Corrección de Instancias Ruidosas mediante Gradientes en Datos Tabulares y Series Temporales

J.Javier Alonso-Ramos[1,2 [0000−0003−2197−2813]],
Ignacio Aguilera-Martos[1,2 [0000−0001−9620−5235]],
Francisco Herrera[1,2 [0000−0002−7283−312X]] y
Andrés Herrera-Poyatos[3 [0000−0001−9183−5262]]

[1] Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España
[2] DaSCI: Instituto Andaluz Interuniversitario en Ciencia de Datos e Inteligencia Computacional, Granada, España
[3] Departamento de Álgebra, Universidad de Granada, Granada, España
{jjalonso,nacheteam,andreshp}@ugr.es, herrera@decsai.ugr.es

Resumen En el paradigma de la Inteligencia Artificial (IA) Centrada en Datos, mejorar la calidad de los mismos es esencial para lograr robustez en algoritmos de aprendizaje automático. Sin embargo, muchos métodos de reducción de ruido dependen de suposiciones estadísticas rígidas o requieren datos de referencia limpios, lo que limita su aplicabilidad en escenarios del mundo real. En este trabajo proponemos CIRPEG, un marco de trabajo basado en gradientes para el refinamiento de datos que aprovecha una red neuronal preentrenada para corregir iterativamente las observaciones ruidosas, optimizando el espacio de entrada mientras mantiene el modelo fijo. CIRPEG es aplicable tanto a regresión con datos tabulares como al pronóstico de series temporales, e incorpora una estrategia basada en consenso para asegurar actualizaciones temporalmente coherentes en entornos secuenciales. Los experimentos realizados en diez conjuntos de datos del mundo real muestran que el enfoque propuesto produce mejoras consistentes en el rendimiento predictivo en tareas posteriores, a la vez que preserva la estructura estadística de los datos, tal y como lo indican las métricas distribucionales y basadas en correlación. Además, CIRPEG puede mejorar la generalización en conjuntos de datos nominalmente limpios, actuando como una forma de regularización a nivel de conjunto de datos. Código: `https://github.com/ari-dasci/S-DenoGrad`.

Palabras clave: IA Centrada en Datos · Ruido · Optimización Basada en Gradientes · Refinamiento de Datos · Aprendizaje Automático · Redes Neuronales Profundas

1. Introducción

En IA Centrada en Datos, la calidad del conjunto de entrenamiento condiciona directamente la generalización de los modelos [15]. En regresión y pronóstico,

la práctica habitual de reducción de ruido sigue dominada por enfoques de filtrado que detectan y eliminan instancias [14], lo que puede reducir la cobertura de la variedad de los datos y descartar muestras difíciles pero válidas [11]. Como alternativa, la corrección directa de observaciones ruidosas suele depender de hipótesis rígidas o de pares limpios-ruidosos, poco realistas en escenarios reales.

Este trabajo propone CIRPEG (Corrección de Instancias Ruidosas en Pronóstico y Estimación mediante Gradientes), un marco de trabajo de refinamiento en diferido para datos continuos. La idea central es congelar una red neuronal profunda entrenada y optimizar iterativamente entradas y objetivos en el espacio de datos, en sentido inverso a la optimización adversarial. Bajo el sesgo espectral de redes profundas, las componentes de baja frecuencia se ajustan antes que el ruido y convierten a los gradientes en direcciones de corrección útiles hacia regiones de mayor coherencia estructural del subespacio de los datos [1, 8, 12].

Validamos CIRPEG en diez conjuntos de datos reales, tanto tabulares como de series temporales, frente a siete métodos de referencia y múltiples modelos posteriores. Evaluamos simultáneamente mejora predictiva e integridad distribucional para verificar que las ganancias de precisión no se logran a costa de distorsionar la estructura de los datos.

Las principales contribuciones de este trabajo son: un marco metodológico independiente del modelo para corregir datos continuos optimizando entradas y objetivos sin referencias limpias, una extensión secuencial por consenso que acumula y promedia gradientes en ventanas solapadas para preservar la coherencia temporal, una validación empírica amplia que muestra mejoras consistentes en el rendimiento posterior y alta fidelidad distribucional frente a métodos de referencia estadísticos, espectrales y neuronales y la observación de un efecto de regularización de datos, con ganancias incluso en escenarios nominalmente limpios, compatibles con la mitigación de ruido latente.

El resto de este artículo se organiza de la siguiente manera. La Sección 2 resume el marco conceptual. La Sección 3 presenta CIRPEG, incluyendo la formulación matemática del proceso de optimización de entradas y la estrategia de consenso para datos secuenciales. La Sección 4 describe la metodología experimental y las líneas base. La Sección 5 analiza los resultados. Finalmente, la Sección 6 resume los hallazgos y el trabajo futuro.

