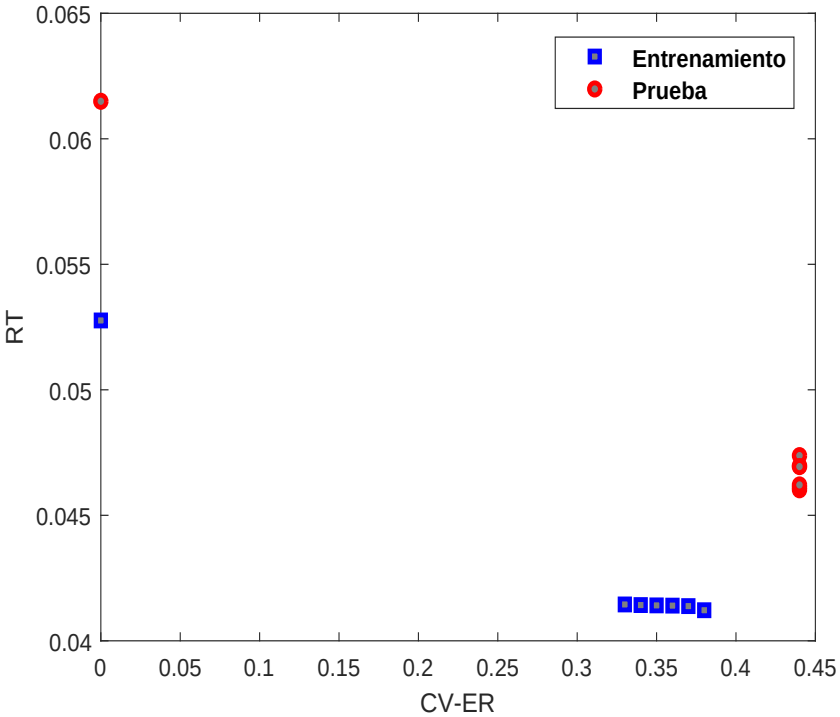


OliveOil

Figura 8.6: Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos Lightning-2 y OliveOil.



Synthetic_control



Trace

Figura 8.7: Comparación entre el frente acumulado (cuadros azules), obtenido a partir de las cinco ejecuciones independientes con el conjunto de entrenamiento, y soluciones evaluadas en el conjunto de prueba (círculos rojos), para las bases de datos Synthetic_control y Trace.

(las series de tiempo son muy parecidas entre las clases establecidas).

- El enfoque multi-objetivo representa una propuesta prometedora para resolver el problema de FMS en bases de datos temporales para encontrar soluciones con un costo computacional bajo sin sacrificar el desempeño de éstas.

IV

Conclusiones finales

9

CONCLUSIONES Y TRABAJO FUTURO

En la vida no existe nada que temer, solo cosas que comprender.

—Marie Curie

9.1. CONCLUSIONES

El desarrollo de esta tesis se enfocó en el estudio y tratamiento del problema de selección de modelo completo en bases de datos temporales a través del uso de metaheurísticas como herramienta de búsqueda.

El desarrollo de esta investigación se resume a un conjunto de etapas que fueron: (1) revisión y actualización de la literatura relacionada al problema de selección de modelo completo, (2) revisión de las principales tareas de minería de datos temporal así como los métodos más usados y (3) revisión de las alternativas de búsqueda para aprendizaje computacional. A partir de estas etapas se propone un marco de referencia para resolver el problema de selección de modelo completo en bases de datos temporales.

Este marco de referencia plantea la búsqueda de modelos completos bajo un esquema que incluya una combinación adecuada de métodos y la optimización de hiperparámetros para llevar a cabo las siguientes tareas: (a) suavizado de ruido en series de tiempo, (b) reducción de dimensionalidad o cambio de representación de series de tiempo, (c) reducción de instancias y (d) clasificación de series de tiempo.

Para la búsqueda de modelos completos bajo el marco de referencia antes mencionado, se propusieron y analizaron tres enfoques, de los cuales describen los principales hallazgos:

En el Capítulo 6, se presentó μ-DEMS, que es la versión micro del conocido algoritmo de Evolución Diferencial. En esta primera propuesta el marco de referencia para la búsqueda de modelos completos solo contaba con dos métodos de suavizado, tres métodos de reducción de dimensión y con el clasificador K-NN con diferentes medidas de distancia. Se realizó un estudio empírico donde se evaluaron cuatro variantes del algoritmo y se analizó el efecto del operador de cruza y el tipo del vector base usado en el operador de mutación.

A partir de este primer enfoque, con un número limitado de métodos para construir modelos completos, la variante de $\mu-$DEMS que usa el mejor vector de la población como vector base junto con la cruza binaria (*best/1/bin*) logra encontrar mejores modelos con un mayor número de evaluaciones. Por otro lado, si se desea obtener una solución competitiva con menos evaluaciones, la mejor opción es usar la variante de μ-DEMS que usa como vector aleatorio como base con una cruza binaria.

Se puede concluir que μ-DEMS es una opción viable como herramienta para la selección de modelos completos en bases de datos temporales. A pesar de que el algoritmo trabaja con una población pequeña y suele llevar a cabo con mayor intensidad la explotación, se logró encontrar una buena configuración para conseguir soluciones prometedoras en generaciones tempranas del algoritmo.

En el Capítulo 7 se realizó una comparación entre las variantes presentadas en el Capítulo 6 y dos enfoques que se basaron en una búsqueda a partir de una única solución. El estudio se enfocó en analizar el comportamiento de las variantes poblacionales (identificadas como P-DEMS) y dos variantes basadas en una solución (identificadas como S-LSMS) para tratar con el problema de selección de modelo completo en bases de datos temporales. El marco de referencia para la construcción de modelos se basó en cuatro métodos de suavizado, tres métodos para la reducción de dimensionalidad, dos métodos para la reducción de la numerosidad y seis clasificadores. En el caso de las variantes poblacionales, se realizó un estudio de la diversidad de las soluciones y en el caso de las variantes basadas en una solución se analizó el uso de diferentes codificaciones para la solución (mixta y binaria). Con este estudio se establecen las siguientes conclusiones:

- La variante basada en una solución con representación binaria es la mejor opción como herramienta de búsqueda de modelos completos para bases de datos que tienen más de dos clases, las series de tiempo y la dimensionalidad se sitúa por encima de 350 valores. Sin embargo, la convergencia de dicho enfoque es más lenta.

- Por otro lado, para obtener soluciones rápidas sin sacrificar el desempeño, se sugiere el uso de la variante poblacional que se basa en rand/1/bin, ya que fue la variante que reportó una mejor diversidad en las soluciones.

- Finalmente con este estudio, se logró observar que algunas bases de datos pueden verse como un problema multi-modal, ya que se lograron encontrar diferentes modelos con un desempeño similar para una determinada base de datos.

Por último, en el Capítulo 8 se presentó N2FMS, que es el enfoque multi-objetivo para tratar con el problema de selección de modelo completo en bases de datos temporales que funciona bajo la estructura del conocido algoritmo multi-objetivo NSGA-II. El desempeño de las soluciones se midió a través del error de clasificación y el consumo de tiempo (de ejecución), los cuales fueron los dos objetivos optimizados. Se diseñaron cuatro estrategias para seleccionar una solución del conjunto de soluciones distribuidas en el frente de Pareto proporcionado por N2FMS. La mejor estrategia fue S-N2FM que se basó en seleccionar la solución más cercana a un punto de referencia ideal en términos de clasificación y tiempo de ejecución de un modelo completo. Se pudo observar que las estrategias que construían un ensamble de soluciones reportaron los peores resultados. Se realizó una comparación del

enfoque multi-objetivo y mono-objetivo para la selección del modelo en bases de datos temporales y se puede concluir que el enfoque multi-objetivo ayuda a conseguir modelos baratos en cuanto a costo computacional sin sacrificar el desempeño y es más robusto cuando trata con bases de datos que poseen clases no separables en las que las instancias son muy parecidas entre si dificultando la clasificación. Por otro lado, si se desea una mejor exactitud en el desempeño de los modelos sin importar el costo computacional, se sugiere el uso del enfoque mono-objetivo.

Uno de los grandes desafíos de la propuesta es la comparación contra otros enfoques del estado del arte. De acuerdo con Shang et al. [159], se ha propuesto una gran variedad de alternativas de solución para automatizar el proceso de aprender a aprender[1]. Estas alternativas han sido diseñadas para solucionar problemas que involucran el ajuste de hiperparámetros y la selección adecuada de cierta familia de métodos para procesar datos de un determinado tipo. Por lo tanto, la comparación justa entre todas estas alternativas representa un problema que debe ser atendido y de cierta manera democratizado. Una de las alternativas a mediano plazo, ante dicho problema, es que las nuevas propuestas de auto aprendizaje acuerden el tipo de tareas que se deben de considerar. Shang et al. [159] sugieren un conjunto de operaciones que deben ser consideradas como son: la limpieza de datos, la ingeniería de características y potencialmente el aumento de datos, la transferencia de aprendizaje y la interacción de usuarios externos.

Se puede concluir de forma general que el enfoque poblacional μ-DEMS logra satisfacer la hipótesis planteada en este trabajo de investigación. Se observó de forma empírica mediante los resultados numéricos que usar la variante *rand/1/bin* proporciona un rendimiento competitivo al momento de seleccionar un modelo completo para bases de datos de series de tiempo. Por otro lado, si se desean resultados más exactos en cuanto a desempeño sin importar el costo computacional, se puede optar por un enfoque basado en una metaheurística basada en un solo punto de búsqueda con una representación binaria. También se cumplió con el objetivo de proporcionar estrategias de selección de preferencias para elegir una solución a partir de un conjunto de soluciones proporcionadas por el enfoque N2FM que se propuso en este trabajo.

9.2. TRABAJO FUTURO

Como parte del trabajo futuro sobre la selección del modelo completo en bases de datos temporales se lista lo siguiente:

- Extender el marco de referencia de las tareas y métodos de minería de datos temporal.
- Explorar otras alternativas de búsqueda, principalmente aquellas alternativas que sean libres de parámetros.
- Evaluar enfoques de optimización multi-modal.
- Proponer un esquema de selección de modelo completo en el contexto de procesamiento de imágenes.
- Evaluar alternativas de distribución para agilizar el proceso de optimización.

[1] Denominado como AutoML, por Auto Machine Learning.

- Establecer un esquema de comparación contra otros trabajos relacionados del estado del arte.

10

Modelos obtenidos por cada variante de μ-DEMS

10.1. Modelos de base de datos Beef

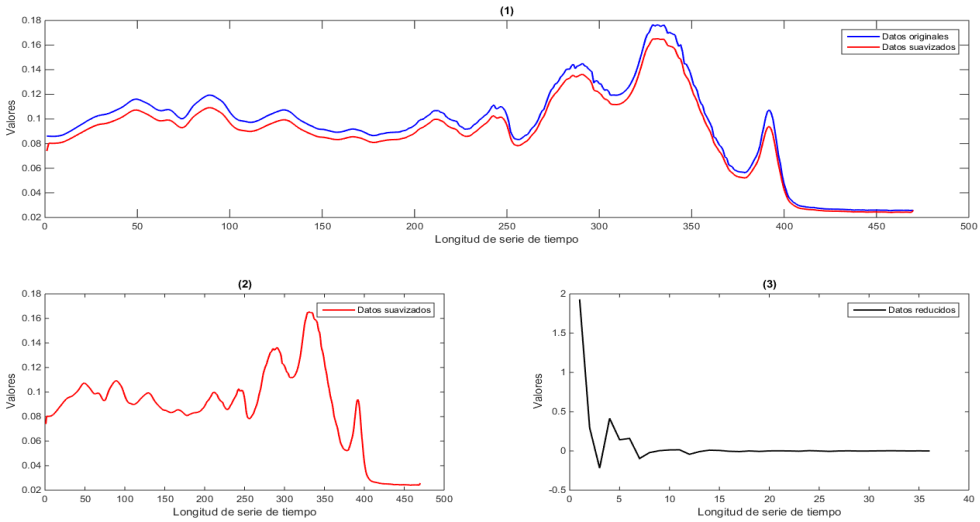


Figura 10.1: Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

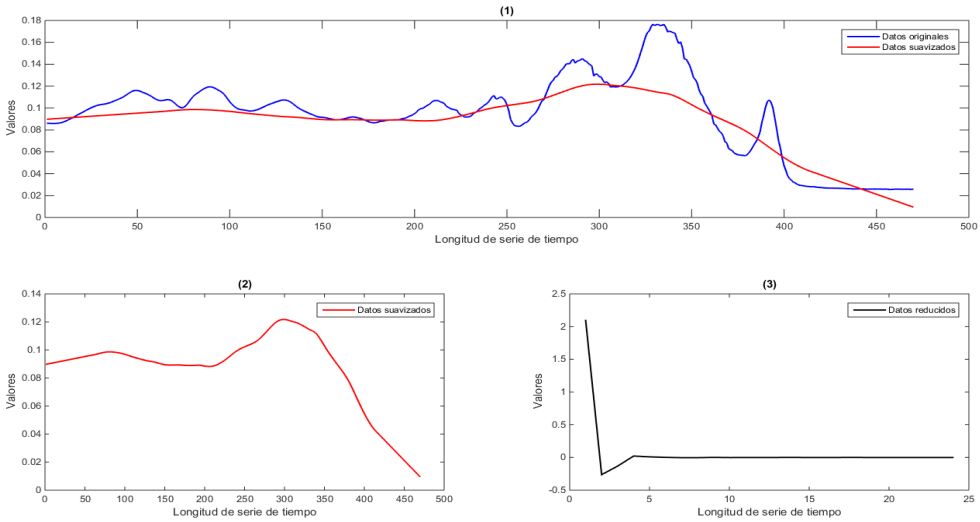


Figura 10.2: Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

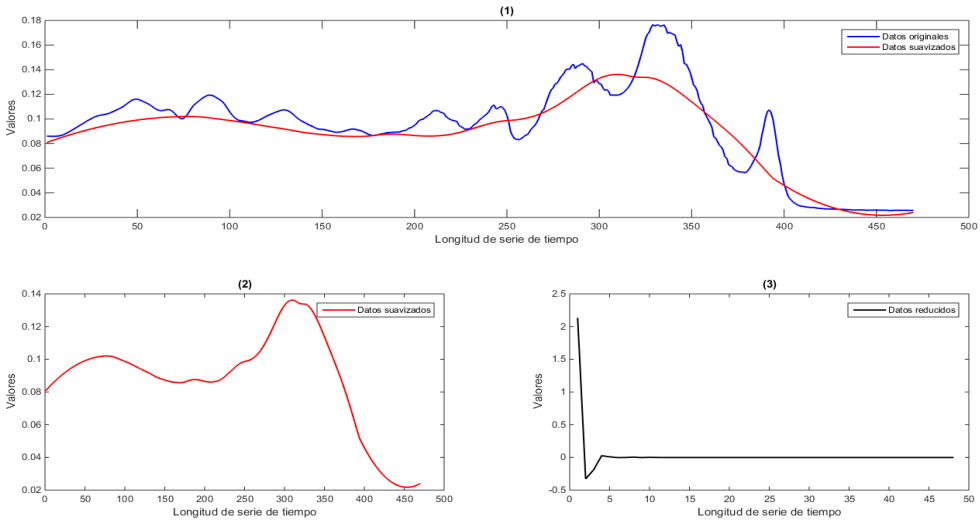


Figura 10.3: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

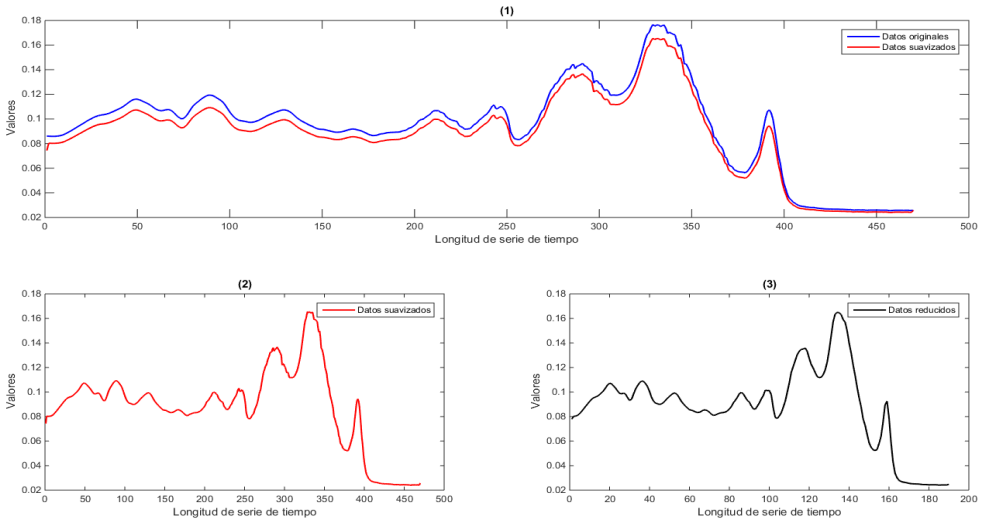


Figura 10.4: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

10.2. MODELOS DE BASE DE DATOS COFFEE

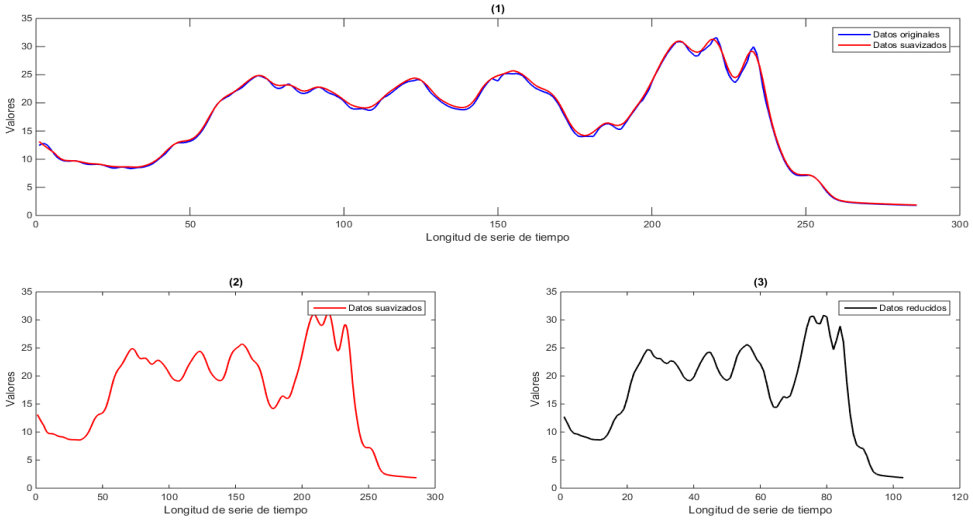


Figura 10.5: Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

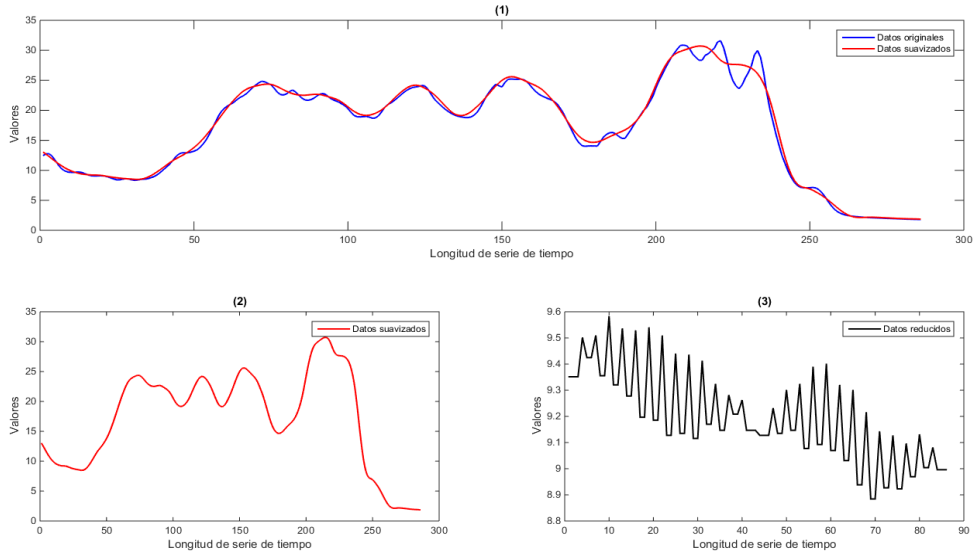


Figura 10.6: Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

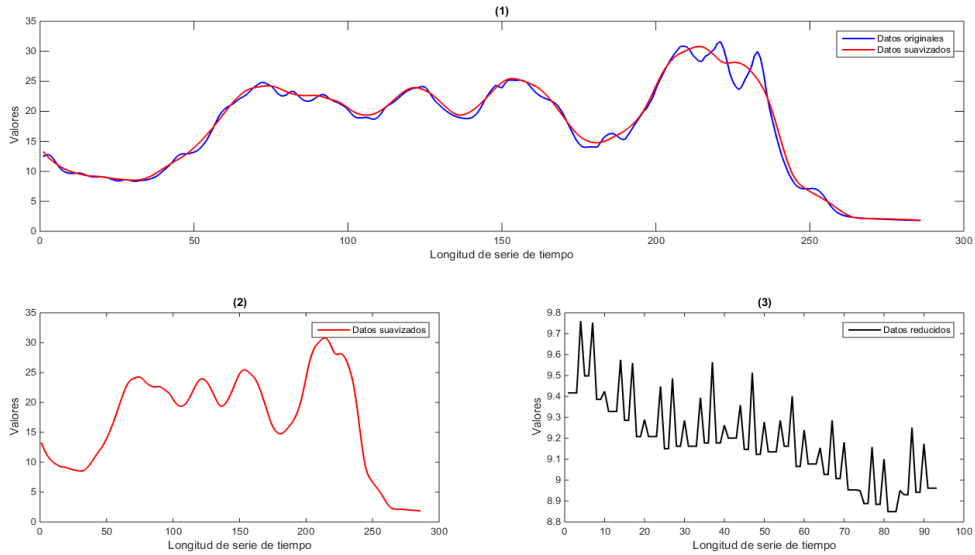


Figura 10.7: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

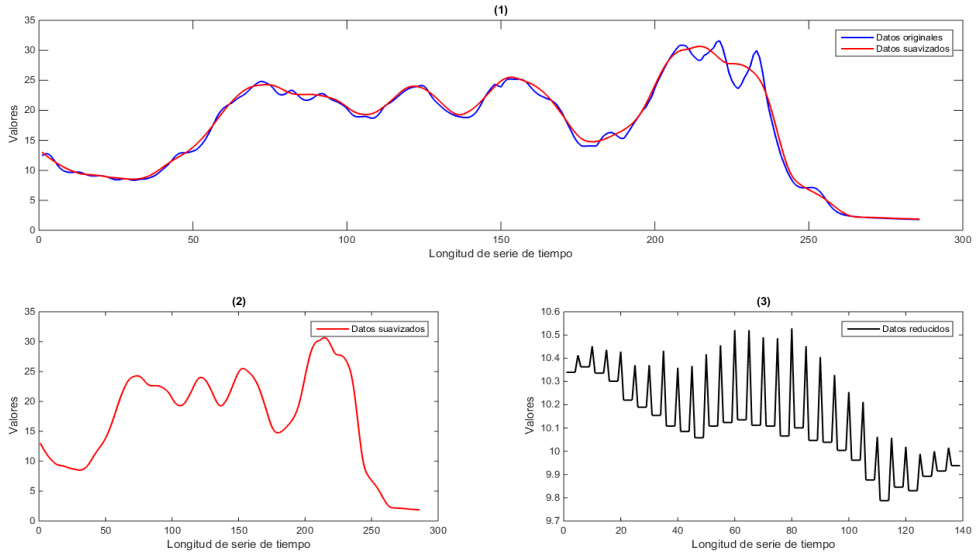


Figura 10.8: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

10.3. MODELOS DE BASE DE DATOS ECG200

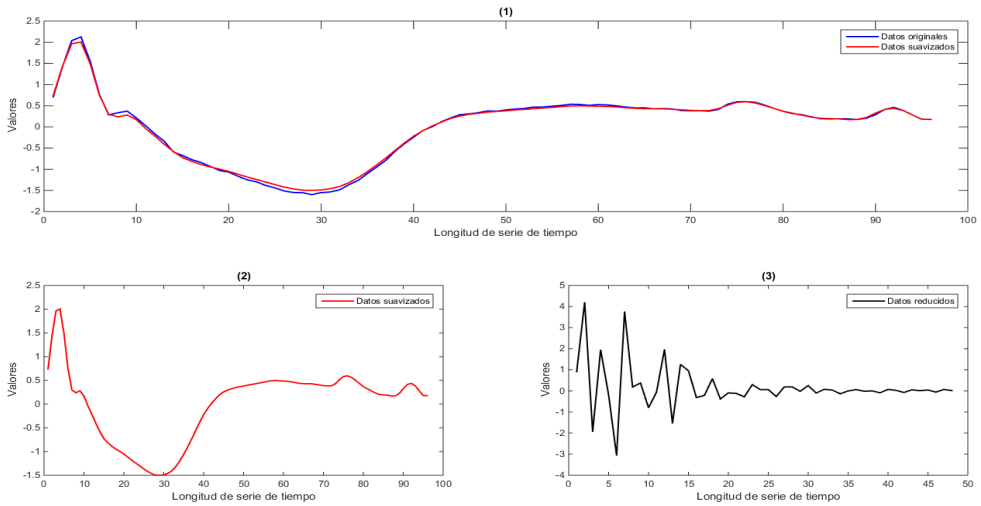


Figura 10.9: Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

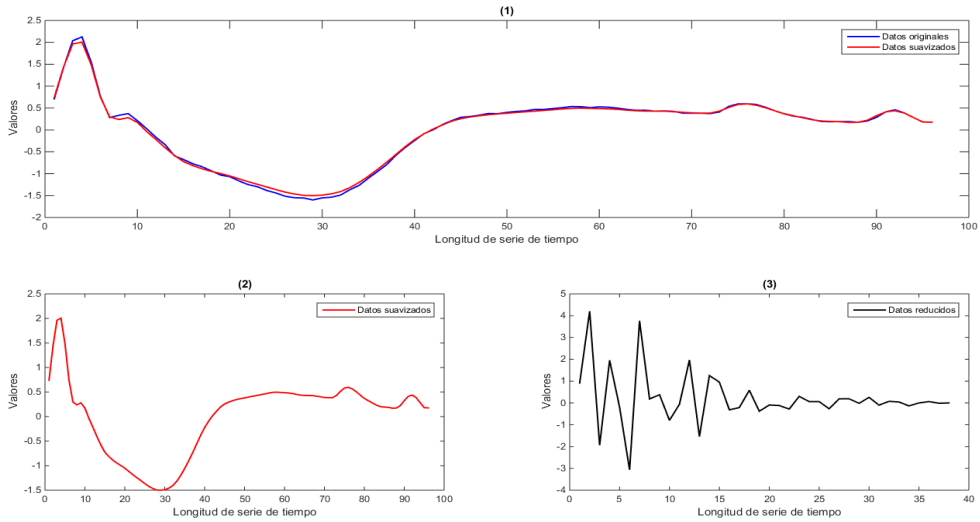


Figura 10.10: Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

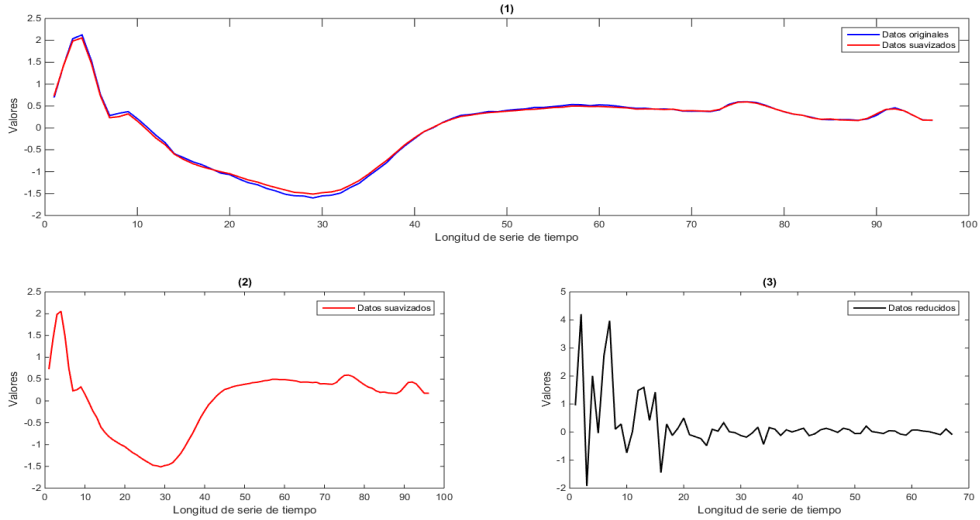


Figura 10.11: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

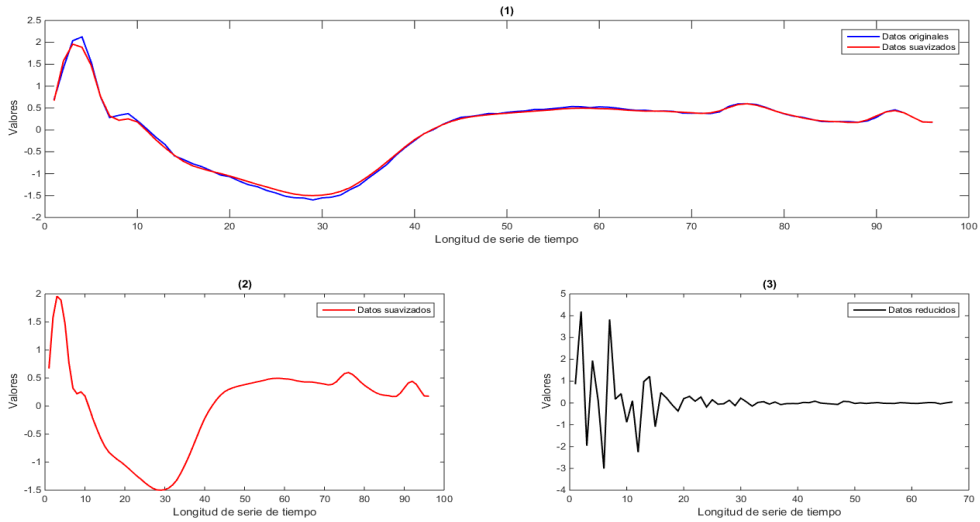


Figura 10.12: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

10.4. MODELOS DE BASE DE DATOS OLIVEOIL

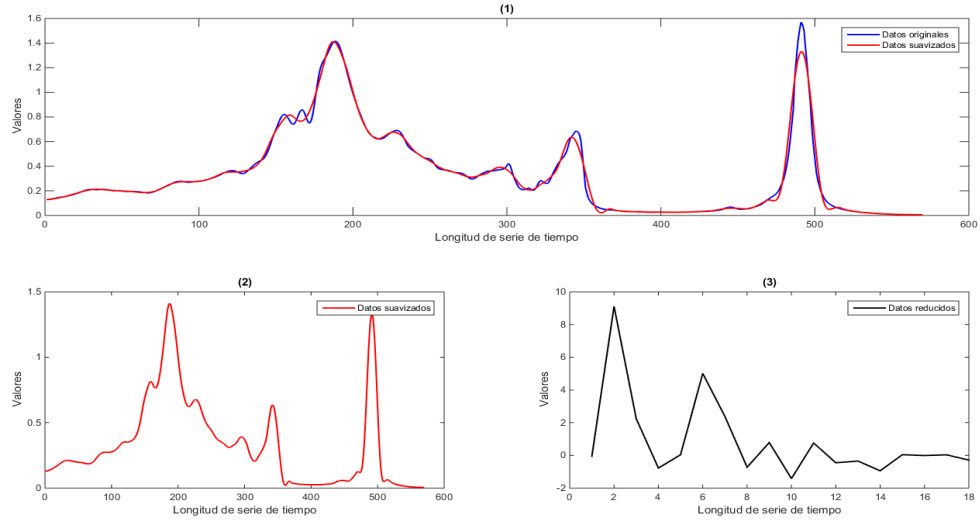


Figura 10.14: Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

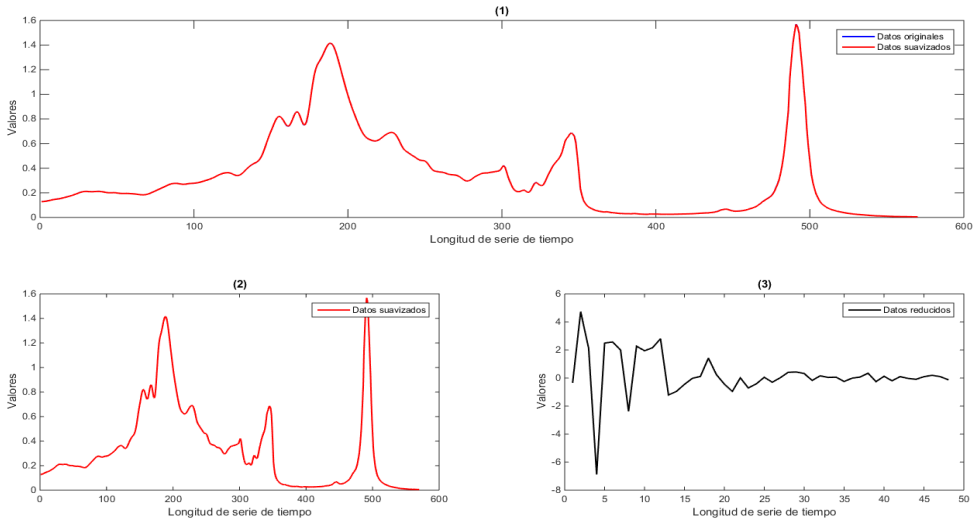


Figura 10.13: Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

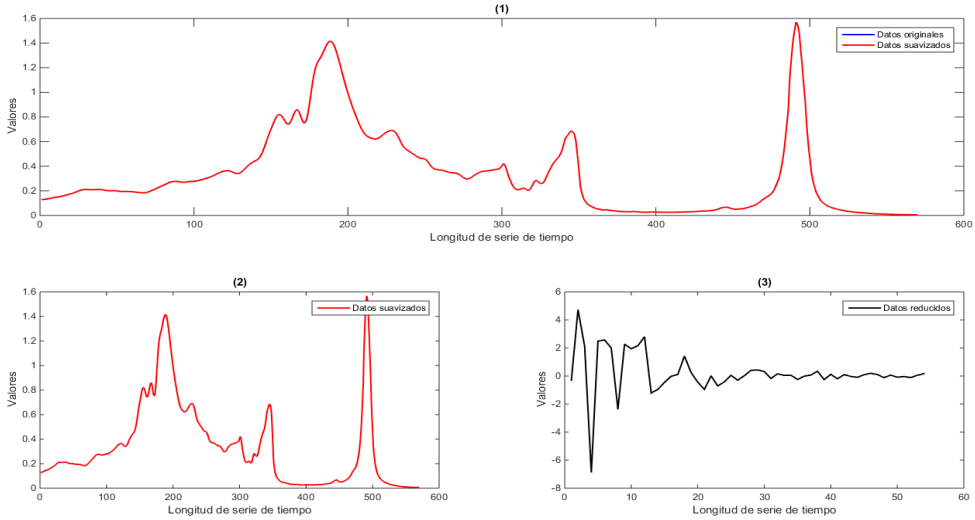


Figura 10.15: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

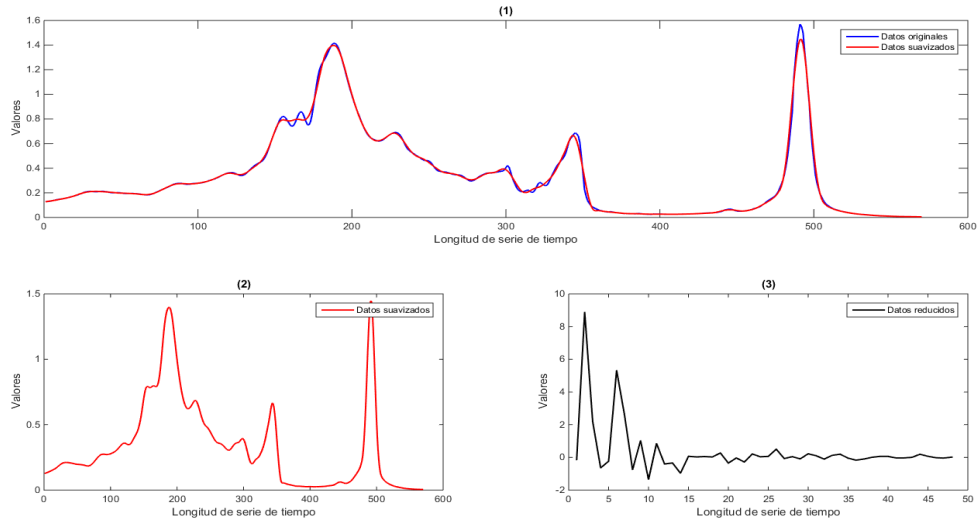


Figura 10.16: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

10.5. MODELOS DE BASE DE DATOS FACEFOUR

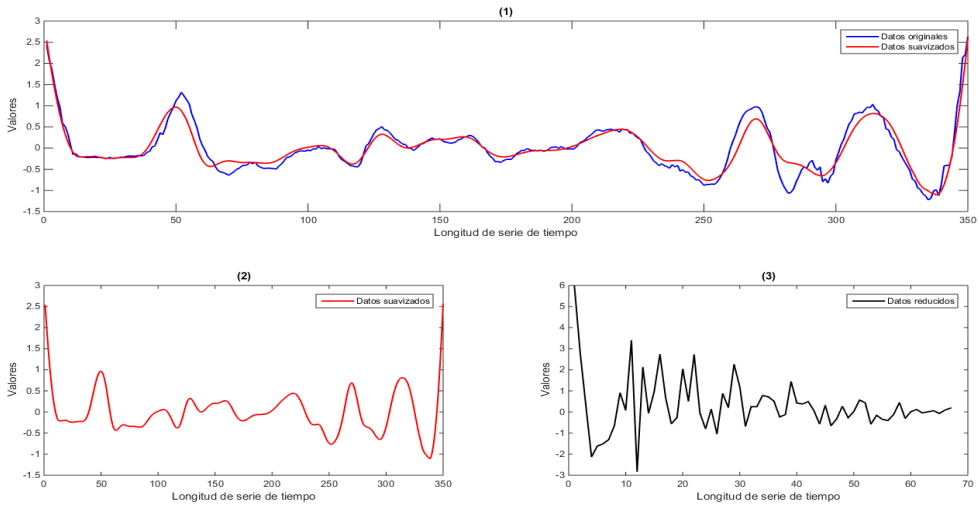


Figura 10.17: Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

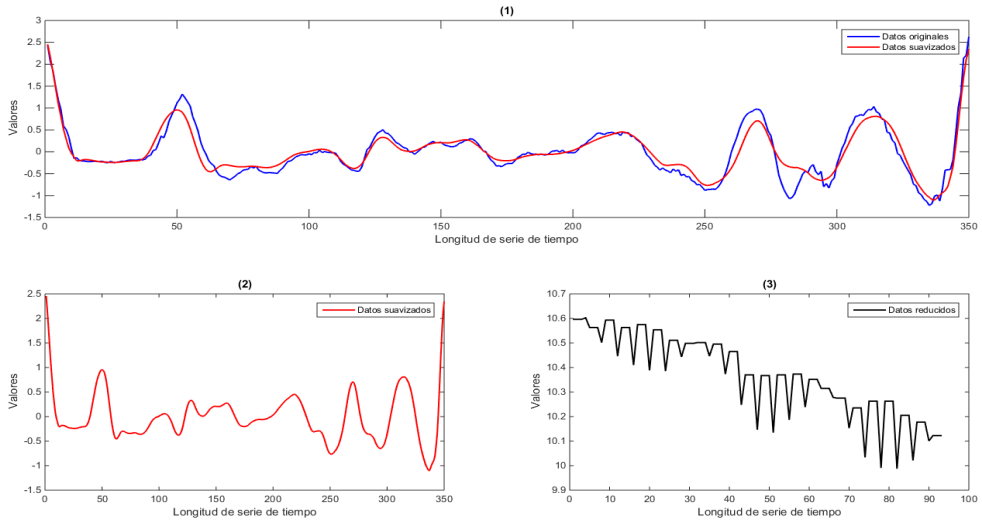


Figura 10.18: Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

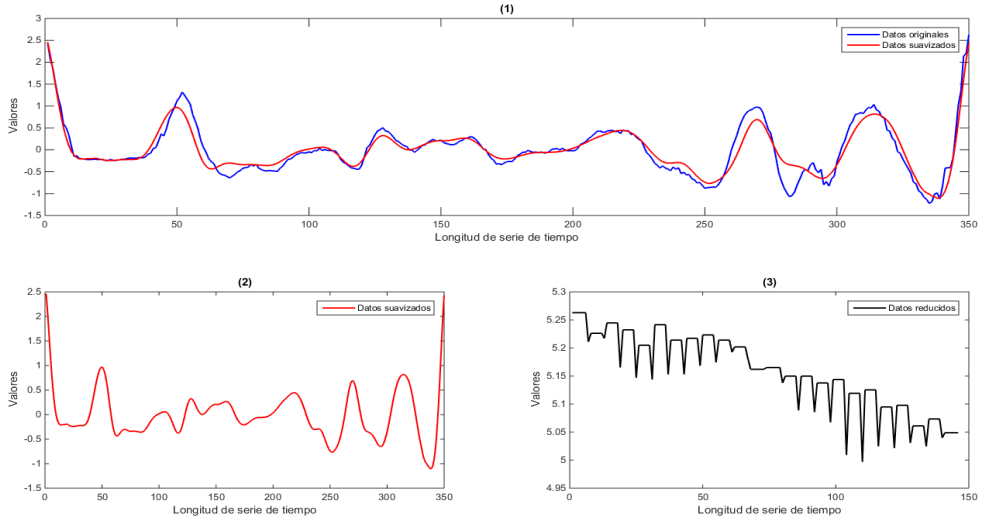


Figura 10.19: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

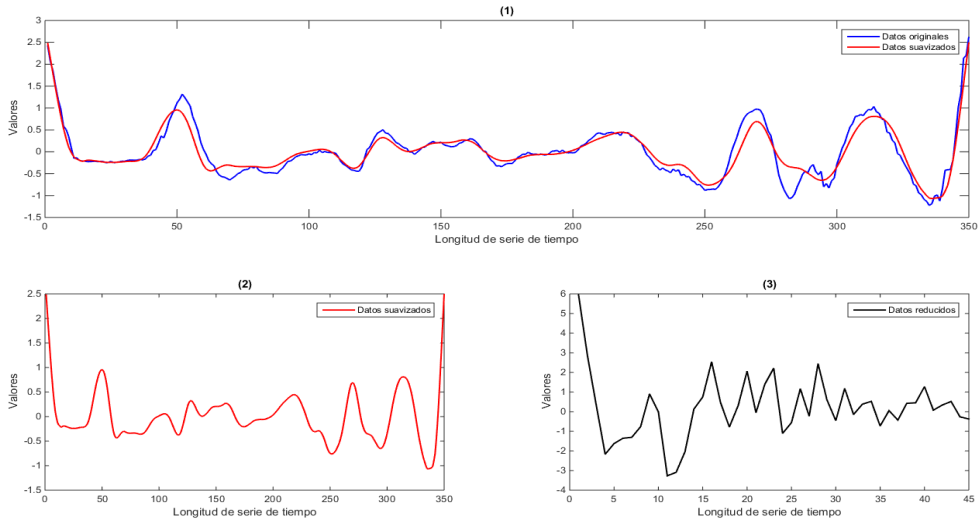


Figura 10.20: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

10.6. MODELOS DE BASE DE DATOS GUNPOINT

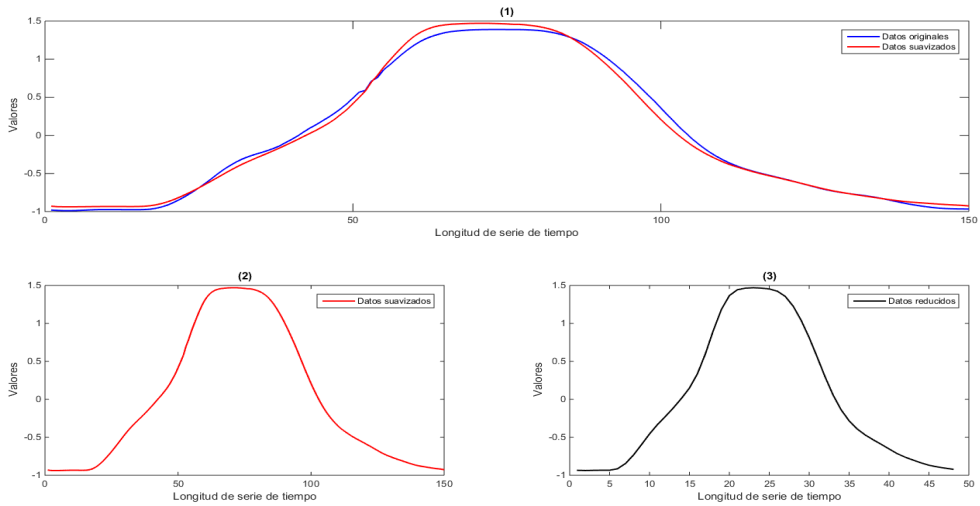


Figura 10.21: Representación gráfica del modelo sugerido por la variante *rand/1/bin* para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

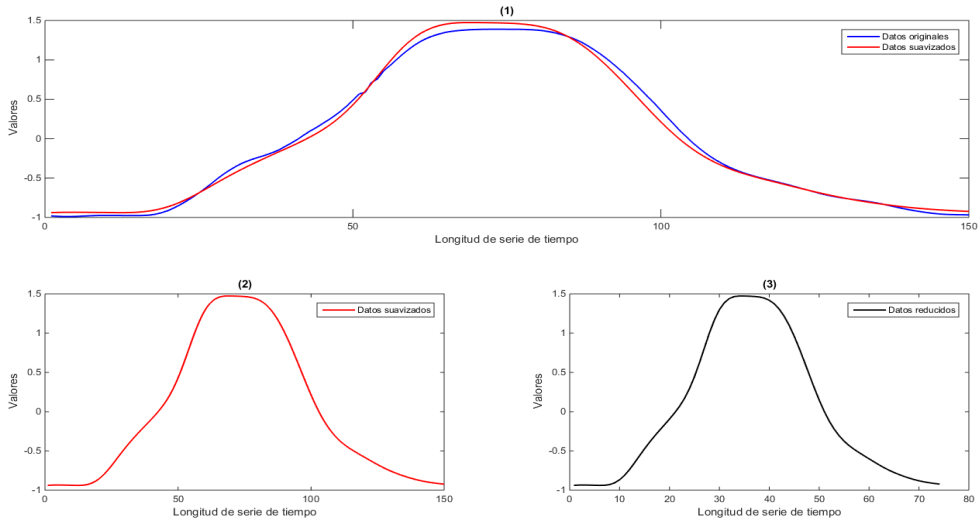


Figura 10.22: Representación gráfica del modelo sugerido por la variante *rand/1/exp* para la base de datos Gun-Point. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