2. Antecedentes y Marco Conceptual

En regresión, el ruido puede afectar tanto a los atributos como a la variable objetivo, alterando espacios continuos de datos. Aunque suele modelarse como $y_{obs} = f(x) + \epsilon$, con $\epsilon \sim N(0, \sigma^2)$, en escenarios reales pueden aparecer sesgos sistemáticos, heterocedasticidad dependiente de la entrada y valores atípicos. Este problema es especialmente relevante en series temporales, donde eliminar observaciones ruidosas puede romper la estructura secuencial, distorsionar dependencias retardadas y propagar errores en predicciones autorregresivas a largo plazo. Por ello, en estos contextos resulta preferible corregir las observaciones corruptas en lugar de descartarlas.

La eliminación de ruido busca recuperar una señal latente limpia a partir de una observación corrupta, y constituye un componente central en problemas inversos, imagen y aprendizaje automático [10]. Las técnicas clásicas incluyen filtros temporales y estadísticos, métodos espectrales y modelos basados en aprendizaje. Los primeros suavizan o proyectan los datos bajo supuestos previos sobre la señal y el ruido, mientras que los enfoques espectrales permiten tratar señales no estacionarias, aunque suelen depender de elecciones manuales o pueden introducir artefactos. Por su parte, los modelos profundos han alcanzado resultados competitivos al aprender reconstrucciones no lineales, como en marcos variacionales guiados por modelos físicos de imagen [3]. Sin embargo, muchos de estos métodos requieren pares de datos limpios y ruidosos, que rara vez están disponibles en aplicaciones reales.

CIRPEG aborda esta limitación desde un enfoque auto-supervisado. En lugar de asumir una forma fija del ruido o depender de etiquetas limpias, utiliza el propio modelo predictivo para guiar la corrección de los datos. Esta idea se relaciona con trabajos que explotan gradientes respecto a la entrada, tanto para modelar distribuciones de datos [13] como para invertir representaciones profundas [9]. No obstante, a diferencia de los métodos adversarios, donde los gradientes se emplean para inducir perturbaciones que degradan la predicción [6], CIRPEG los utiliza con el objetivo opuesto: desplazar las entradas hacia regiones más coherentes con la estructura aprendida por el modelo.

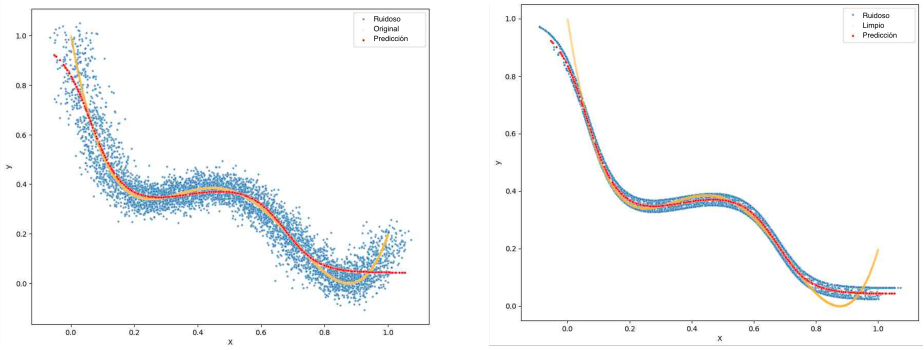
La justificación de este proceso se apoya en la dinámica de entrenamiento de las redes neuronales profundas. Estas tienden a aprender primero patrones simples, de baja frecuencia y generalizables, antes de memorizar variaciones ruidosas o ejemplos complejos [1, 12]. Esta propiedad, conocida como sesgo espectral, también se ha observado en problemas con ruido en las etiquetas, donde la regularización del aprendizaje temprano ayuda a evitar la memorización del ruido [8]. Así, incluso entrenado con datos ruidosos, el modelo puede capturar la estructura dominante de la señal y servir como guía para refinar las entradas.

3. CIRPEG: Un *Marco de Trabajo* de Reducción de Ruido Basado en Gradientes

CIRPEG es una metodología independiente del modelo, operando con cualquier predictor diferenciable, aunque su eficacia depende de la calidad representacional del modelo neuronal base. Con la red θ congelada, CIRPEG actualiza iterativamente instancias en el espacio de datos para alinearlas con el subespacio aprendido, usando la señal de gradiente como guía de corrección.

Esta alineación se ilustra en la Figura 1. Un modelo neuronal de regresión entrenado con datos ruidosos captura la tendencia dominante, y la función aprendida actúa como un atractor: después de CIRPEG, las muestras se mueven hacia el subespacio aprendido por la red neuronal, reduciendo la varianza mientras se preserva la estructura global.