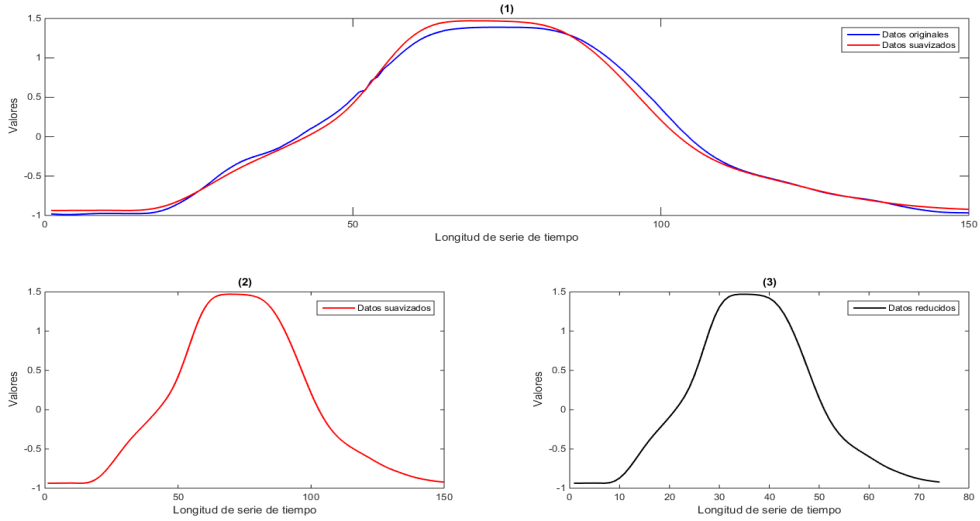


Figura 10.23: Representación gráfica del modelo sugerido por la variante *best/1/bin* para la base de datos Gun-Point. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

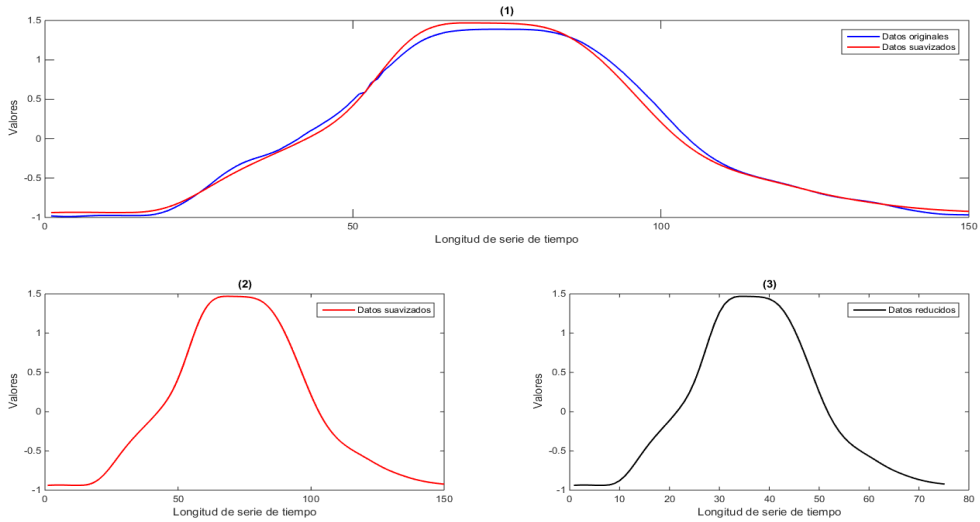


Figura 10.24: Representación gráfica del modelo sugerido por la variante *best/1/exp* para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

11

Modelos obtenidos por cada Metaheurística

11.1. Modelos de base de datos Beef

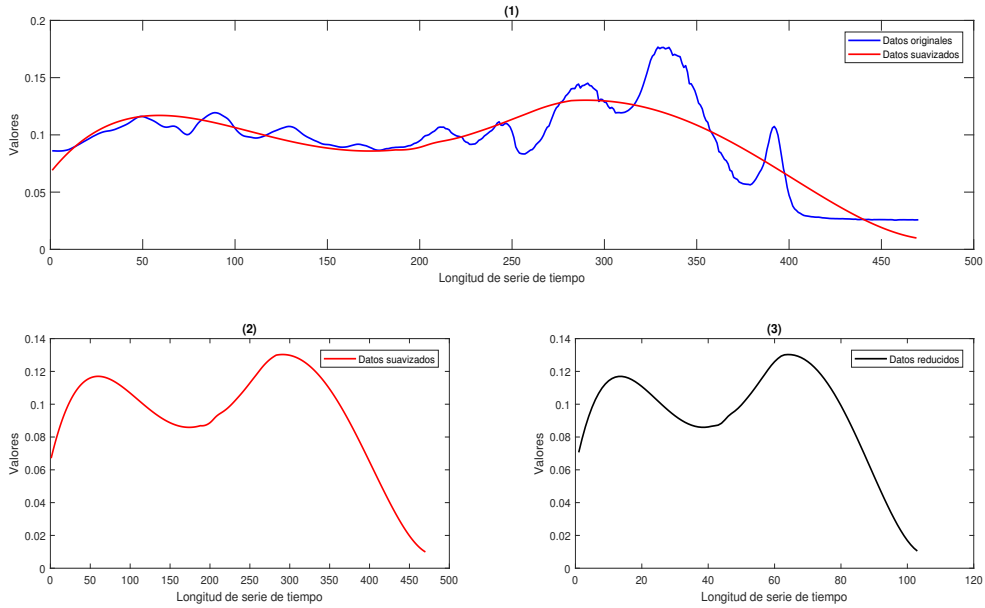


Figura 11.1: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

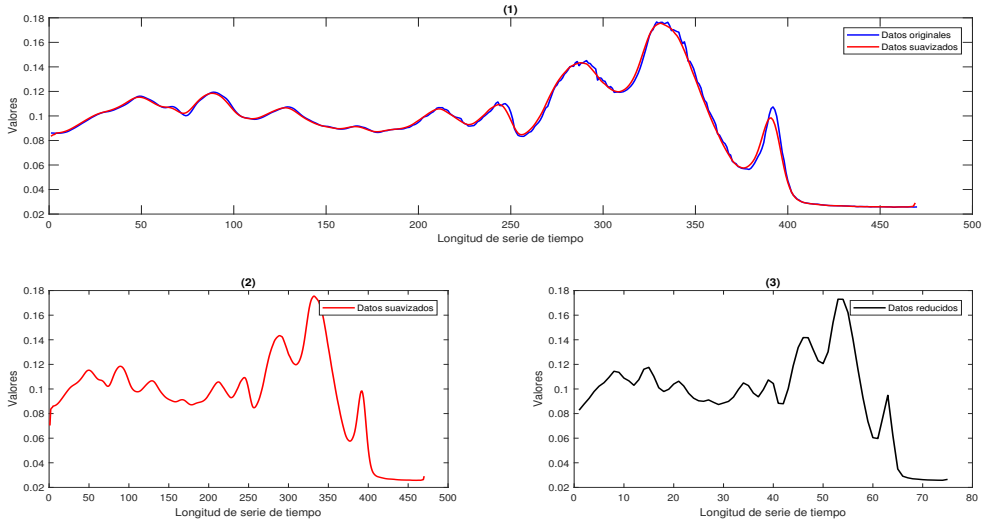


Figura 11.2: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

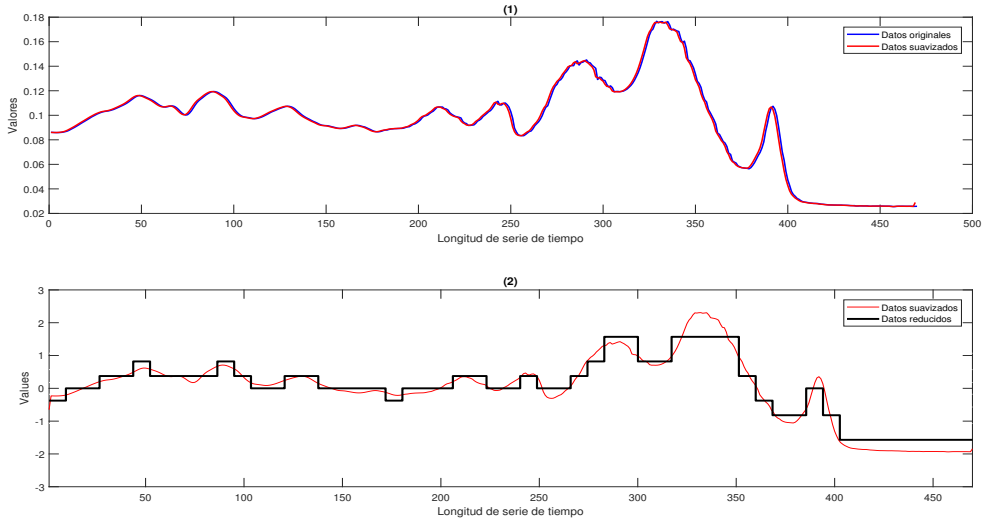


Figura 11.3: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

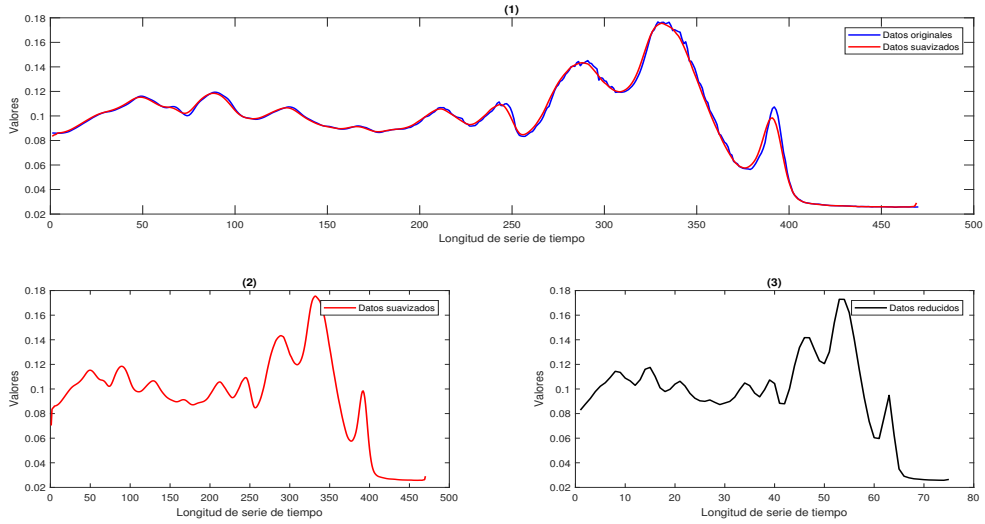


Figura 11.4: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

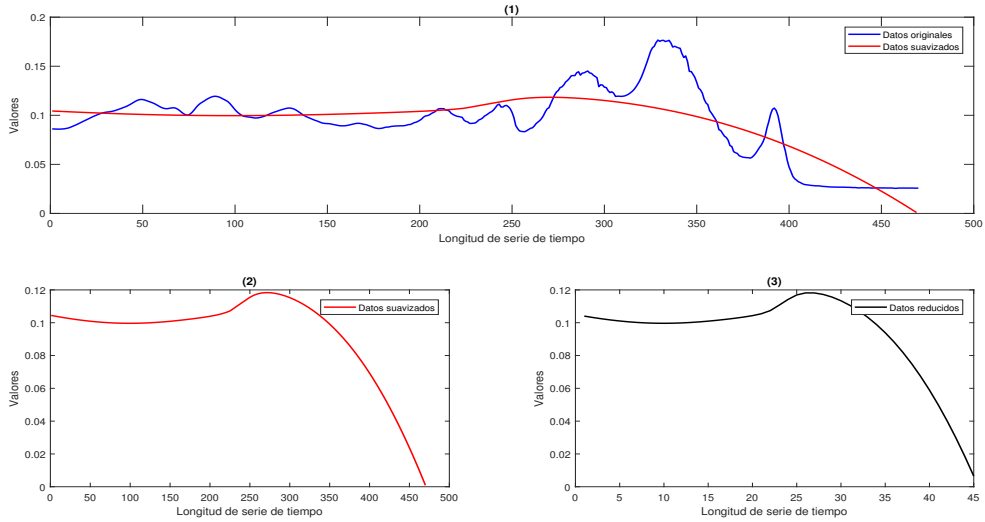


Figura 11.5: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

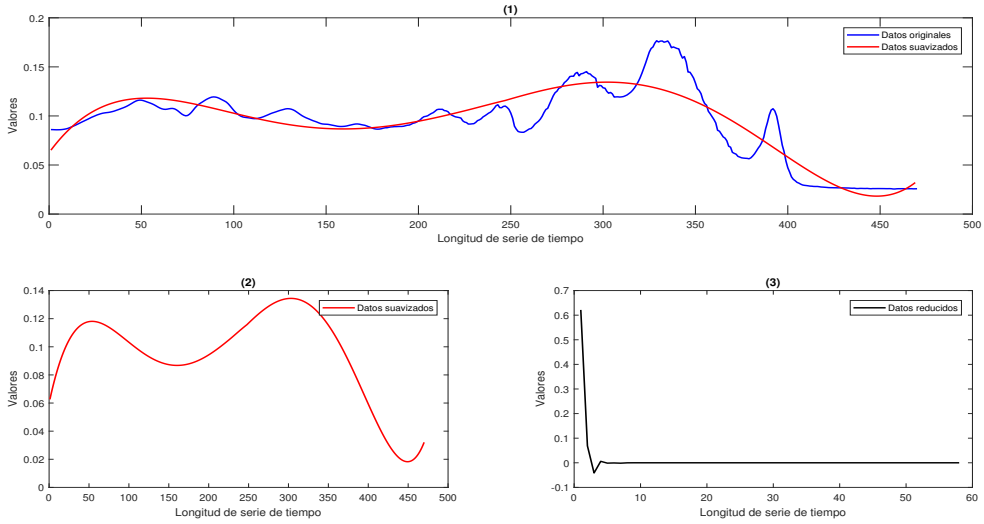


Figura 11.6: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Beef. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

11.2. MODELOS DE BASE DE DATOS CBF

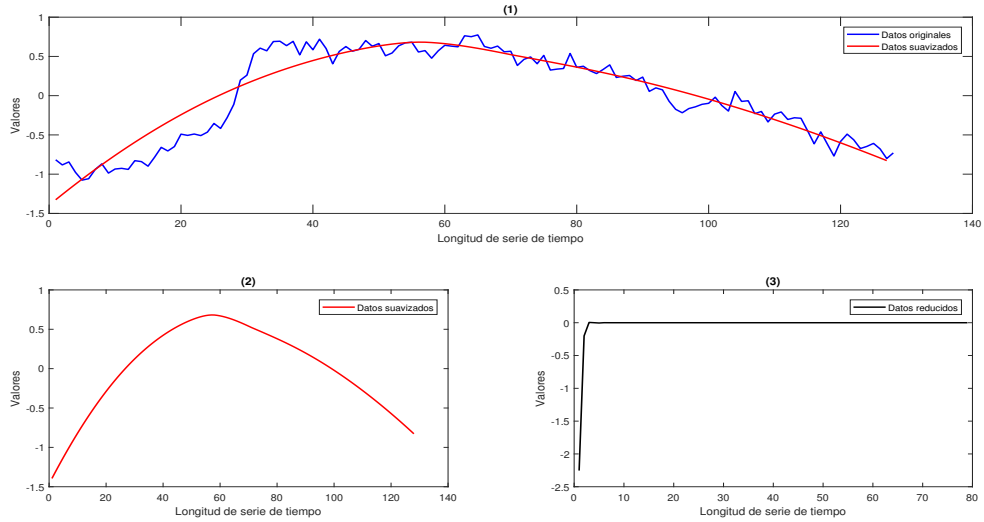


Figura 11.7: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

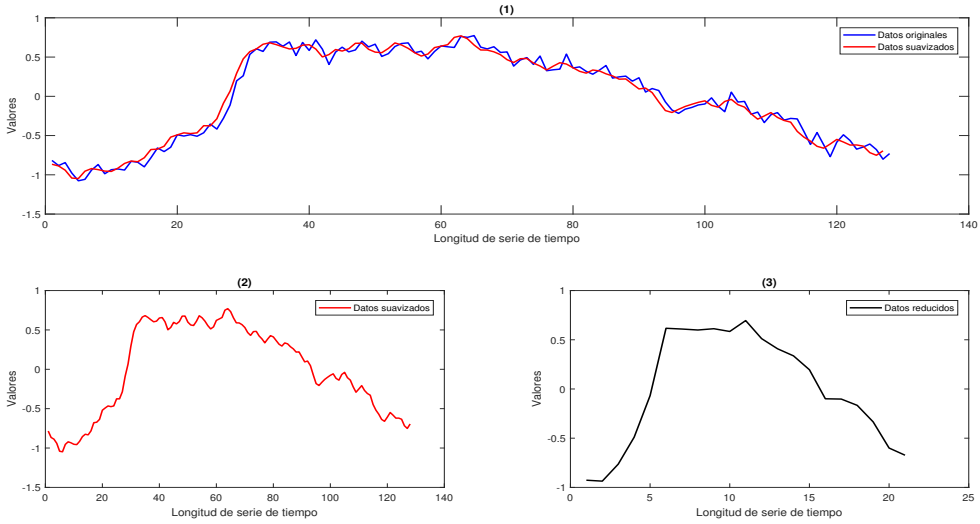


Figura 11.8: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

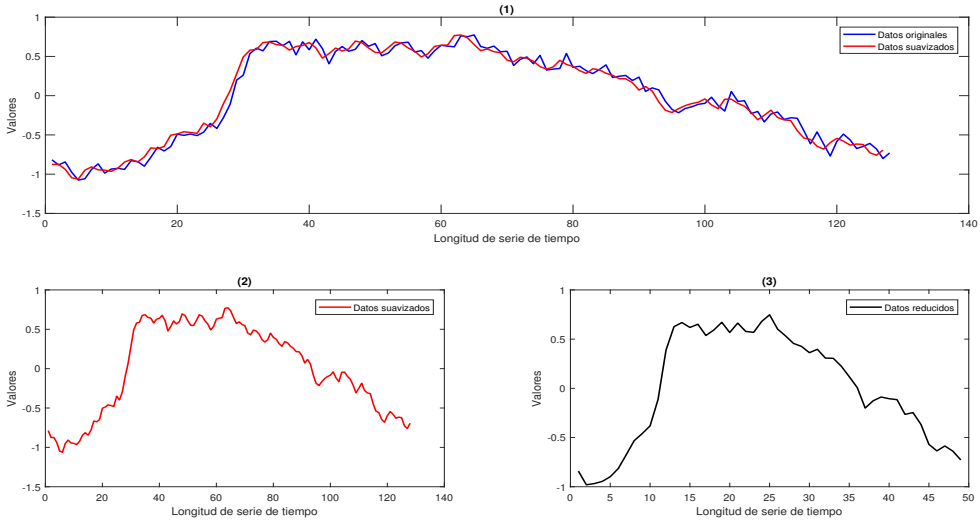


Figura 11.9: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

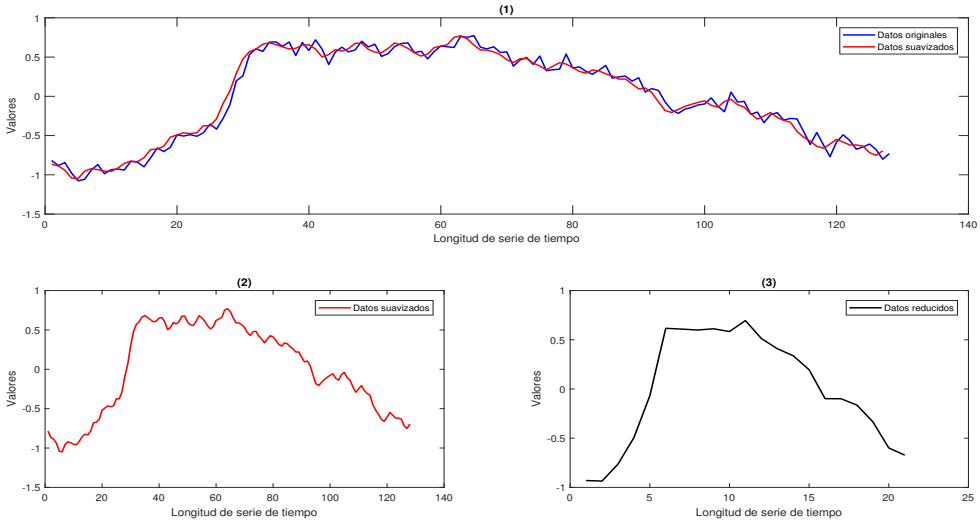


Figura 11.10: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

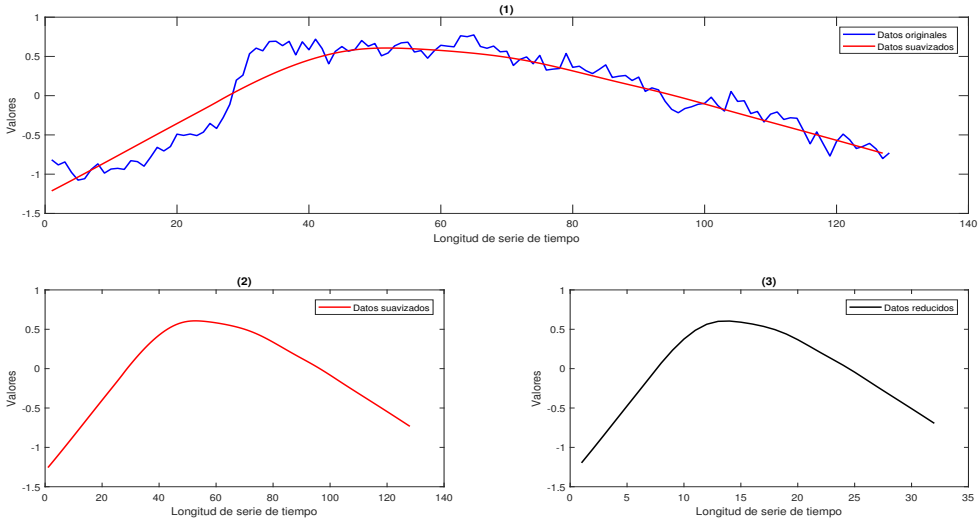


Figura 11.11: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

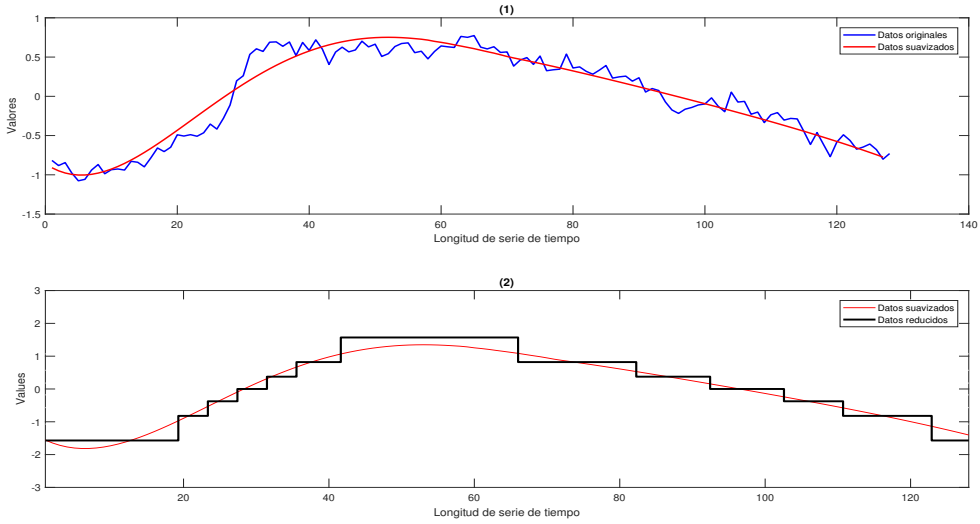


Figura 11.12: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos CBF. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

11.3. MODELOS DE BASE DE DATOS COFFEE

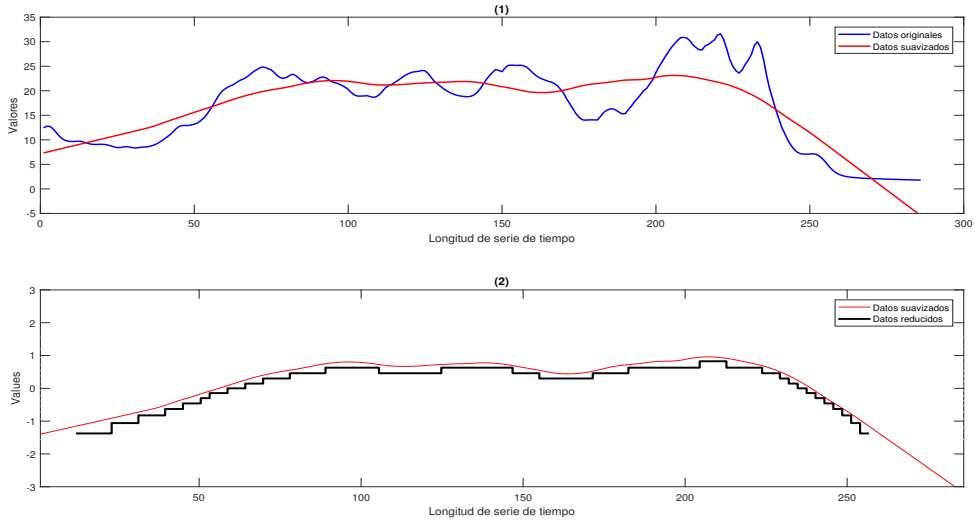


Figura 11.13: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

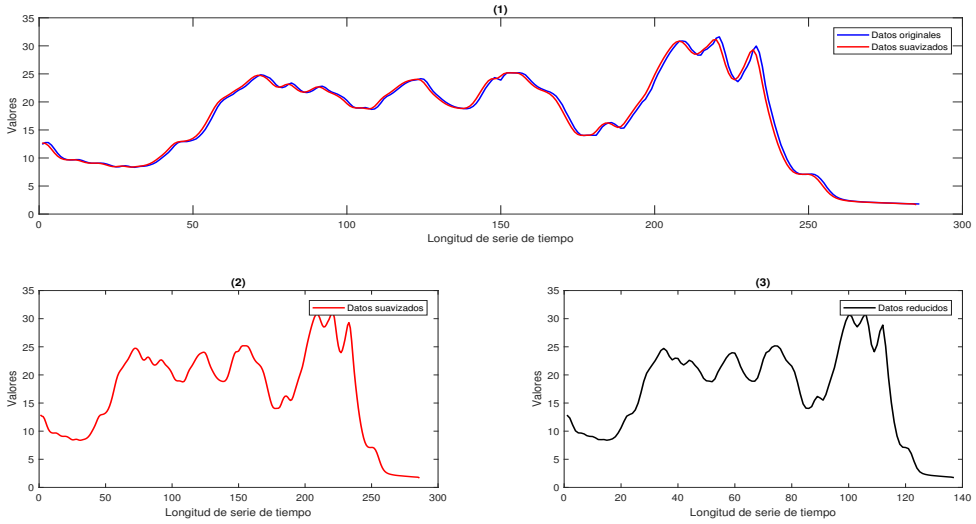


Figura 11.14: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

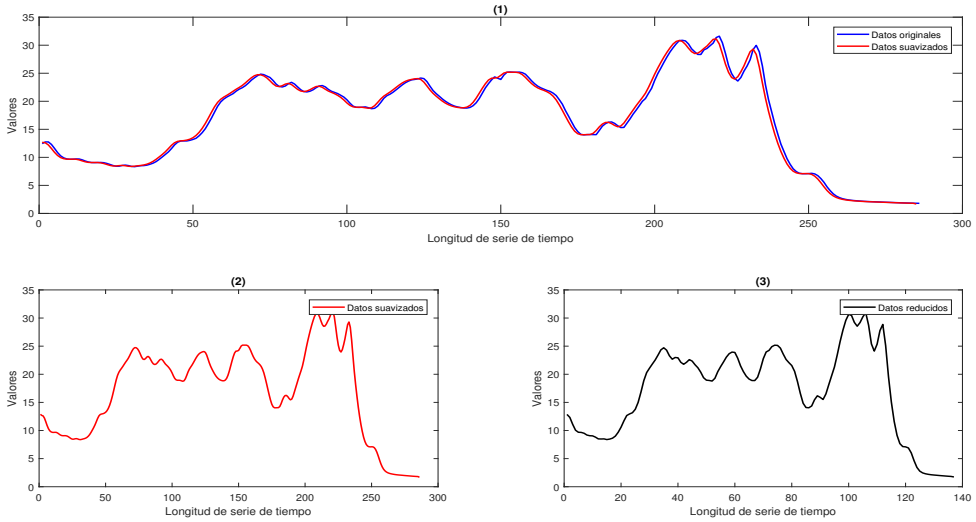


Figura 11.15: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

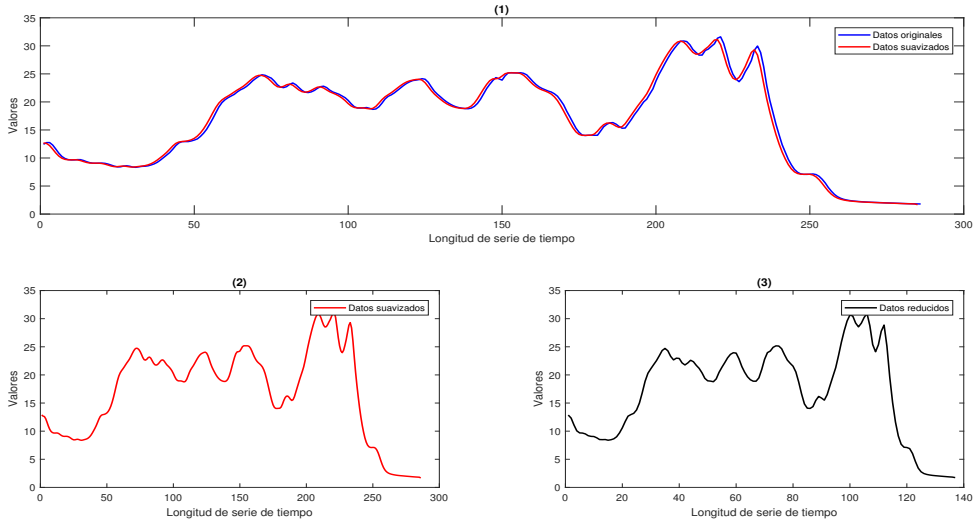


Figura 11.16: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

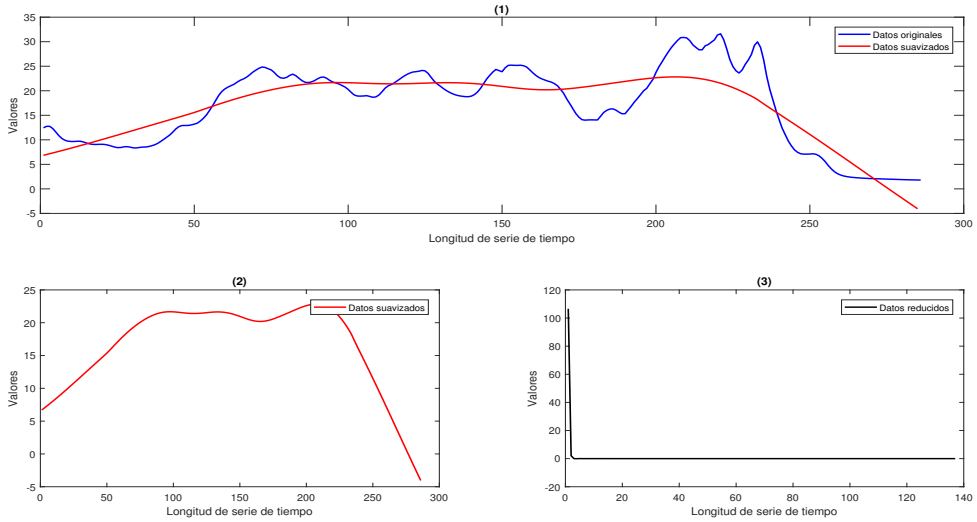


Figura 11.17: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

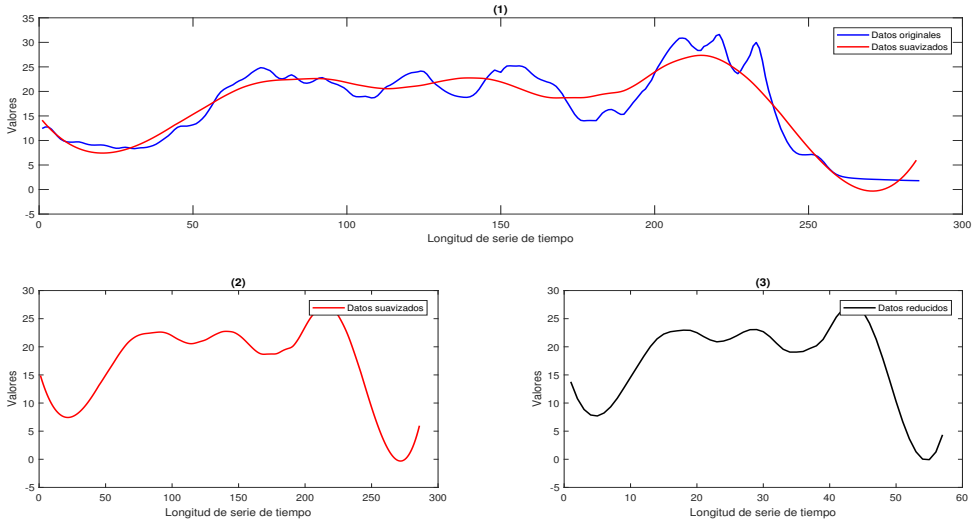


Figura 11.18: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Coffee. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

11.4. MODELOS DE BASE DE DATOS ECG200

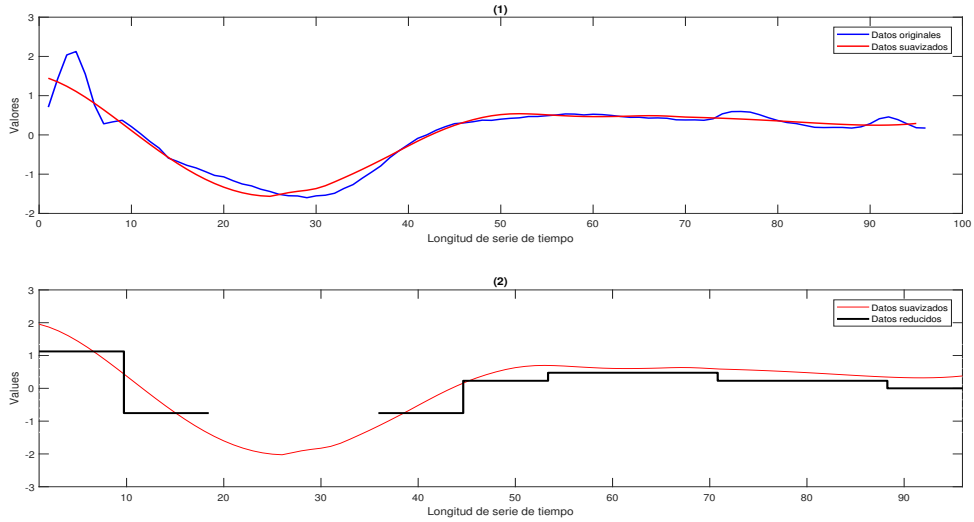


Figura 11.19: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

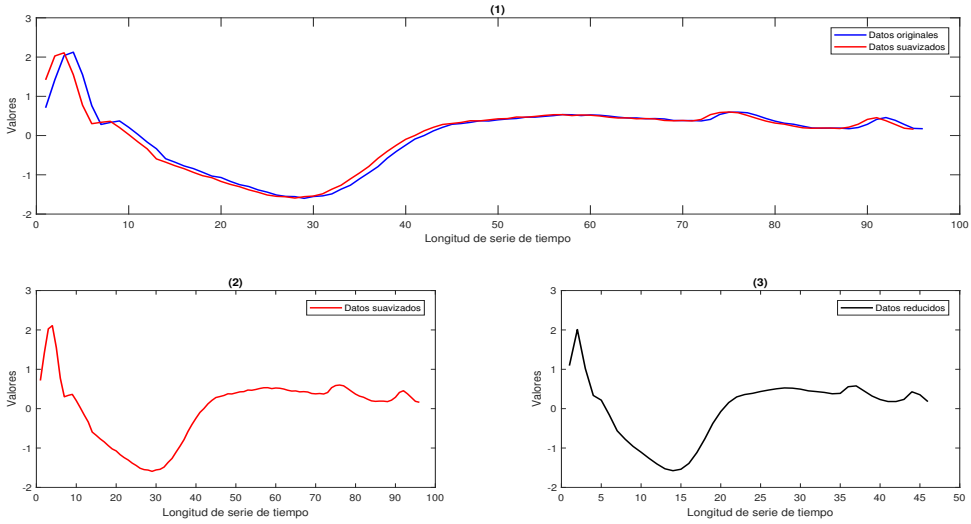


Figura 11.20: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

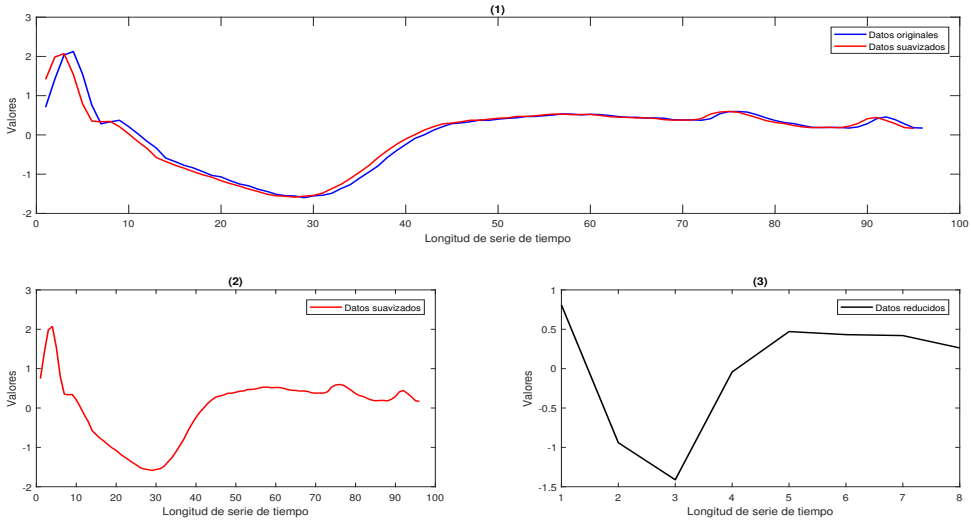


Figura 11.21: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

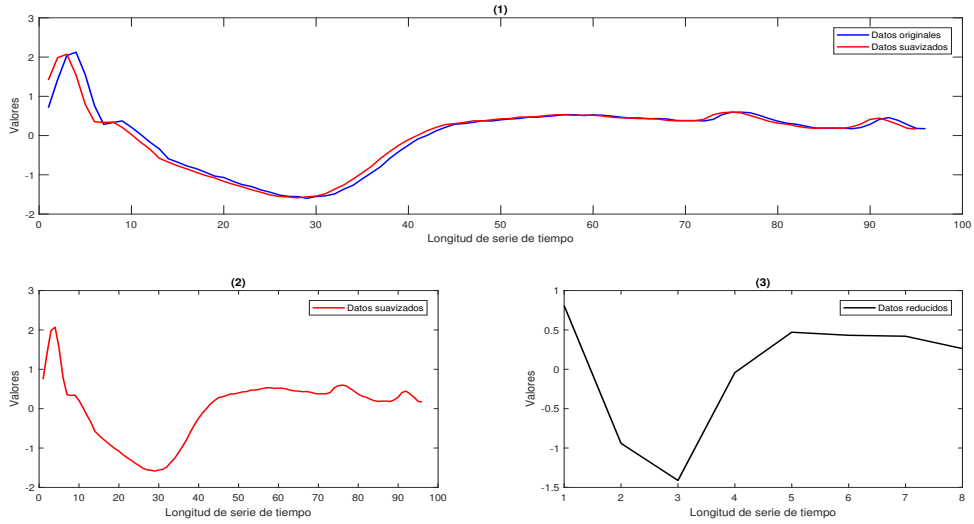


Figura 11.22: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

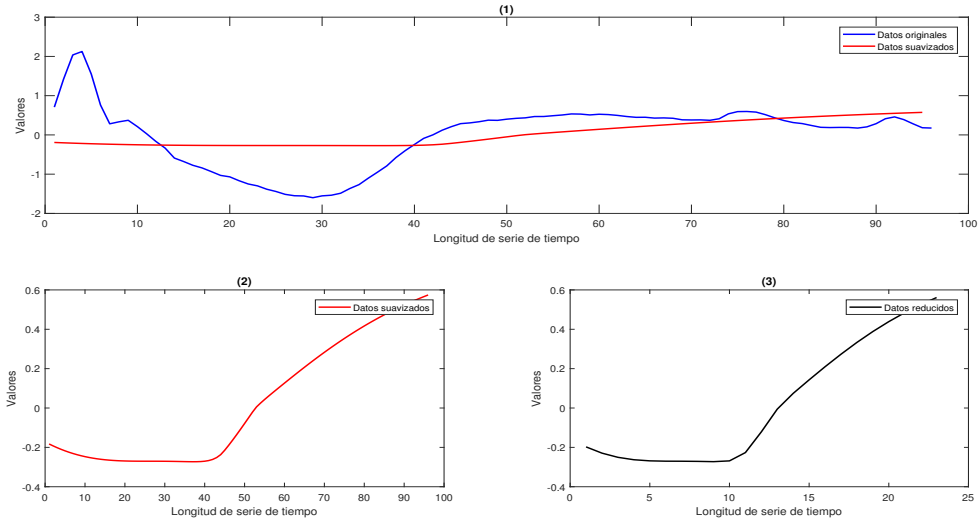


Figura 11.23: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

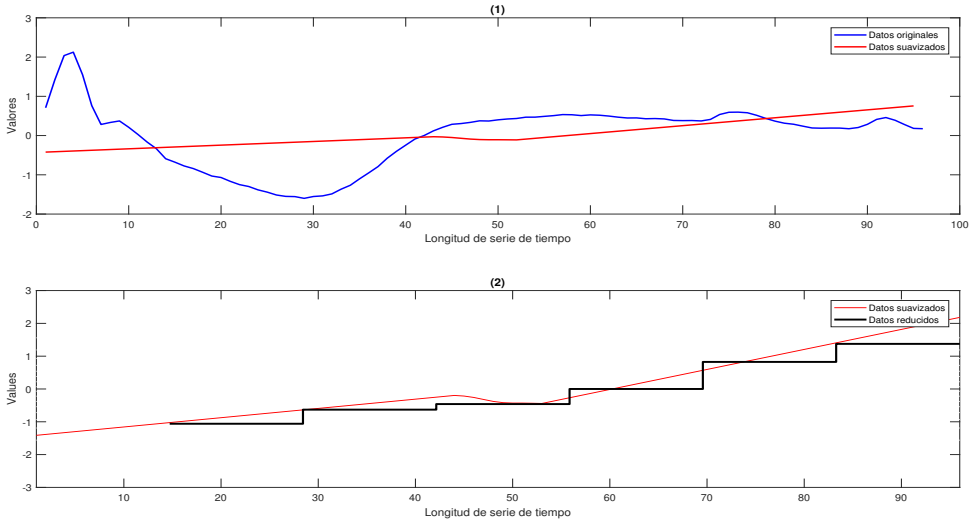


Figura 11.24: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos ECG200. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

11.5. MODELOS DE BASE DE DATOS FACEFOUR

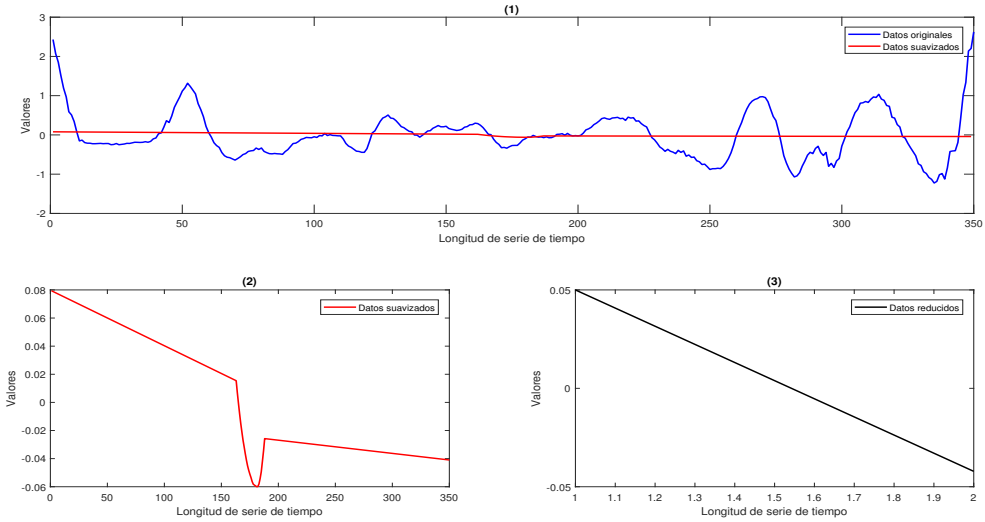


Figura 11.25: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

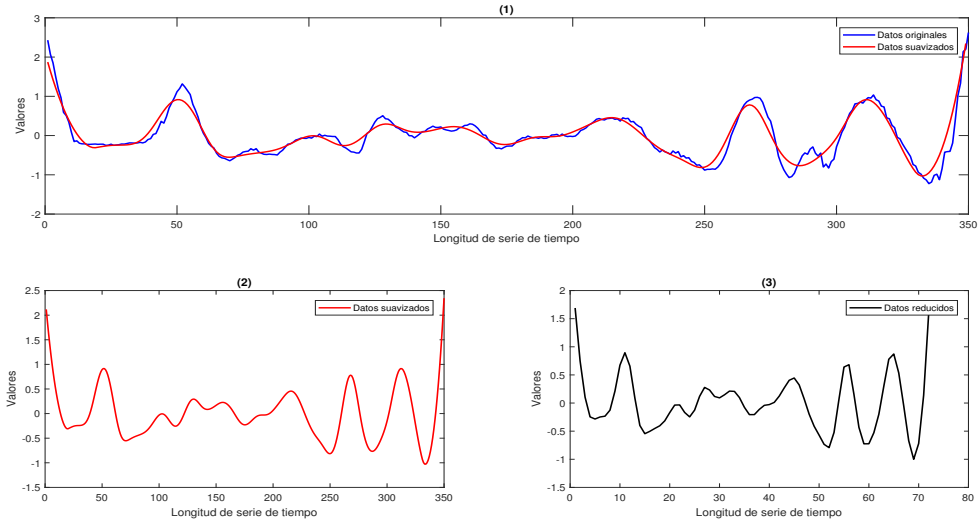


Figura 11.26: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

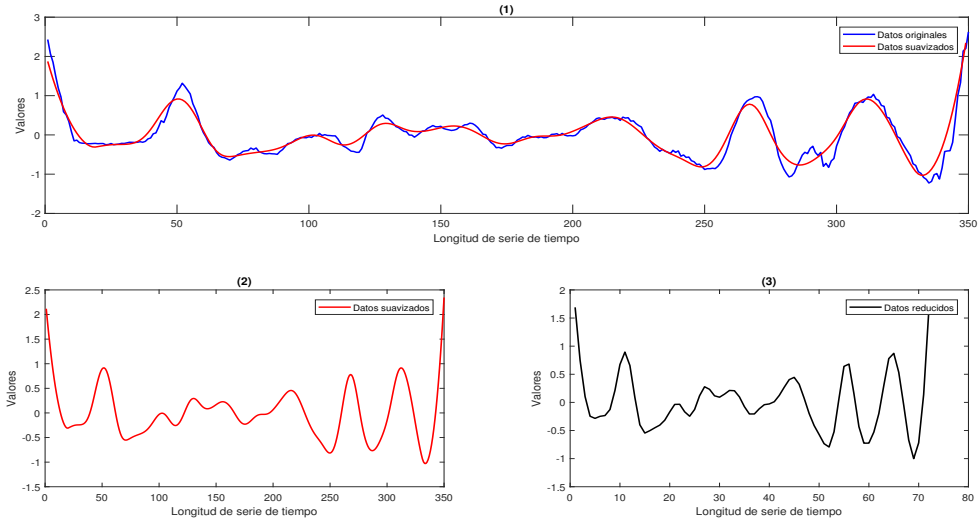


Figura 11.27: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

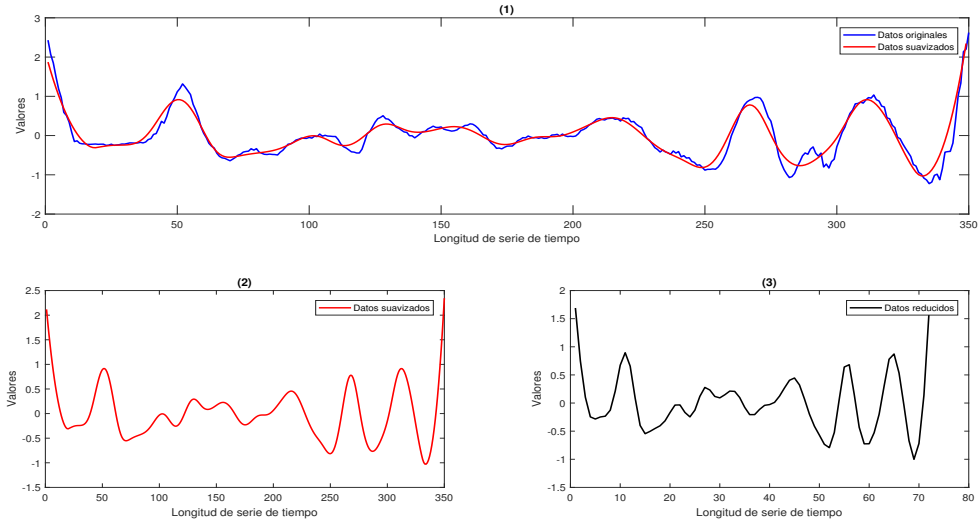


Figura 11.28: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

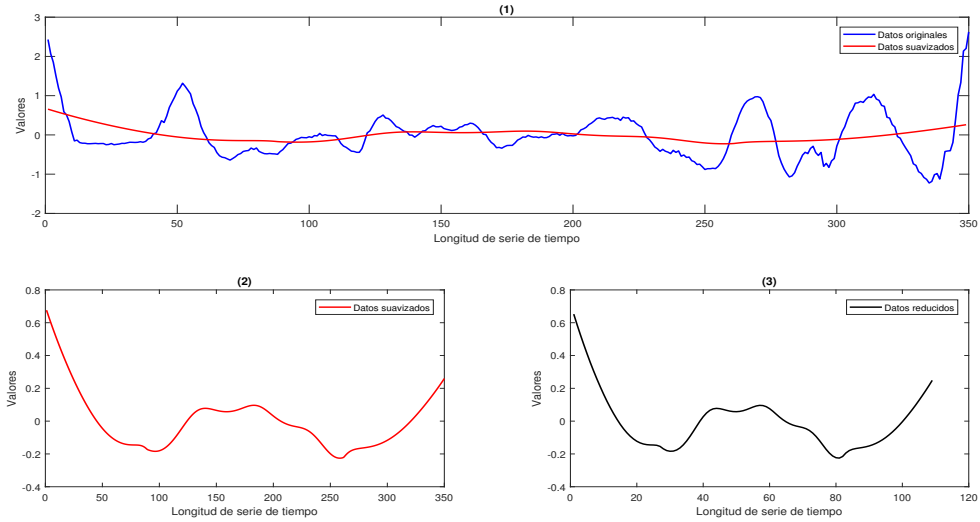


Figura 11.29: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

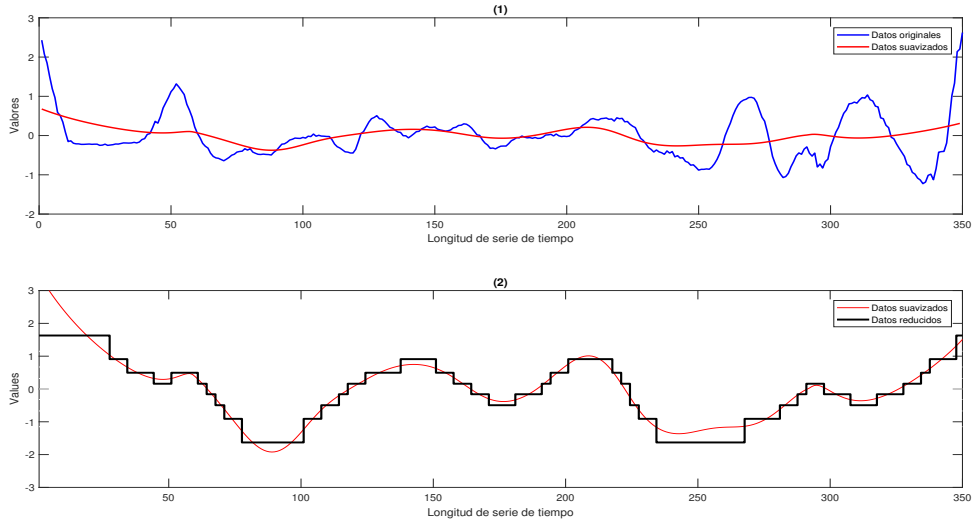


Figura 11.30: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos FaceFour. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

11.6. MODELOS DE BASE DE DATOS GUNPOINT

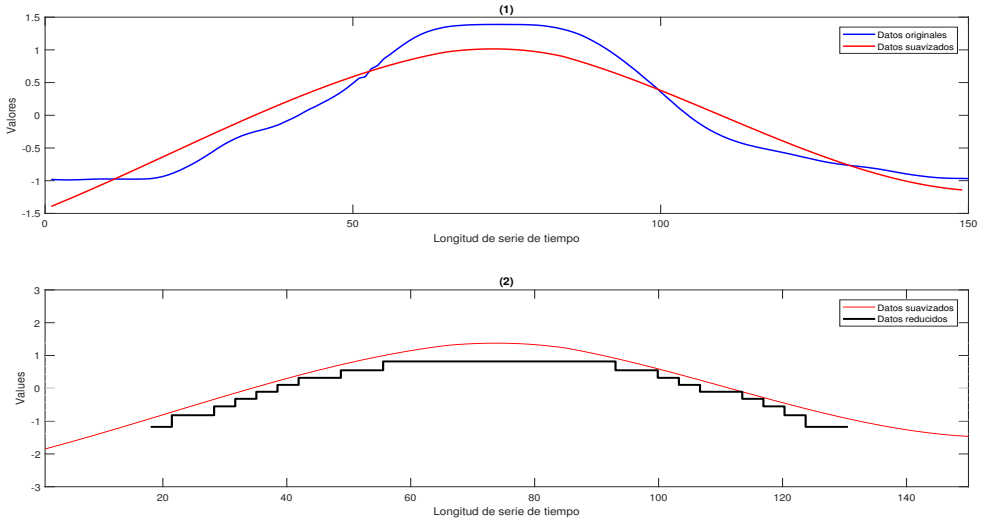


Figura 11.31: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

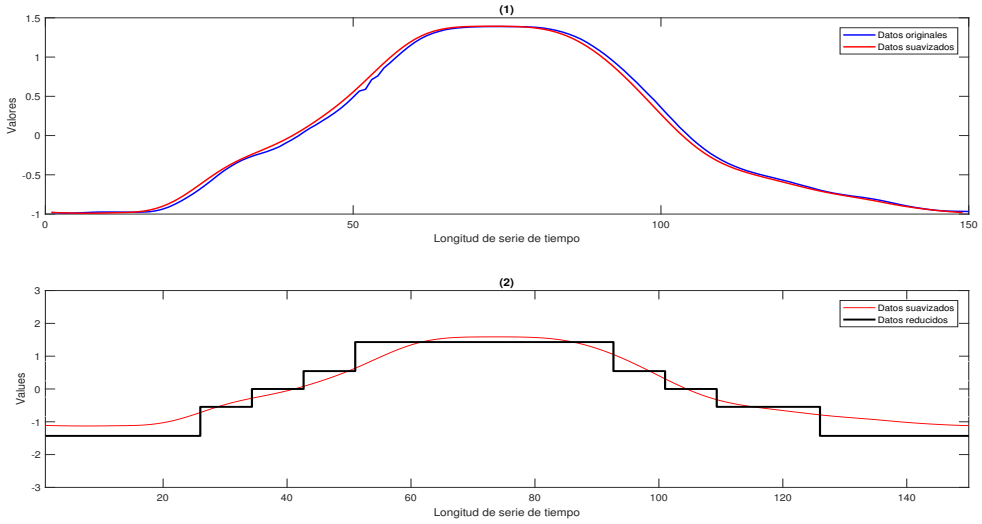


Figura 11.32: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

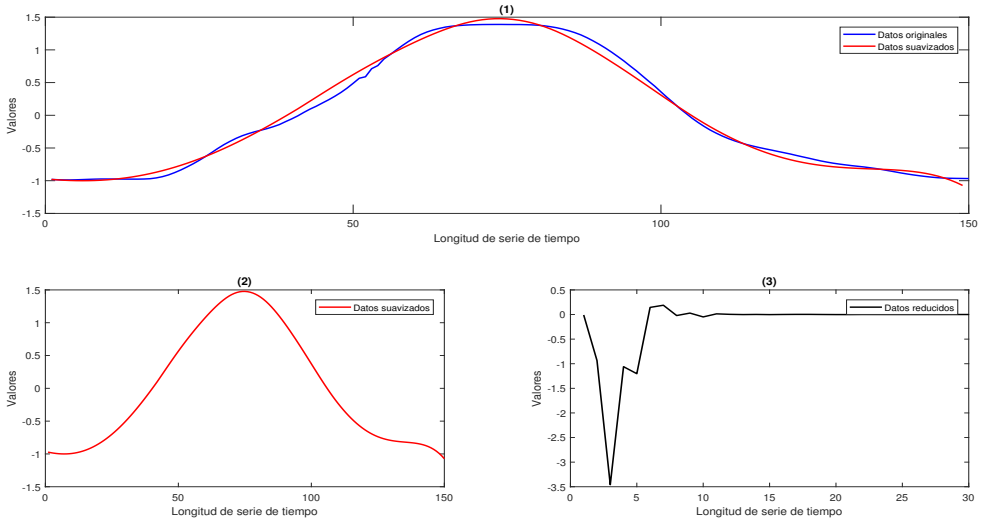


Figura 11.33: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

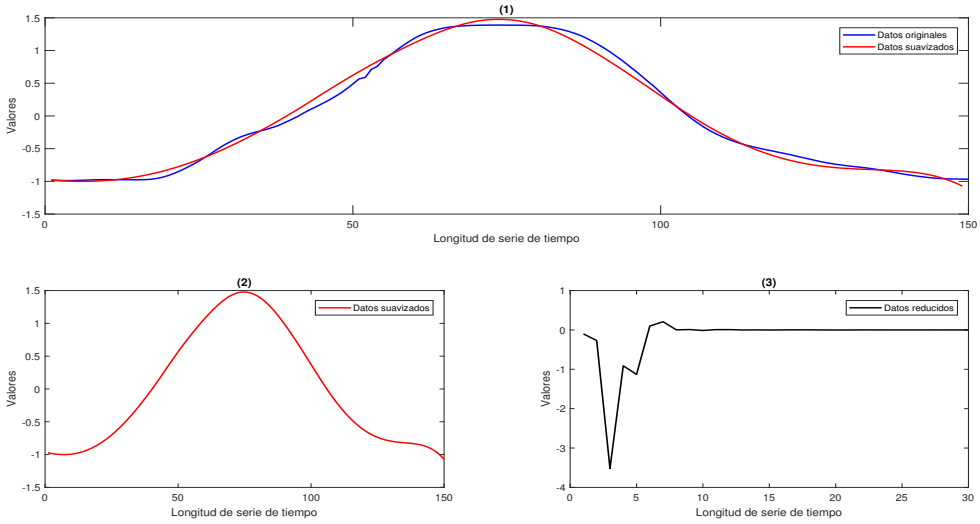


Figura 11.34: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

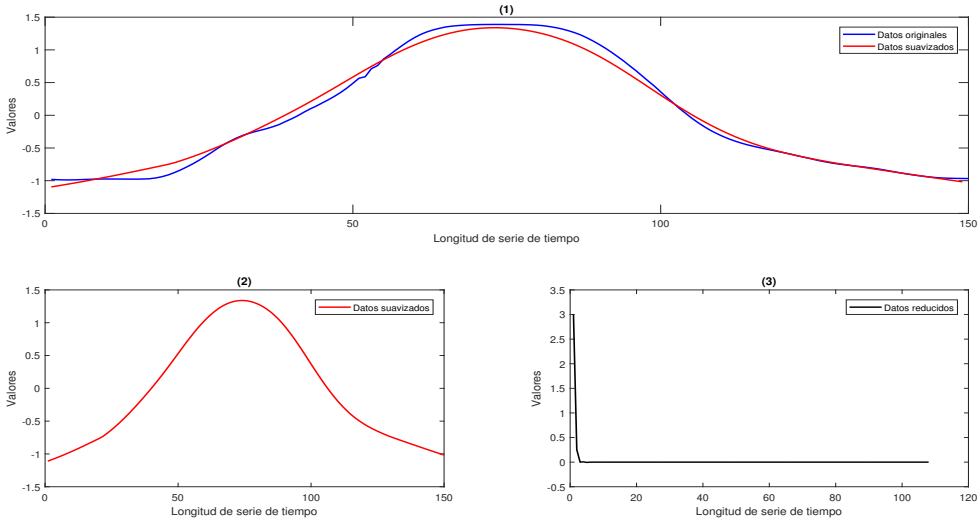


Figura 11.35: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

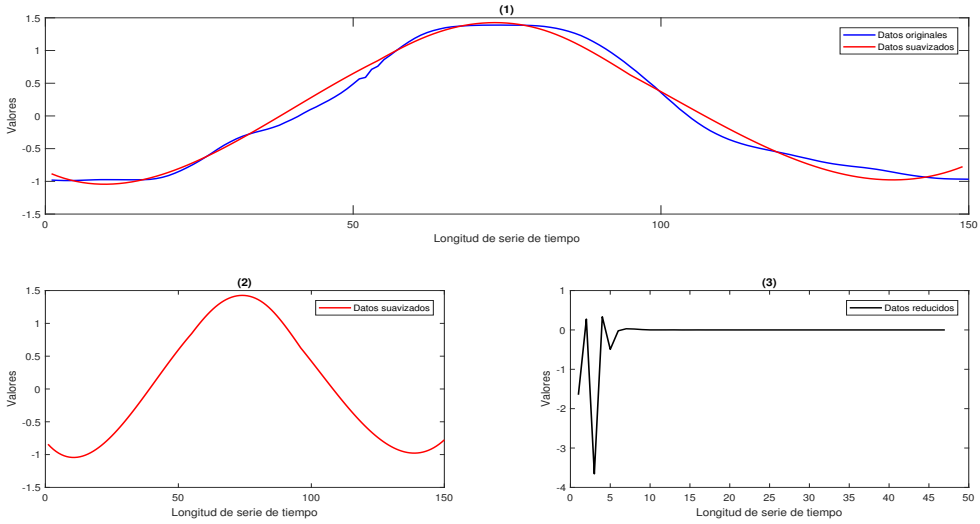


Figura 11.36: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos GunPoint. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

11.7. MODELOS DE BASE DE DATOS LIGHTNING-2

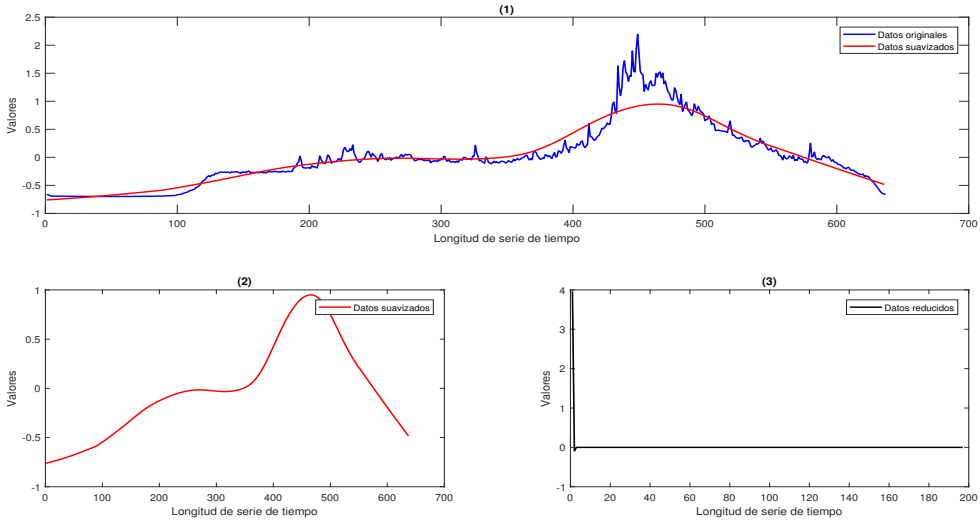


Figura 11.37: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

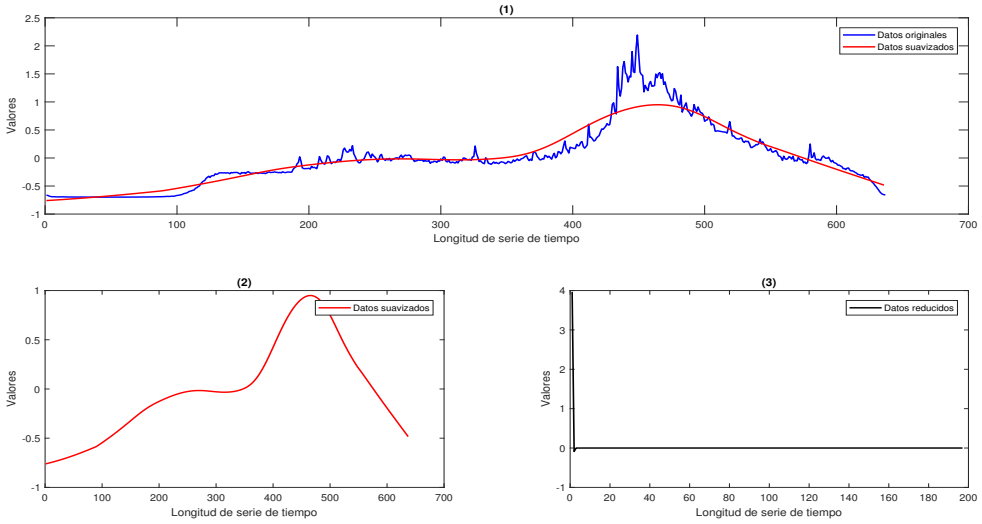


Figura 11.38: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

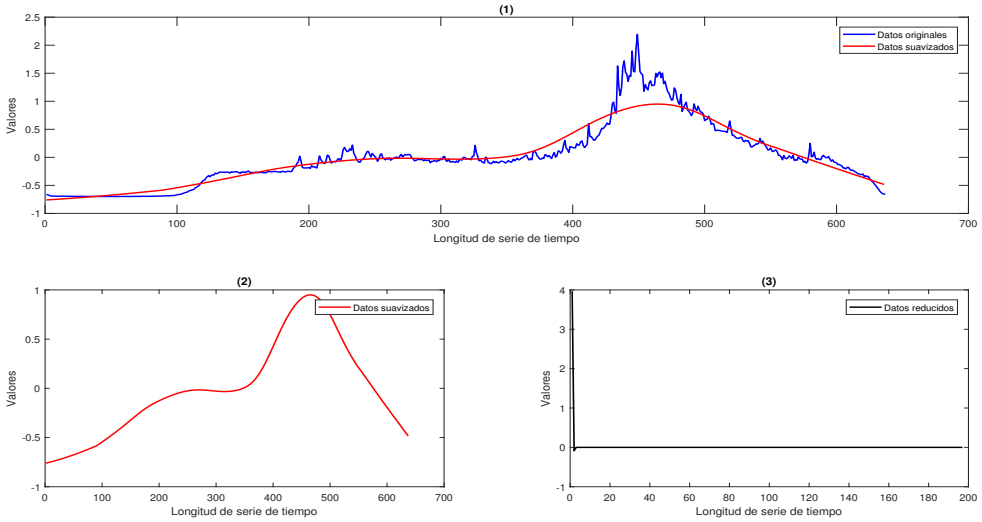


Figura 11.39: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

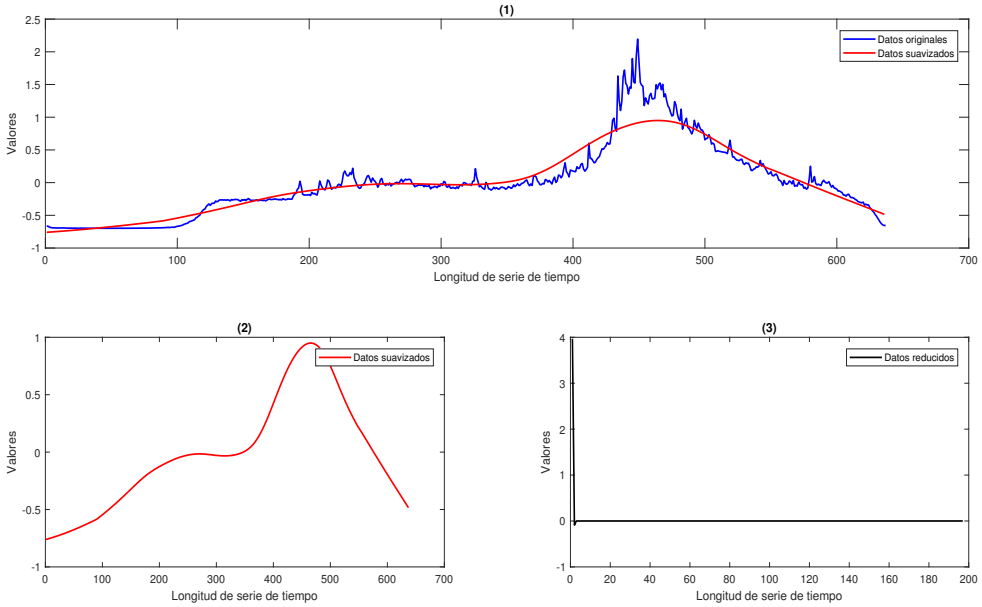


Figura 11.40: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

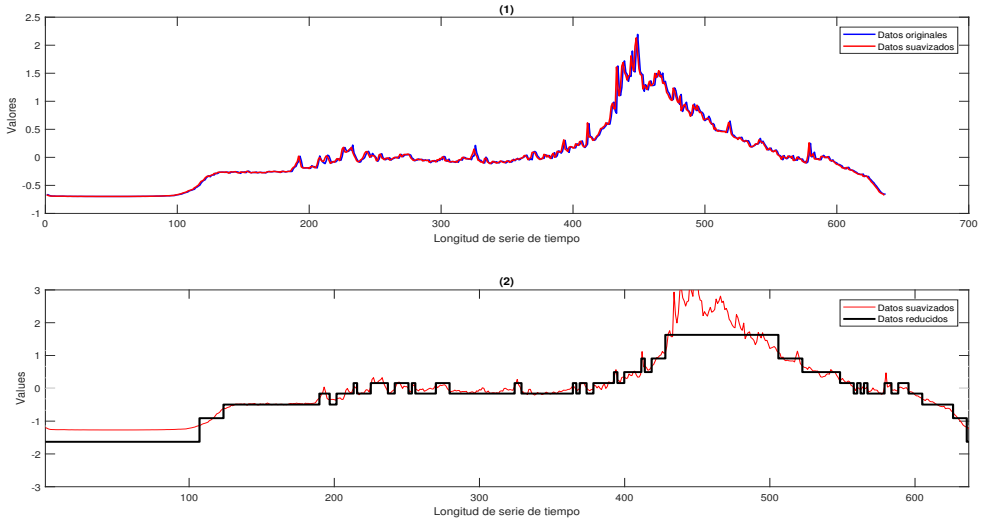


Figura 11.41: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

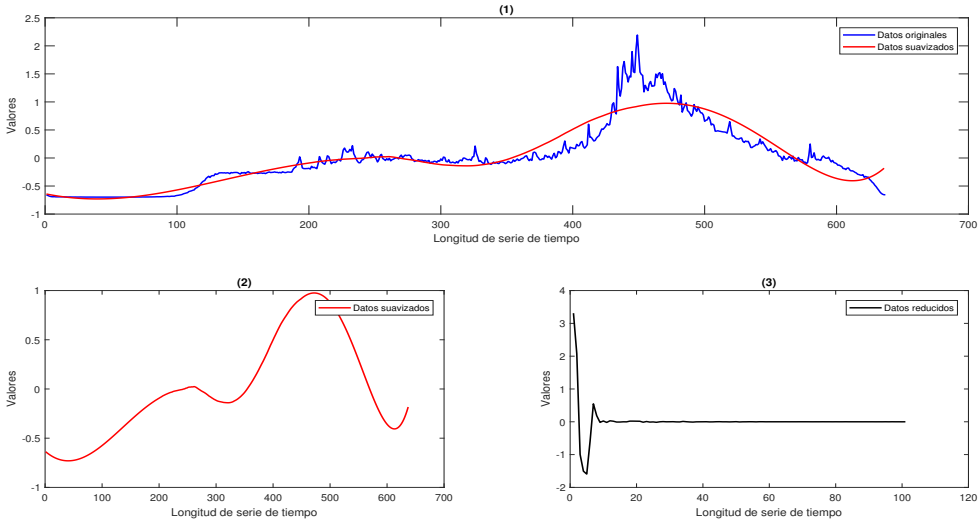


Figura 11.42: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Lightning-2. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

11.8. MODELOS DE BASE DE DATOS LIGHTNING-7

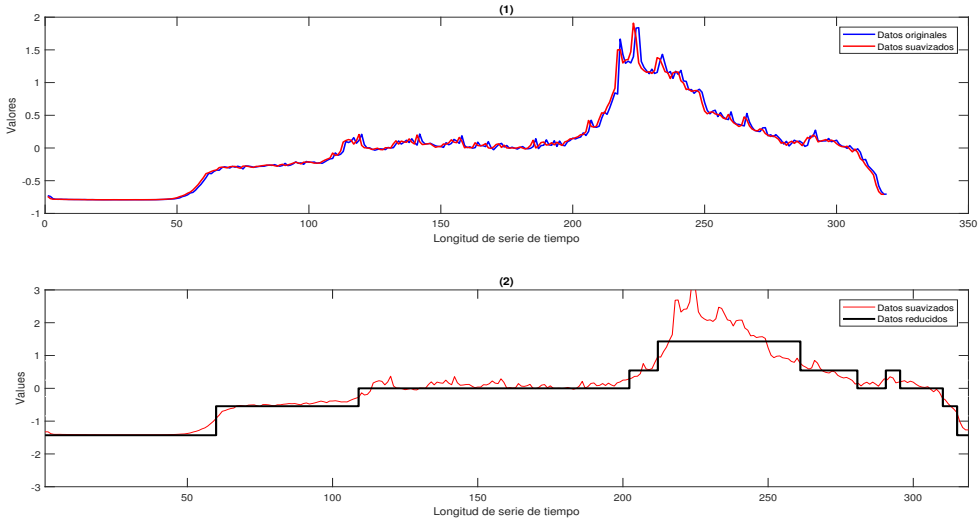


Figura 11.43: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

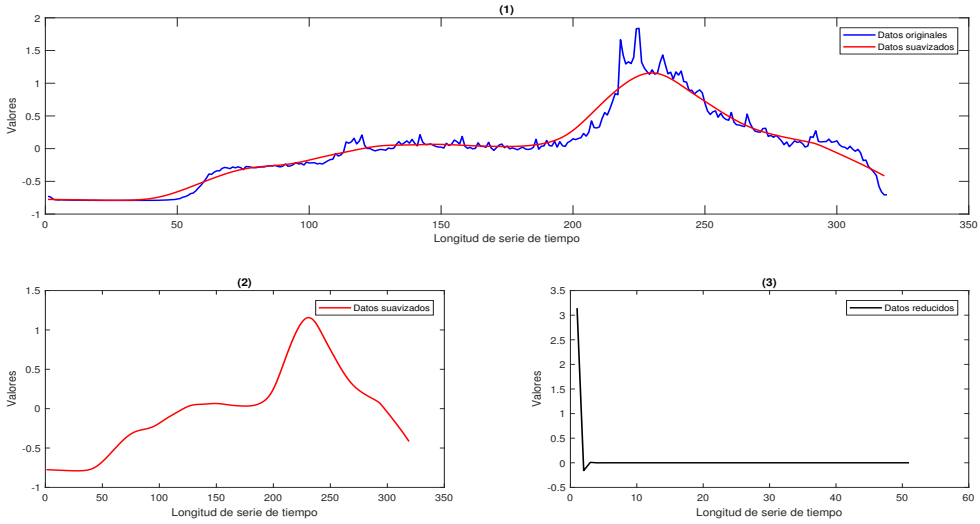


Figura 11.44: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

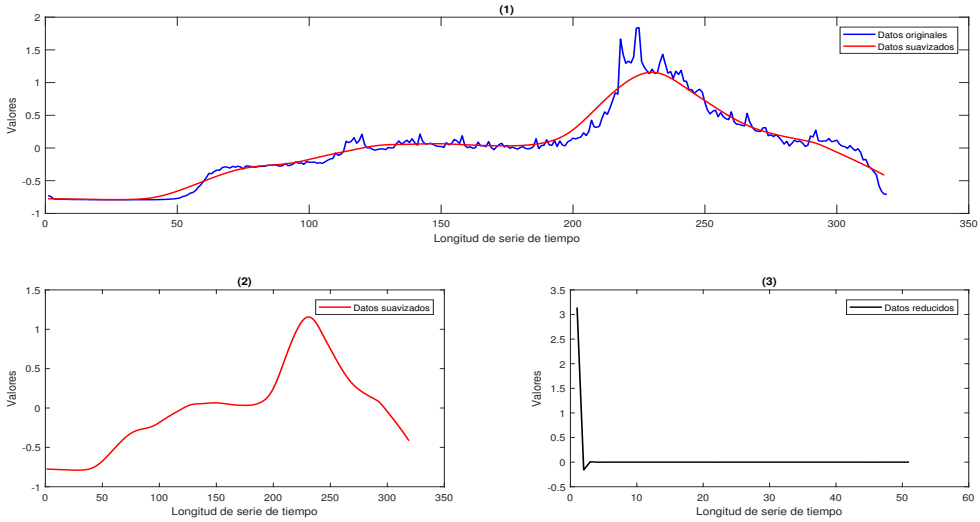


Figura 11.45: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

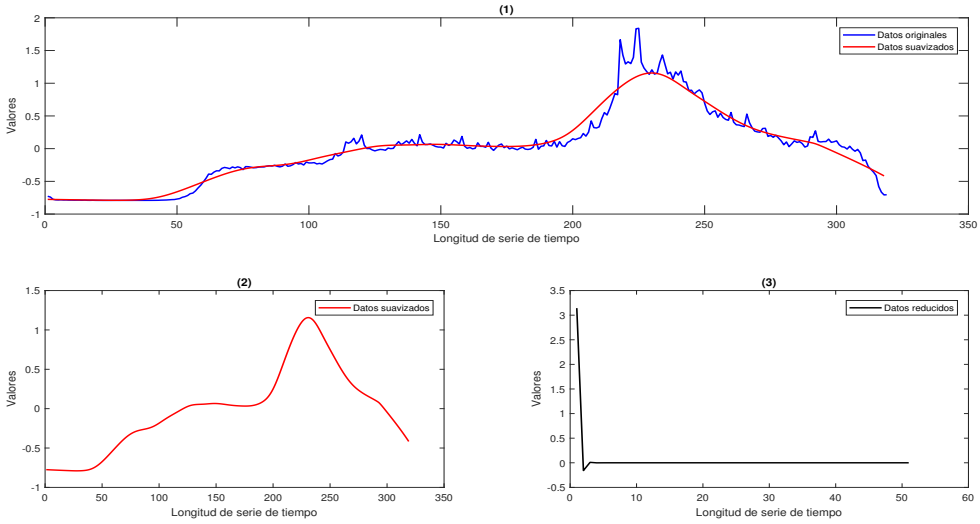


Figura 11.46: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

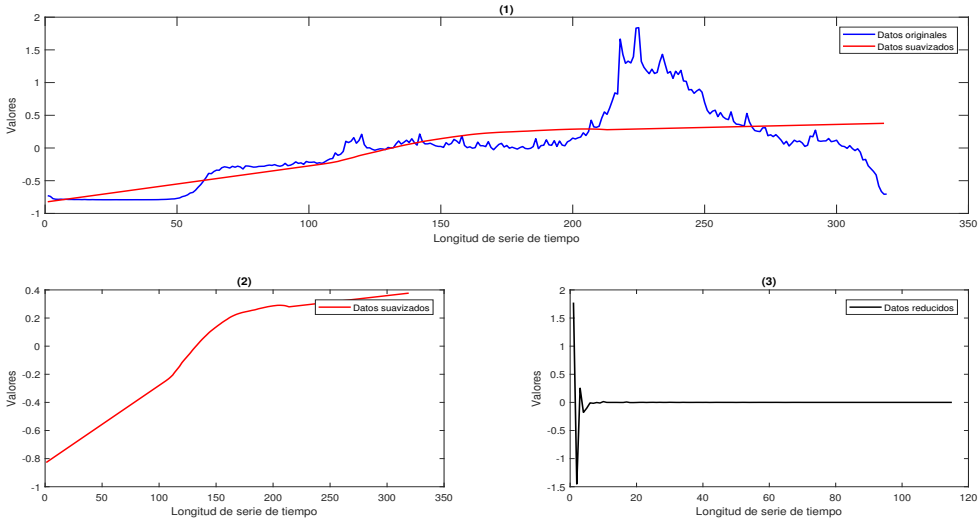


Figura 11.47: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

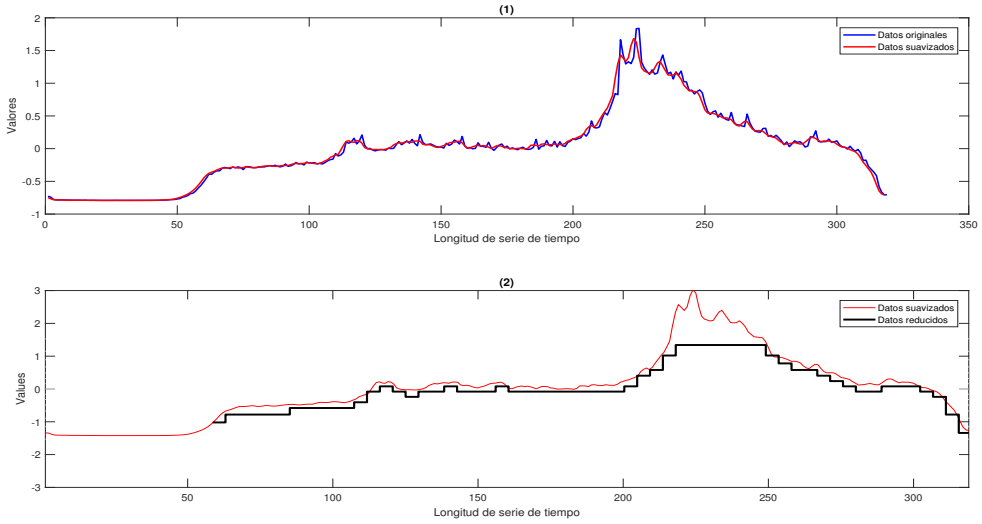


Figura 11.48: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Lightning-7. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

11.9. MODELOS DE BASE DE DATOS OLIVEOIL

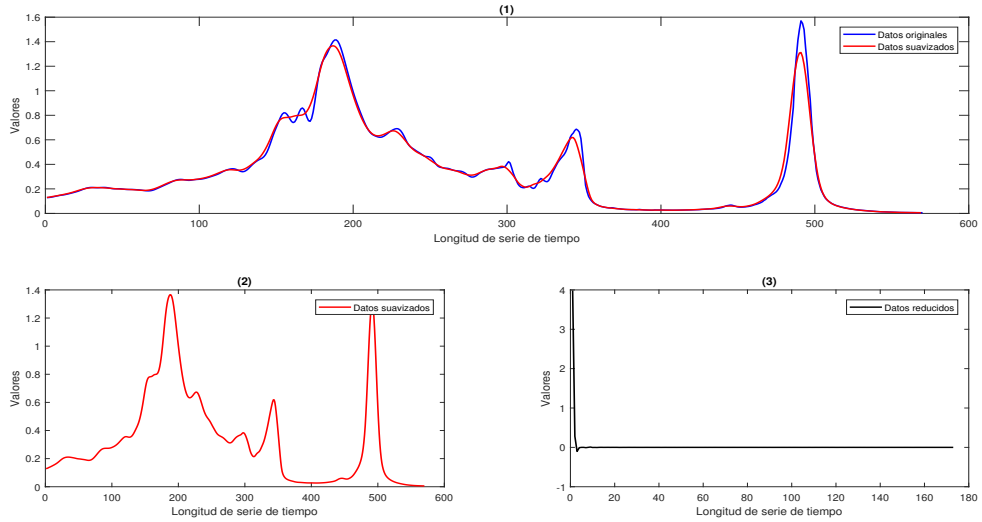


Figura 11.49: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

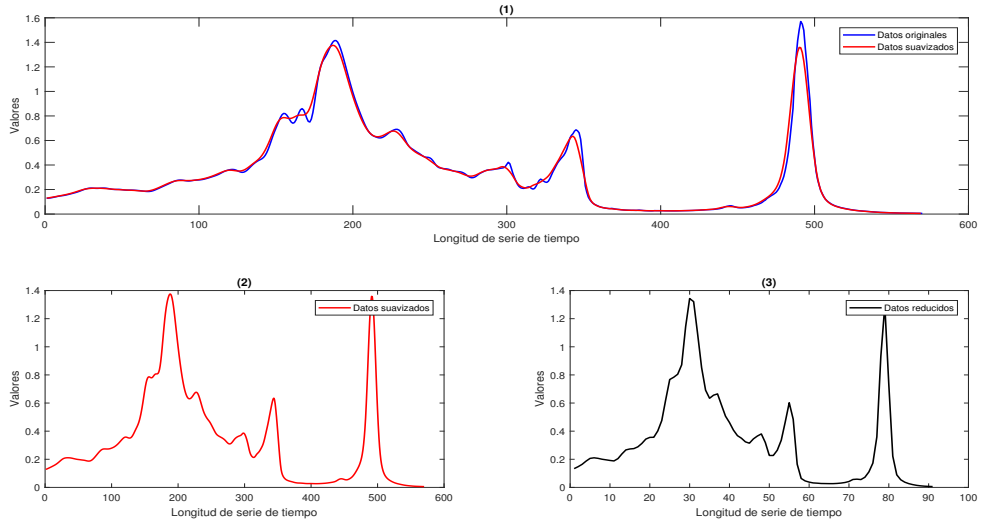


Figura 11.50: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

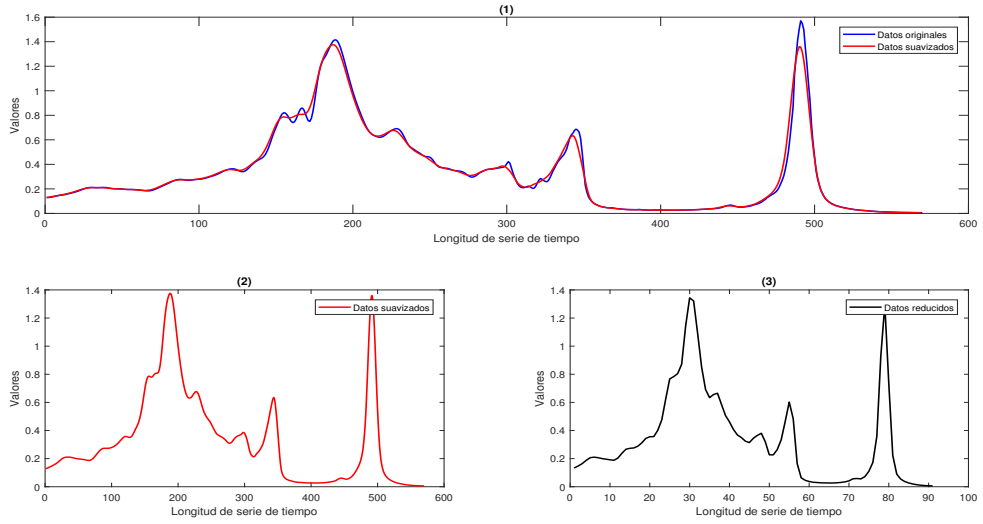


Figura 11.51: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

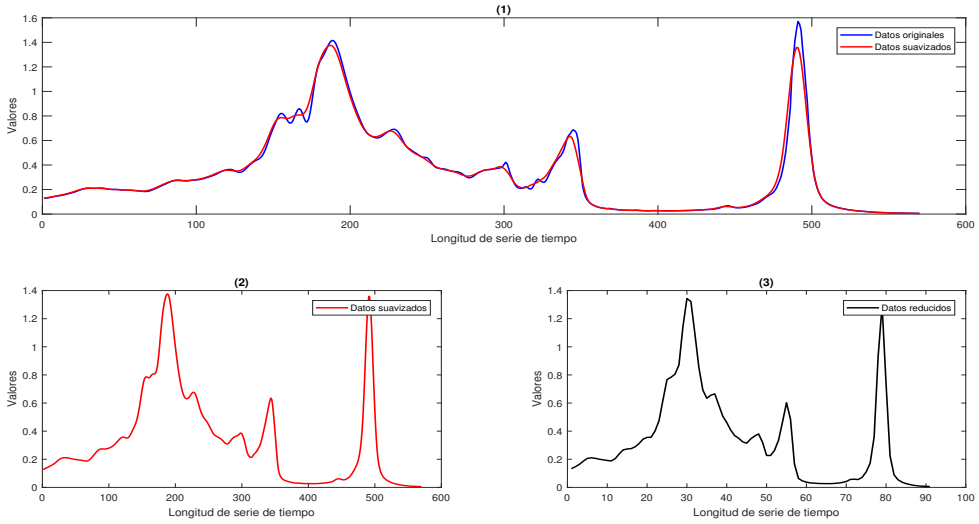


Figura 11.52: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

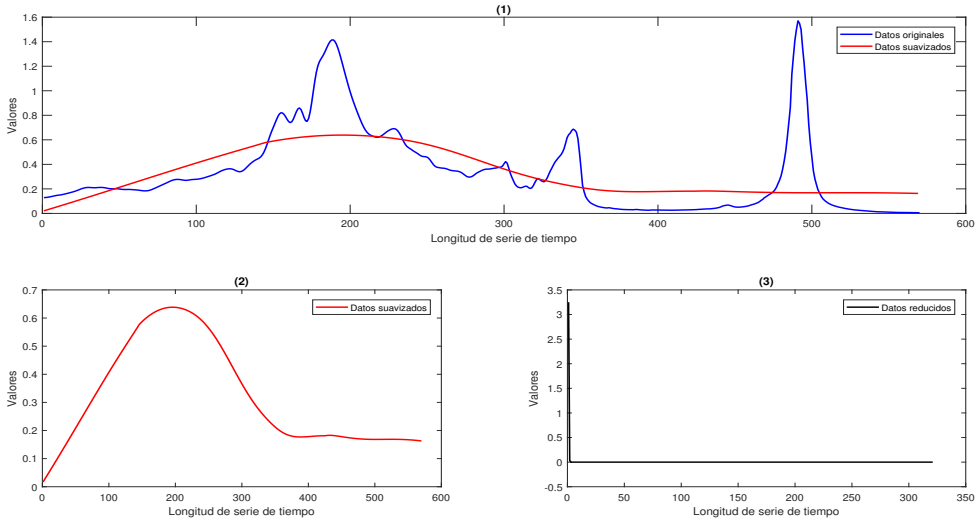


Figura 11.53: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

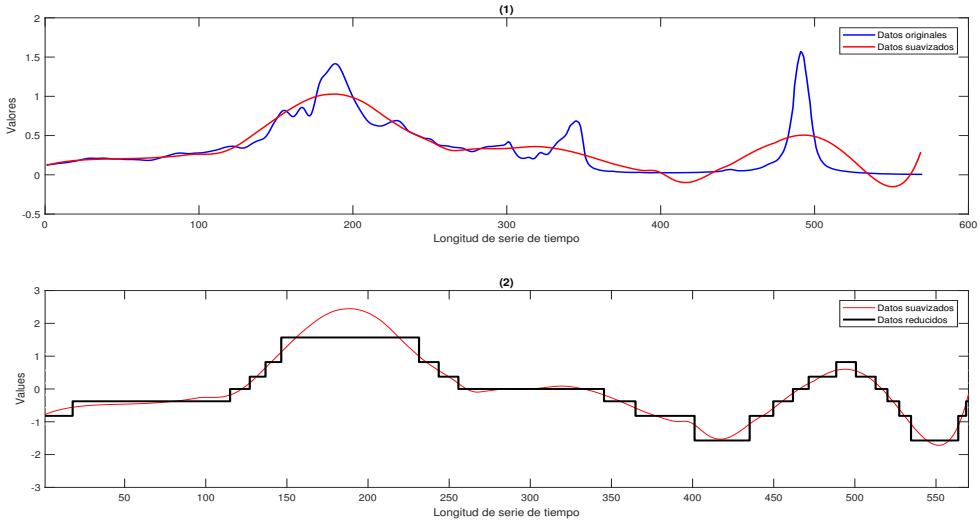


Figura 11.54: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos OliveOil. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

11.10. MODELOS DE BASE DE DATOS TRACE

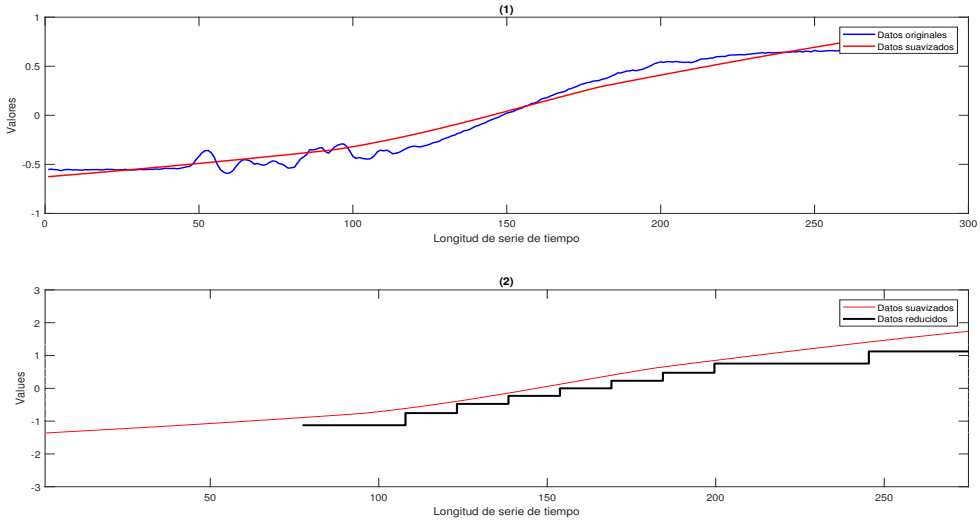


Figura 11.55: Representación gráfica del modelo sugerido por P-DEMS1 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

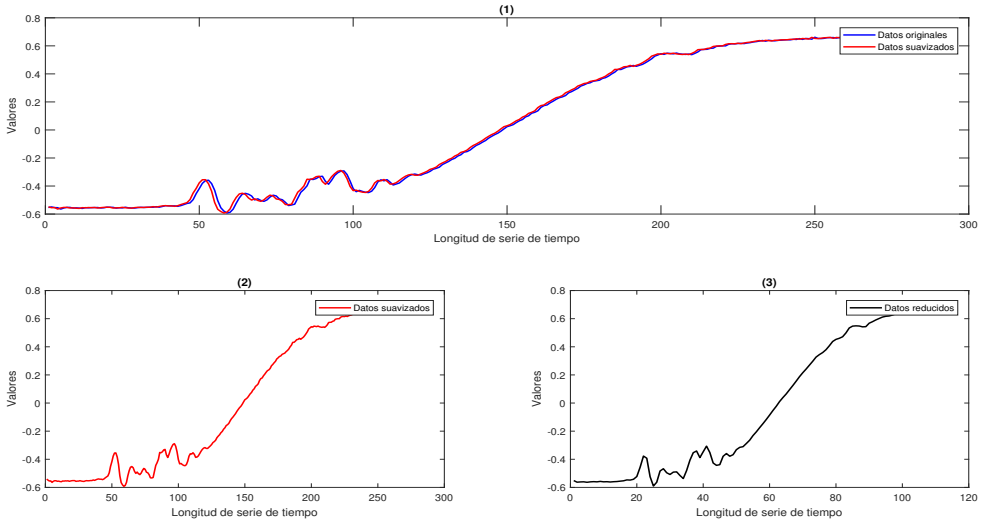


Figura 11.56: Representación gráfica del modelo sugerido por P-DEMS2 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

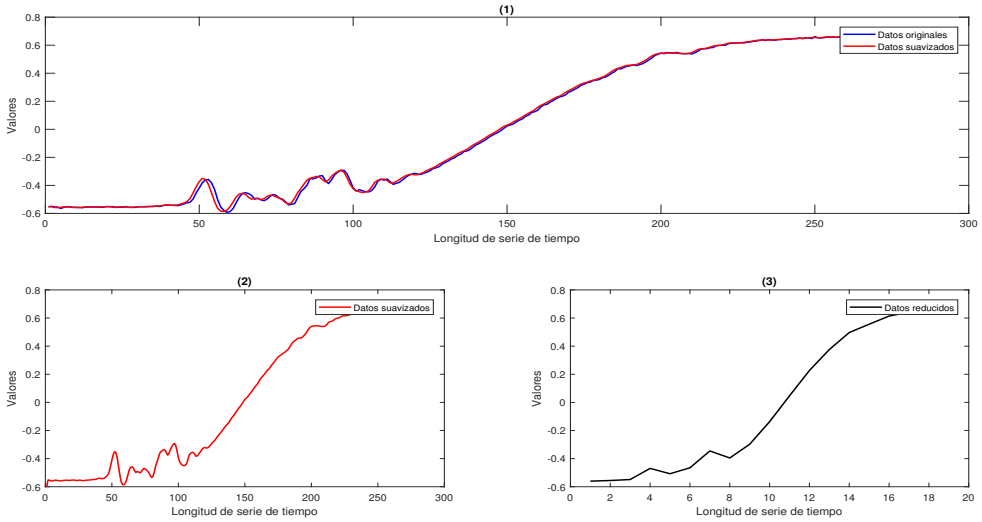


Figura 11.57: Representación gráfica del modelo sugerido por P-DEMS3 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

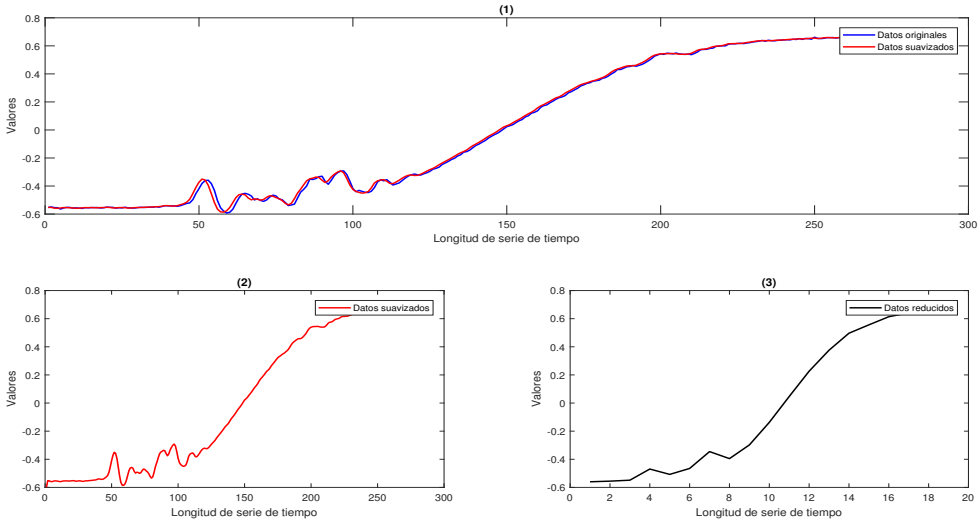


Figura 11.58: Representación gráfica del modelo sugerido por P-DEMS4 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

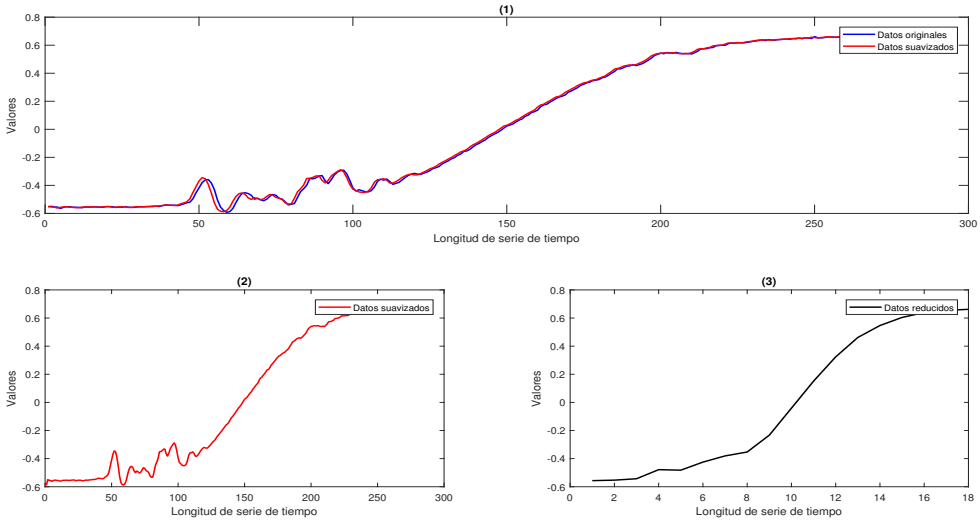


Figura 11.59: Representación gráfica del modelo sugerido por S-LSMS1 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) La media del conjunto de datos de prueba antes del cambio de representación. (3) La media del conjunto de datos de prueba después del cambio de representación y selección de instancias.

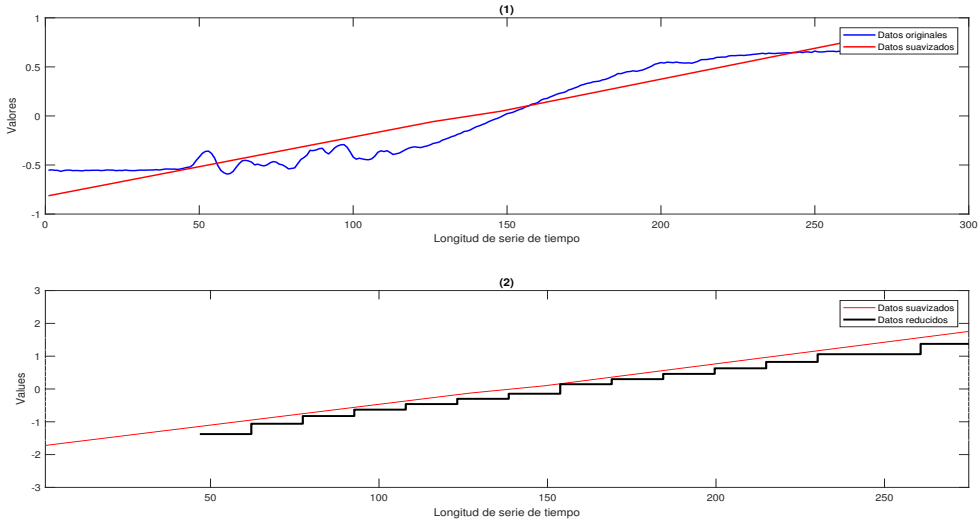


Figura 11.60: Representación gráfica del modelo sugerido por S-LSMS2 para la base de datos Trace. (1) Comparación entre la media del conjunto de datos originales de prueba y la media del conjunto de datos de prueba con los datos suavizados. (2) Comparación entre la media del conjunto de datos de prueba suavizado y la media del conjunto de datos con cambio de representación y selección de instancias.

BIBLIOGRAFÍA

[1] D. W. Aha. Generalizing from case studies: A case study. In D. Sleeman and P. Edwards, editors, *Machine Learning Proceedings 1992*, pages 1 – 10. Morgan Kaufmann, San Francisco (CA), 1992. ISBN 978-1-55860-247-2. doi: https://doi.org/10.1016/B978-1-55860-247-2.50006-1. URL https://www.sciencedirect.com/science/article/pii/B9781558602472500061.

[2] O. Al-Jowder, E. Kemsley, and R. H. Wilson. Detection of adulteration in cooked meat products by mid-infrared spectroscopy. *Journal of agricultural and food chemistry*, 50 (6):1325–1329, 2002.

[3] S. Ali and K. A. Smith-Miles. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*, 70(1):173 – 186, 2006. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2006.03.004. URL http://www.sciencedirect.com/science/article/pii/S0925231206001056. Neural Networks.

[4] A. Arias-Montano, C. A. C. Coello, and E. Mezura-Montes. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Transactions on Evolutionary Computation*, 16(5):662–694, Oct 2012. ISSN 1089-778X. doi: 10.1109/TEVC.2011.2169968.

[5] B. Arinze, S.-L. Kim, and M. Anandarajan. Combining and selecting forecasting models using rule based induction. *Computers & Operations Research*, 24(5):423 – 433, 1997. ISSN 0305-0548. doi: https://doi.org/10.1016/S0305-0548(96)00062-7. URL http://www.sciencedirect.com/science/article/pii/S0305054896000627.

[6] J. Arora. *Introduction to Optimum Design*. Elsevier Science, 2016. ISBN 9780128009185. URL https://books.google.com.mx/books?id=h-9eBwAAQBAJ.

[7] G. Atluri, A. Karpatne, and V. Kumar. Spatio-temporal data mining: A survey of problems and methods. *CoRR*, abs/1711.04710, 2017. URL http://arxiv.org/abs/1711.04710.

[8] J. M. Azevedo, R. Almeida, and P. Almeida. Using data mining with time series data in short-term stocks prediction. *International Journal of Intelligence Science*, 2(4A):177–181, 2012.

[9] K. P. Bennett, J. Hu, X. Ji, G. Kunapuli, and J.-S. Pang. Model selection via bilevel optimization. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1922–1929, 2006. doi: 10.1109/IJCNN.2006.246935.

[10] D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, Jul 2017. ISSN 1573-0565. doi: 10.1007/s10994-017-5633-9. URL https://doi.org/10.1007/s10994-017-5633-9.

[11] H.-G. Beyer and H.-P. Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.

[12] S. Bilgaiyan, S. Sagnika, and M. Das. A multi-objective cat swarm optimization algorithm for workflow scheduling in cloud computing environment. In L. C. Jain, S. Patnaik, and N. Ichalkaranje, editors, *Intelligent Computing, Communication and Devices*, pages 73–84, New Delhi, 2015. Springer India. ISBN 978-81-322-2012-1.

[13] M. Boullé. Compression-based averaging of selective naive bayes classifiers. *J. Mach. Learn. Res.*, 8:1659–1685, Dec. 2007. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1314498.1314554.

[14] I. Boussaïd, J. Lepagnot, and P. Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82 – 117, 2013. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.02.041. Prediction, Control and Diagnosis using Advanced Neural Computations.

[15] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug 1996. ISSN 1573-0565. doi: 10.1007/BF00058655. URL https://doi.org/10.1007/BF00058655.

[16] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN 9780412048418. URL https://books.google.com.mx/books?id=JwQx-WOmSyQC.

[17] R. Briandet, E. K. Kemsley, and R. H. Wilson. Discrimination of arabica and robusta in instant coffee by fourier transform infrared spectroscopy and chemometrics. *Journal of agricultural and food chemistry*, 44(1):170–174, 1996.

[18] C. E. Brodley. Addressing the selective superiority problem: Automatic algorithm/model class selection. In *Proceedings of the tenth international conference on machine learning*, pages 17–24, 1993.

[19] K. Buza, A. Nanopoulos, and L. Schmidt-Thieme. *INSIGHT: Efficient and Effective Instance Selection for Time-Series Classification*, pages 149–160. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-20847-8. doi: 10.1007/978-3-642-20847-8_13. URL https://doi.org/10.1007/978-3-642-20847-8_13.

[20] J. C. F. Cabrera and C. A. C. Coello. Micro-mopso: A multi-objective particle swarm optimizer that uses a very small population size. In N. Nedjah, L. dos Santos Coelho, and L. de Macedo Mourelle, editors, *Multi-Objective Swarm Intelligent Systems: Theory & Experiences*, pages 83–104, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-05165-4. doi: 10.1007/978-3-642-05165-4_4. URL https://doi.org/10.1007/978-3-642-05165-4_4.

[21] Y. Cai and R. Ng. Indexing spatio-temporal trajectories with chebyshev polynomials. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD Ó4, pages 599–610, New York, NY, USA, 2004. ACM. ISBN 1-58113-859-8. doi: 10.1145/1007568.1007636. URL http://doi.acm.org/10.1145/1007568.1007636.

[22] F. Camastra and A. Vinciarelli. *Machine Learning for Audio, Image and Video Analysis: Theory and Applications*. Advanced Information and Knowledge Processing. Springer London, 2007. ISBN 9781848000070. URL https://books.google.com.mx/books?id=1HmBZjoLbpYC.

[23] F. Caraffini, F. Neri, and I. Poikolainen. Micro-differential evolution with extra moves along the axes. In *2013 IEEE Symposium on Differential Evolution (SDE)*, pages 46–53, April 2013. doi: 10.1109/SDE.2013.6601441.

[24] K.-P. Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pages 126–133, Mar 1999. doi: 10.1109/ICDE.1999.754915.

[25] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[26] V. Cherkassky and F. M. Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

[27] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan. *Data mining: a knowledge discovery approach*. Springer Science & Business Media, 2007.

[28] W. S. Cleveland and S. J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.

[29] P. Compieta, S. D. Martino, M. Bertolotto, F. Ferrucci, and T. Kechadi. Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages & Computing*, 18(3):255 – 279, 2007. ISSN 1045-926X. doi: https://doi.org/10.1016/j.jvlc.2007.02.006. URL http://www.sciencedirect.com/science/article/pii/S1045926X07000134. Visual Languages and Techniques for Human-GIS Interaction.

[30] D. W. Corne, J. D. Knowles, and M. J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, pages 839–848, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-45356-7.

[31] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO'01, pages 283–290, San

Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-774-9. URL http://dl.acm.org/citation.cfm?id=2955239.2955289.

[32] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL https://doi.org/10.1007/BF00994018.

[33] I. Das. On characterizing the "knee" of the pareto curve based on normal-boundary intersection. *Structural optimization*, 18(2):107–115, Oct 1999. ISSN 1615-1488. doi: 10.1007/BF01195985.

[34] K. Deb and M. Goyal. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

[35] K. Deb and D. Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 047187339X.

[36] K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 635–642, New York, NY, USA, 2006. ACM. ISBN 1-59593-186-4. doi: 10.1145/1143997.1144112. URL http://doi.acm.org/10.1145/1143997.1144112.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002. ISSN 1089-778X. doi: 10.1109/4235.996017.

[38] K. Deb, L. Zhu, and S. Kulkarni. Handling multiple scenarios in evolutionary multi-objective numerical optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2017. ISSN 1089-778X. doi: 10.1109/TEVC.2017.2776921.

[39] J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[40] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar. A survey on new generation metaheuristic algorithms. *Computers and Industrial Engineering*, 137:106040, 2019. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2019.106040.

[41] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[42] D. R. Eads, D. Hill, S. Davis, S. J. Perkins, J. Ma, R. B. Porter, and J. P. Theiler. Genetic algorithms and support vector machines for time series classification. In *International Symposium on Optical Science and Technology*, volume 4787, pages 74–85, 2002. doi: 10.1117/12.453526.

[43] H. J. Escalante. Epsms and the document occurrence representation for authorship identification - notebook for pan at clef 2011. In *CLEF (Notebook Papers/Labs/Workshop)*. CLEF, Amsterdam, Netherlands, October 2011.

[44] H. J. Escalante, M. M. y. Gomez, and L. E. Sucar. Psms for neural networks on the ijcnn 2007 agnostic vs prior knowledge challenge. In *2007 International Joint Conference on Neural Networks*, pages 678–683, Aug 2007. doi: 10.1109/IJCNN.2007.4371038.

[45] H. J. Escalante, M. Montes, and L. E. Sucar. Particle swarm model selection. *J. Mach. Learn. Res.*, 10:405–440, June 2009. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1577069.1577084.

[46] H. J. Escalante, M. Montes, and L. Villaseñor. Particle swarm model selection for authorship verification. In E. Bayro-Corrochano and J.-O. Eklundh, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 563–570, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-10268-4.

[47] H. J. Escalante, M. Montes-y-Gómez, and L. E. Sucar. Ensemble particle swarm model selection. In *International Joint Conference on Neural Networks, IJCNN 2010, Barcelona, Spain, 18-23 July, 2010*, pages 1–8, 2010. doi: 10.1109/IJCNN.2010.5596915. URL https://doi.org/10.1109/IJCNN.2010.5596915.

[48] H. J. Escalante, M. Montes-y-Gómez, J. A. González, P. Gómez-Gil, L. A. Robles, C. A. R. García, C. Reta, and A. Rosales-Pérez. Acute leukemia classification by ensemble particle swarm model selection. *Artificial Intelligence in Medicine*, 55(3):163–175, 2012. doi: 10.1016/j.artmed.2012.03.005. URL https://doi.org/10.1016/j.artmed.2012.03.005.

[49] H. J. Escalante, M. Montes-y-Gómez, and L. E. Sucar. Multi-class particle swarm model selection for automatic image annotation. *Expert Syst. Appl.*, 39(12):11011–11021, 2012. doi: 10.1016/j.eswa.2012.03.023. URL https://doi.org/10.1016/j.eswa.2012.03.023.

[50] P. Esling and C. Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34, Dec. 2012. ISSN 0360-0300. doi: 10.1145/2379776.2379788. URL http://doi.acm.org/10.1145/2379776.2379788.

[51] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, SIGMOD ́94, pages 419–429, New York, NY, USA, 1994. ACM. ISBN 0-89791-639-5. doi: 10.1145/191839.191925. URL http://doi.acm.org/10.1145/191839.191925.

[52] J. D. Farmer and J. J. Sidorowich. Optimal shadowing and noise reduction. *Physica D: Nonlinear Phenomena*, 47(3):373 – 392, 1991. ISSN 0167-2789. doi: https://doi.org/10.1016/0167-2789(91)90037-A. URL http://www.sciencedirect.com/science/article/pii/016727899190037A.

[53] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88, 1996.

[54] L. J. Fogel. *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1999. ISBN 0-471-33250-X.

[55] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1-55860-299-2. URL http://dl.acm.org/citation.cfm?id=645513.657757.

[56] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, pages 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc. ISBN 1-55860-419-7. URL http://dl.acm.org/citation.cfm?id=3091696.3091715.

[57] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL http://www.sciencedirect.com/science/article/pii/S002200009791504X.

[58] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28 (2):337–407, 2000.

[59] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001. doi: 10.1214/aos/1013203451. URL https://doi.org/10.1214/aos/1013203451.

[60] T.-c. Fu. A review on time series data mining. *Eng. Appl. Artif. Intell.*, 24(1):164–181, Feb. 2011. ISSN 0952-1976. doi: 10.1016/j.engappai.2010.09.007. URL http://dx.doi.org/10.1016/j.engappai.2010.09.007.

[61] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044 – 2064, 2010. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2009.12.010. URL http://www.sciencedirect.com/science/article/pii/S0020025509005404. Special Issue on Intelligent Distributed Information Systems.

[62] M. S. Ghaemi, B. Agard, V. P. Nia, and M. Trépanier. Challenges in spatial-temporal data analysis targeting public transportÕ. *IFAC-PapersOnLine*, 48(3):442 – 447, 2015. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2015.06.121. URL http://www.sciencedirect.com/science/article/pii/S2405896315003602. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[63] F. Giannotti and D. Pedreschi. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Mobility, Data Mining and Privacy. Springer Science & Business Media, 2008. ISBN 978-3-540-75176-2.

[64] C. Giraud-Carrier, P. Flach, Y. Peng, J. Farrand, and H. Bensusan. Metal: A meta-learning assistant for providing user support in machine learning and data mining. *Accessed on 20th July*, 2013.

[65] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. ISBN 0201157675.

[66] J. L. Guiñón, E. Ortega, J. García-Antón, and V. Pérez-Herranz. Moving average and savitzki-golay smoothing filters using mathcad. In *International Conference on Engineering Education*, volume 1, pages 1–4, 2007.

[67] I. Guyon, A. Saffari, G. Dror, and G. Cawley. Model selection: Beyond the bayesian/frequentist divide. *J. Mach. Learn. Res.*, 11:61–87, Mar. 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1756009.

[68] J. Han, J. Pei, and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011. ISBN 9780123814807. URL https://books.google.com.mx/books?id=pQws07tdpjoC.

[69] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2001. ISBN 9780387952840.

[70] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0262082136.

[71] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87 vol.1, Jun 1994. doi: 10.1109/ICEC.1994.350037.

[72] C. W. Hsu, C. C. Chang, and C. J. Lin. *A practical guide to support vector classification*. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2003. URL http://www.csie.ntu.edu.tw/~{}cjlin/papers/guide/guide.pdf.

[73] B. Hu, Y. Chen, and E. J. Keogh. Classification of streaming time series under more realistic assumptions. *Data Min. Knowl. Discov.*, 30(2):403–437, 2016. doi: 10.1007/s10618-015-0415-0. URL https://doi.org/10.1007/s10618-015-0415-0.

[74] T. Huang and A. S. Mohan. Micro-particle swarm optimizer for solving high dimensional optimization problems (μpso for high dimensional optimization problems). *Applied Mathematics and Computation*, 181(2):1148 – 1154, 2006. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2006.01.088. URL http://www.sciencedirect.com/science/article/pii/S0096300306001998.

[75] J. W. Hunt and T. G. Szymanski. A fast algorithm for computing longest common subsequences. *Commun. ACM*, 20(5):350–353, May 1977. ISSN 0001-0782. doi: 10.1145/359581.359603. URL http://doi.acm.org/10.1145/359581.359603.

[76] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72, Feb 1975. ISSN 0096-3518. doi: 10.1109/TASSP.1975.1162641.

[77] A. Kalousis and T. Theoharis. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*, 3(5):319 – 337, 1999. ISSN 1088-467X. doi: https://doi.org/10.1016/S1088-467X(99)00026-8. URL http://www.sciencedirect.com/science/article/pii/S1088467X99000268.

[78] M. Kantardzic. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[79] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[80] J. Kennedy. Particle swarm optimization. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 760–766, Boston, MA, 2010. Springer US. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_630. URL https://doi.org/10.1007/978-0-387-30164-8_630.

[81] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, Mar. 2005. ISSN 0219-1377. doi: 10.1007/s10115-004-0154-9. URL http://dx.doi.org/10.1007/s10115-004-0154-9.

[82] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, Aug 2001. ISSN 0219-1377. doi: 10.1007/PL00011669. URL https://doi.org/10.1007/PL00011669.

[83] E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 206–215, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014077. URL http://doi.acm.org/10.1145/1014052.1014077.

[84] R. D. King, C. Feng, and A. Sutherland. Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal*, 9 (3):289–333, 1995.

[85] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172, June 2000. ISSN 1063-6560. doi: 10.1162/106365600568167. URL http://dx.doi.org/10.1162/106365600568167.

[86] M. Koppen, K. Franke, and R. Vicente-Garcia. Tiny gas for image processing applications. *IEEE Computational Intelligence Magazine*, 1(2):17–26, May 2006. ISSN 1556-603X. doi: 10.1109/MCI.2006.1626491.

[87] M. Kordos and K. Łapa. Multi-objective evolutionary instance selection for regression tasks. *Entropy*, 20(10):746, 2018.

[88] E. J. Kostelich and J. A. Yorke. Noise reduction in dynamical systems. *Phys. Rev. A*, 38: 1649–1652, Aug 1988. doi: 10.1103/PhysRevA.38.1649. URL https://link.aps.org/doi/10.1103/PhysRevA.38.1649.

[89] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[90] G. Kunapuli, K. Bennett, J. Hu, and J.-S. Pang. Bilevel model selection for support vector machines. *Manuscript, Department of Mathematical Sciences, Rensselaer Polytechnic Institute (March 2007)*, 2008.

[91] M. Last, A. Kandel, and H. Bunke. *Data mining in time series databases*, volume 57. World scientific, 2004.

[92] A. Letchford, J. Gao, and L. Zheng. *Penalized Least Squares for Smoothing Financial Time Series*, pages 72–81. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-25832-9. doi: 10.1007/978-3-642-25832-9_8. URL https://doi.org/10.1007/978-3-642-25832-9_8.

[93] Z.-X. Li, J.-S. Guo, X.-B. Hui, and F.-F. Song. Dimension reduction method for multivariate time series based on common principal component. *Control and Decision*, 28(4): 531–536, 2013.

[94] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3):203–228, Sep 2000. ISSN 1573-0565. doi: 10.1023/A:1007608224229. URL https://doi.org/10.1023/A:1007608224229.

[95] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, Oct 2007. ISSN 1573-756X. doi: 10.1007/s10618-007-0064-z. URL https://doi.org/10.1007/s10618-007-0064-z.

[96] C. Loader. Smoothing: local regression techniques. In *Handbook of Computational Statistics*, pages 571–596. Springer, 2012.

[97] F. Lobato, M. Sousa, M. Silva, and A. Machado. Multi-objective optimization and bio-inspired methods applied to machinability of stainless steel. *Applied Soft Computing*, 22:261 – 271, 2014. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2014.05.004. URL http://www.sciencedirect.com/science/article/pii/S1568494614002300.

[98] J. Luo, Q. Liu, Y. Yang, X. Li, M. rong Chen, and W. Cao. An artificial bee colony algorithm for multi-objective optimisation. *Applied Soft Computing*, 50:235 – 251, 2017. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2016.11.014. URL http://www.sciencedirect.com/science/article/pii/S156849461630583X.

[99] W. Martinez, A. Martinez, A. Martinez, and J. Solka. *Exploratory Data Analysis with MATLAB, Second Edition*. Chapman & Hall/CRC Computer Science & Data Analysis. CRC Press, 2010. ISBN 9781439812211. URL https://books.google.com.mx/books?id=_J3MBQAAQBAJ.

[100] E. n. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello. A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 485–492, New York, NY, USA, 2006. ACM. ISBN 1-59593-186-4. doi: 10.1145/1143997.1144086. URL http://doi.acm.org/10.1145/1143997.1144086.

[101] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: interactive approaches. In *Multiobjective Optimization*, pages 27–57. Springer, 2008.

[102] M. M. Millonas. Swarms, phase transitions, and collective intelligence. *arXiv preprint adap-org/9306002*, 1993.

[103] S. Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4):1053–1073, May 2016. ISSN 1433-3058. doi: 10.1007/s00521-015-1920-1. URL https://doi.org/10.1007/s00521-015-1920-1.

[104] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. dos S. Coelho. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, 47:106 – 119, 2016. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2015.10.039. URL http://www.sciencedirect.com/science/article/pii/S0957417415007435.

[105] S. Mirjalili, P. Jangir, and S. Saremi. Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence*, 46(1): 79–95, Jan 2017. ISSN 1573-7497. doi: 10.1007/s10489-016-0825-8. URL https://doi.org/10.1007/s10489-016-0825-8.

[106] B. S. P. Mishra, S. Dehuri, and S.-B. Cho. Swarm intelligence in multiple and many objectives optimization: A survey and topical study on eeg signal analysis. In S. Dehuri, A. K. Jagadev, and M. Panda, editors, *Multi-objective Swarm Intelligence: Theoretical Advances and Applications*, pages 27–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-662-46309-3. doi: 10.1007/978-3-662-46309-3_2. URL https://doi.org/10.1007/978-3-662-46309-3_2.

[107] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[108] T. Mitsa. *Temporal data mining*. CRC Press, 2010.

[109] Á. Moreno, F. J. García-Haro, B. Martínez, and M. A. Gilabert. Noise reduction and gap filling of fapar time series using an adapted local regression filter. *Remote Sensing*, 6(9): 8238–8260, 2014.

[110] R. W. Morrison and K. A. De Jong. Measurement of population diversity. In P. Collet, C. Fonlupt, J.-K. Hao, E. Lutton, and M. Schoenauer, editors, *Artificial Evolution*, pages 31–41, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-46033-6.

[111] M. Müller. Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74048-3. doi: 10.1007/978-3-540-74048-3_4. URL https://doi.org/10.1007/978-3-540-74048-3_4.

[112] S. Nahmias and Y. Cheng. *Production and operations analysis*, volume 4. McGraw-Hill/Irwin New York, 2009.

[113] F. Neri and V. Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33(1):61–106, Feb 2010. ISSN 1573-7462. doi: 10.1007/s10462-009-9137-2. URL https://doi.org/10.1007/s10462-009-9137-2.

[114] V. Niennattrakul, E. Keogh, and C. A. Ratanamahatana. Data editing techniques to allow the application of distance-based outlier detection to streams. In *2010 IEEE International Conference on Data Mining*, pages 947–952, Dec 2010. doi: 10.1109/ICDM.2010.56.

[115] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2000. ISBN 9780387987934. URL https://books.google.com.mx/books?id=epc5fX0lqRIC.

[116] D. E. O'Leary. Enterprise knowledge management. *Computer*, 31(3):54–61, 1998.

[117] R. T. Olszewski. Generalized feature extraction for structural pattern recognition in time-series data. Technical report, DTIC Document, 2001.

[118] I. H. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5):511–623, Oct 1996. ISSN 1572-9338. doi: 10.1007/BF02125421. URL https://doi.org/10.1007/BF02125421.

[119] S. Ougiaroglou, L. Karamitopoulos, C. Tatoglou, G. Evangelidis, and D. A. Dervos. Applying prototype selection and abstraction algorithms for efficient time-series classification. In P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, editors, *Artificial Neural Networks: Methods and Applications in Bio-/Neuroinformatics*, pages 333–348. Springer International Publishing, Cham, 2015. ISBN 978-3-319-09903-3. doi: 10.1007/978-3-319-09903-3_16. URL https://doi.org/10.1007/978-3-319-09903-3_16.

[120] H. Ouyang, G. Ma, G. Liu, Z. Li, and X. Zhong. Hybrid teaching-learning based optimization with harmony search for engineering optimization problems. In *2017 36th Chinese Control Conference (CCC)*, pages 2714–2717, July 2017. doi: 10.23919/ChiCC.2017.8027774.

[121] K. E. Parsopoulos. Cooperative micro-differential evolution for high-dimensional problems. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary*

Computation, GECCO '09, pages 531–538, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1569975. URL http://doi.acm.org/10.1145/1569901.1569975.

[122] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems*, 22(3):52–67, 2002.

[123] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. J. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowl. Inf. Syst.*, 47(1):1–26, 2016. doi: 10.1007/s10115-015-0878-8. URL https://doi.org/10.1007/s10115-015-0878-8.

[124] B. Pfahringer, H. Bensusan, and C. G. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 743–750, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2. URL http://dl.acm.org/citation.cfm?id=645529.658105.

[125] E. A. Portilla-Flores, A. Sánchez-Márquez, L. Flores-Pulido, E. Vega-Alvarado, M. B. C. Yáñez, J. A. Aponte-Rodríguez, and P. A. Niño-Suárez. Enhancing the harmony search algorithm performance on constrained numerical optimization. *IEEE Access*, 5:25759–25780, 2017. doi: 10.1109/ACCESS.2017.2771741.

[126] N. Pérez-Castro, H. G. Acosta-Mesa, E. Mezura-Montes, and N. Cruz-Ramírez. Towards the full model selection in temporal databases by using micro-differential evolution. an empirical study. In *2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6, Nov 2015. doi: 10.1109/ROPEC.2015.7395161.

[127] N. Pérez-Castro, A. Márquez-Grajales, H. G. Acosta-Mesa, and E. Mezura-Montes. Full model selection issue in temporal data through evolutionary algorithms: A brief review. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2451–2457, June 2017. doi: 10.1109/CEC.2017.7969602.

[128] K. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[129] A. L. Prodromidis, P. K. Chan, and S. J. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. In *Advances of Distributed Data Mining*. AAAI Press, 2000.

[130] F. Provost and T. Fawcett. Data science and its relationship to big data and data-driven decision making. *Big Data*, 1(1):51–59, 2013.

[131] R. B. Prudêncio and T. B. Ludermir. Meta-learning approaches to selecting time series models. *Neurocomputing*, 61:121–137, 2004.

[132] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986. ISSN 1573-0565. doi: 10.1007/BF00116251. URL https://doi.org/10.1007/BF00116251.

[133] J. Rada-Vilela, M. Chica, Ã“scar CordÃ³n, and S. Damas. A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem. *Applied Soft Computing*, 13(11):4370 – 4382, 2013. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2013.06.014. URL http://www.sciencedirect.com/science/article/pii/S1568494613002032.

[134] S. Rahnamayan and H. R. Tizhoosh. Image thresholding using micro opposition-based differential evolution (micro-ode). In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 1409–1416, June 2008. doi: 10.1109/CEC.2008.4630979.

[135] C. A. Ralanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das. Mining time series data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 1069–1103. Springer US, Boston, MA, 2005. ISBN 978-0-387-25465-4. doi: 10.1007/0-387-25465-X_51. URL https://doi.org/10.1007/0-387-25465-X_51.

[136] C. Ramirez-Atencia, S. Mostaghim, and D. Camacho. A knee point based evolutionary multi-objective optimization for mission planning problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 1216–1223, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4920-8. doi: 10.1145/3071178.3071319. URL http://doi.acm.org/10.1145/3071178.3071319.

[137] C. A. Ratanamahatana and E. Keogh. Making time-series classification more accurate using learned constraints. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 11–22. SIAM, 2004.

[138] C. A. Ratanamahatana and E. Keogh. Three myths about dynamic time warping data mining. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 506–510. SIAM, 2005.

[139] S. Rathee and S. Ratnoo. Feature selection using multi-objective chc genetic algorithm. *Procedia Computer Science*, 167:1656 – 1664, 2020. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2020.03.376. URL http://www.sciencedirect.com/science/article/pii/S1877050920308425. International Conference on Computational Intelligence and Data Science.

[140] F. Rechy-Ramírez. Discretización de series de tiempo usando programación evolutiva con función multiobjetivo. Master's thesis, Universidad Veracruzana, 2010.

[141] P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer, 2009.

[142] L. Rendell and H. Cho. Empirical learning as a function of concept character. *Machine Learning*, 5(3):267–298, Aug 1990. ISSN 1573-0565. doi: 10.1007/BF00117106. URL https://doi.org/10.1007/BF00117106.

[143] M. Reyes-Sierra, C. C. Coello, et al. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2 (3):287–308, 2006.

[144] J. R. Rice. The algorithm selection problem**this work was partially supported by the national science foundation through grant gp-32940x. this chapter was presented as the george e. forsythe memorial lecture at the computer science conference, february 19, 1975, washington, d. c. volume 15 of *Advances in Computers*, pages 65 – 118. Elsevier, 1976. doi: https://doi.org/10.1016/S0065-2458(08)60520-3. URL http://www.sciencedirect.com/science/article/pii/S0065245808605203.

[145] J. Ronkkonen, S. Kukkonen, and K. V. Price. Real-parameter optimization with differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 506–513 Vol.1, Sept 2005. doi: 10.1109/CEC.2005.1554725.

[146] A. Rosales-Pérez, H. J. Escalante, J. A. Gonzalez, C. A. R. García, and C. A. C. Coello. Bias and variance multi-objective optimization for support vector machines model selection. In J. M. Sanches, L. Micó, and J. S. Cardoso, editors, *Pattern Recognition and Image Analysis - 6th Iberian Conference, IbPRIA 2013, Funchal, Madeira, Portugal, June 5-7, 2013. Proceedings*, volume 7887 of *Lecture Notes in Computer Science*, pages 108–116. Springer, 2013. doi: 10.1007/978-3-642-38628-2_12. URL https://doi.org/10.1007/978-3-642-38628-2_12.

[147] A. Rosales-Pérez, J. A. Gonzalez, C. A. C. Coello, H. J. Escalante, and C. A. Reyes-Garcia. Multi-objective model type selection. *Neurocomputing*, 146:83 – 94, 2014. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.05.077. URL http://www.sciencedirect.com/science/article/pii/S0925231214008789. Bridging Machine learning and Evolutionary Computation (BMLEC) Computational Collective Intelligence.

[148] D. Roverso. Multivariate temporal classification by windowed wavelet decomposition and recurrent neural networks. In *3rd ANS international topical meeting on nuclear plant instrumentation, control and human-machine interface*, volume 20, pages 527–538, 2000.

[149] T. P. Runarsson and Xin Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, 2000.

[150] D. Ruta and B. Gabrys. Classifier selection for majority voting. *Information Fusion*, 6(1):63 – 81, 2005. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2004.04.008. URL http://www.sciencedirect.com/science/article/pii/S1566253504000417. Diversity in Multiple Classifier Systems.

[151] N. Saito. Local feature extraction and its applications using a library of bases. *Topics in Analysis and Its Applications: Selected Theses*, pages 269–451, 2000.

[152] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26 (1):43–49, Feb 1978. ISSN 0096-3518. doi: 10.1109/TASSP.1978.1163055.

[153] S. L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240, Sep 1994. ISSN 1573-0565. doi: 10.1007/BF00993309. URL https://doi.org/10.1007/BF00993309.

[154] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[155] G. Scepi. Clustering algorithms for large temporal data sets. In *Data Analysis and Classification*, pages 369–377. Springer, 2010.

[156] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc. ISBN 0-8058-0426-9. URL http://dl.acm.org/citation.cfm?id=645511.657079.

[157] R. E. Schapire. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, pages 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1624312.1624417.

[158] A. Seshadri. Nsga - ii: A multi-objective optimization algorithm, 2009. https://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm.

[159] Z. Shang, E. Zgraggen, B. Buratti, F. Kossmann, P. Eichmann, Y. Chung, C. Binnig, E. Upfal, and T. Kraska. Democratizing data science through interactive curation of ml pipelines. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 1171–1188, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450356435. doi: 10.1145/3299869.3319863. URL https://doi.org/10.1145/3299869.3319863.

[160] S. Sharma. *Metric Space*. DPH mathematics series. Discovery Publishing House, 2006. ISBN 9788183561181. URL https://books.google.com.mx/books?id=L_A3HbGYQ0cC.

[161] J. Shlens. A tutorial on principal component analysis. In *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*, 2005.

[162] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In W. W. Cohen and H. Hirsh, editors, *Machine Learning Proceedings 1994*, pages 293 – 301. Morgan Kaufmann, San Francisco (CA), 1994. ISBN 978-1-55860-335-6. doi: https://doi.org/10.1016/B978-1-55860-335-6.50043-X. URL https://www.sciencedirect.com/science/article/pii/B978155860335650043X.

[163] K. A. Smith, F. Woo, V. Ciesielski, and R. Ibrahim. Matching data mining algorithm suitability to data characteristics using a self-organizing map. In A. Abraham and M. Köppen, editors, *Hybrid Information Systems*, pages 169–179, Heidelberg, 2002. Physica-Verlag HD. ISBN 978-3-7908-1782-9.

[164] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.*, 2(3):221–248, Sept. 1994. ISSN 1063-6560. doi: 10.1162/evco.1994.2.3.221. URL http://dx.doi.org/10.1162/evco.1994.2.3.221.

[165] N. Stehr. Knowledge societies. In *The Wiley-Blackwell Encyclopedia of Globalization*. John Wiley & Sons, Ltd, 2012. ISBN 9780470670590. doi: 10.1002/9780470670590.wbeog342. URL http://dx.doi.org/10.1002/9780470670590.wbeog342.

[166] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, Dec 1997. ISSN 1573-2916. doi: 10.1023/A:1008202821328. URL https://doi.org/10.1023/A:1008202821328.

[167] S. Sumathi and S. Sivanandam. *Introduction to Data Mining and its Applications*. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2006. ISBN 9783540343516. URL http://www.springer.com/gp/book/9783540343509.

[168] Q. Sun, B. Pfahringer, and M. Mayo. Towards a framework for designing full model selection and optimization systems. In Z.-H. Zhou, F. Roli, and J. Kittler, editors, *Multiple Classifier Systems*, pages 259–270, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-38067-9.

[169] E. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009. ISBN 9780470496909.

[170] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367.

[171] H. S. Tapp, M. Defernez, and E. K. Kemsley. Ftir spectroscopy and multivariate analysis can distinguish the geographic origin of extra virgin olive oils. *Journal of agricultural and food chemistry*, 51(21):6110–6115, 2003.

[172] T. Tharakeshwar, K. Seetharamu, and B. D. Prasad. Multi-objective optimization using bat algorithm for shell and tube heat exchangers. *Applied Thermal Engineering*, 110:1029 – 1038, 2017. ISSN 1359-4311. doi: https://doi.org/10.1016/j.applthermaleng.2016.09.031. URL http://www.sciencedirect.com/science/article/pii/S1359431116316271.

[173] I. Triguero, J. Derrac, S. Garcia, and F. Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1):86–100, Jan 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2010.2103939.

[174] P. Tyndale. A taxonomy of knowledge management software tools: origins and applications. *Evaluation and Program Planning*, 25(2):183 – 190, 2002. ISSN 0149-7189. doi: https://doi.org/10.1016/S0149-7189(02)00012-5. URL http://www.sciencedirect.com/science/article/pii/S0149718902000125.

[175] d. A. Universidad. Signal smoothing algorithms. URL http://195.134.76.37/applets/AppletSmooth/Appl_Smooth2.html. Acceso. 30.09.2017.

[176] A. Venkatachalam and J. E. Sohl. An intelligent model selection and forecasting system. *Journal of Forecasting*, 18(3):167–180, 1999.

[177] Vikas and S. J. Nanda. Multi-objective moth flame optimization. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2470–2476, Sept 2016. doi: 10.1109/ICACCI.2016.7732428.

[178] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, Jun 2002. ISSN 1573-7462. doi: 10.1023/A:1019956318069. URL https://doi.org/10.1023/A:1019956318069.

[179] F. Viveros Jiménez, E. Mezura Montes, and A. Gelbukh. Empirical analysis of a micro-evolutionary algorithm for numerical optimization. *Int. J. Phys. Sci*, 7:1235–1258, 2012.

[180] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, Mar 2013. ISSN 1573-756X. doi: 10.1007/s10618-012-0250-5. URL https://doi.org/10.1007/s10618-012-0250-5.

[181] L. Wasserman. Bayesian model selection and model averaging. *Journal of Mathematical Psychology*, 44(1):92 – 107, 2000. ISSN 0022-2496. doi: https://doi.org/10.1006/jmps.1999.1278. URL http://www.sciencedirect.com/science/article/pii/S0022249699912786.

[182] K. Watanabe and M. M. A. Hashem. *Evolutionary Optimization of Constrained Problems*, pages 53–64. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-39883-7. doi: 10.1007/978-3-540-39883-7_3. URL https://doi.org/10.1007/978-3-540-39883-7_3.

[183] T. Weise. Global optimization algorithms-theory and application. *Self-published*, 2, 2009.

[184] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, pages 468–486, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg. ISBN 978-3-642-48782-8.

[185] K. M. Wiig. Chapter 1 - knowledge management: An emerging discipline rooted in a long history. In C. Despres and D. Chauvel, editors, *Knowledge Horizons*, pages 3 – 26. Butterworth-Heinemann, Boston, 2000. ISBN 978-0-7506-7247-4. doi: https://doi.org/10.1016/B978-0-7506-7247-4.50004-5. URL https://www.sciencedirect.com/science/article/pii/B9780750672474500045.

[186] S. J. Wilson. Data representation for time series data mining: time domain approaches. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(1), 2017.

[187] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[188] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 1033–1040, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143974. URL http://doi.acm.org/10.1145/1143844.1143974.

[189] Y. H. Yacov, S. L. Leon, and A. W. David. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):296–297, 1971.

[190] X.-S. Yang. Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, 29(2):175–184, Apr 2013. ISSN 1435-5663. doi: 10.1007/s00366-012-0254-1. URL https://doi.org/10.1007/s00366-012-0254-1.

[191] X.-S. Yang. Chapter 14 - multi-objective optimization. In X.-S. Yang, editor, *Nature-Inspired Optimization Algorithms*, pages 197 – 211. Elsevier, Oxford, 2014. ISBN 978-0-12-416743-8. doi: https://doi.org/10.1016/B978-0-12-416743-8.00014-2. URL http://www.sciencedirect.com/science/article/pii/B9780124167438000142.

[192] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, Dec 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2007.892759.

[193] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition, 2012. ISBN 1439830037, 9781439830031.

[194] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms — a comparative case study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature — PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings*, pages 292–301, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-49672-4. doi: 10.1007/BFb0056872. URL https://doi.org/10.1007/BFb0056872.

[195] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, Nov 1999. ISSN 1089-778X. doi: 10.1109/4235.797969.

[196] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

[197] W. Zucchini. An introduction to model selection. *Journal of Mathematical Psychology*, 44(1):41 – 61, 2000. ISSN 0022-2496. doi: https://doi.org/10.1006/jmps.1999.1276. URL http://www.sciencedirect.com/science/article/pii/S0022249699912762.