Una ventaja práctica clave es la eficiencia en producción. A diferencia de los reductores de ruido de aprendizaje profundo que entrenan un modelo de limpieza



(a) Red neuronal profunda (rojo) ajustada a datos ruidosos (azul) junto a la verdad base (naranja).

(b) Después de CIRPEG: las muestras se desplazan hacia el subespacio aprendido.

Figura 1: Datos ruidosos en 2D antes y después de aplicar CIRPEG.

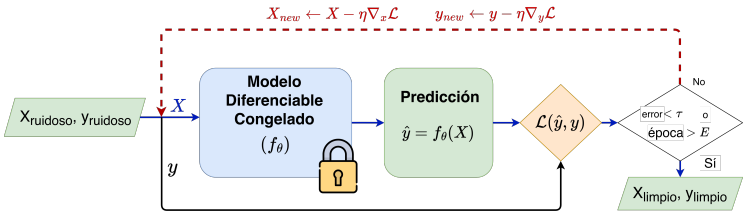


Figura 2: Flujo general de CIRPEG: entrenamiento del modelo neuronal base y refinamiento iterativo del conjunto de datos.

separado, CIRPEG reutiliza la misma red neuronal predictiva usada en tareas posteriores.

CIRPEG se distribuye como una biblioteca de Python de código abierto, disponible para su instalación a través del Python Package Index (PyPI)[4]. El código fuente completo, incluyendo documentación y ejemplos de uso, está alojado en el repositorio oficial de GitHub.

3.1. Formulación Matemática

A continuación, proporcionamos una descripción formal de CIRPEG en tres partes: definición del problema, objetivo de optimización con hiperparámetros y refinamiento guiado por gradiente.

Formulación del Problema. Sea $\mathcal{D}_{ruidoso} = \{(x_i, y_i)\}_{i=1}^{N}$ un conjunto de datos observado, donde cada instancia es una realización ruidosa de una muestra

[4] Comando de instalación: `pip install denograd`

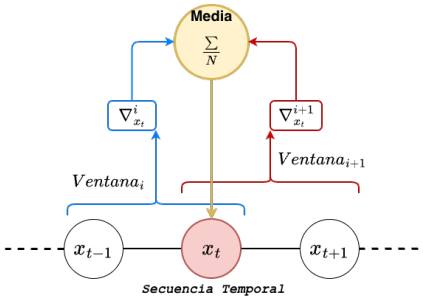


Figura 3: Estrategia de consenso para series temporales mediante acumulación y promedio de gradientes en ventanas solapadas.

limpia subyacente (x_i^*, y_i^*), es decir, $x_i = x_i^* + \epsilon_{x,i}$ y $y_i = y_i^* + \epsilon_{y,i}$, con términos de ruido estocástico $\epsilon_{x,i}$ y $\epsilon_{y,i}$. CIRPEG refina (x_i, y_i) hacia una aproximación coherente del subespacio aprendido por una red neuronal profunda f_θ entrenada en $\mathcal{D}_{ruidoso}$. En este marco, y no se interpreta como una etiqueta discreta externa, sino como otra magnitud continua de la observación; por ello se refina junto con X dentro de la misma dinámica de optimización.

Objetivo e Hiperparámetros. El refinamiento se formula como una minimización de la pérdida con respecto a las instancias de datos mientras se mantienen fijos los parámetros del modelo θ. En regresión, utilizamos el error cuadrático medio (ECM): $\mathcal{L}(f_\theta(x), y) = \|f_\theta(x) - y\|_2^2$

La optimización está regida por dos hiperparámetros clave:

- Tasa de Reducción de Ruido (TRR-η): controla la magnitud de cada paso de actualización en el espacio de características.
- Umbral de Tolerancia (τ): define una condición de compuerta que evita las actualizaciones cuando el error de predicción satisface $|f_\theta(x) - y| \leq \tau$, preservando las muestras de alta confianza y evitando un suavizado excesivo.

El mecanismo de selección está codificado por una máscara binaria $\mathbb{I}_{ruidoso}$ que activa las actualizaciones solo para las instancias consideradas ruidosas. Al preservar una estocasticidad controlada en muestras de bajo error, actúa como una regularización implícita, evita el colapso hacia un hiperplano artificialmente liso y mejora la generalización en tareas posteriores [2].

Refinamiento Basado en Gradientes. CIRPEG calcula la sensibilidad de la pérdida con respecto tanto a las entradas como a los objetivos como

$$g_x = \nabla_x \mathcal{L}(f_\theta(x), y), \quad g_y = \nabla_y \mathcal{L}(f_\theta(x), y) \tag{1}$$

Para equilibrar las correcciones entre variables heterogéneas, los gradientes se normalizan conjuntamente concatenándolos y dividiéndolos por su norma L_2:

$$\hat{g}_x = \frac{g_x}{\|[g_x, g_y]\|_2 + \epsilon}, \quad \hat{g}_y = \frac{g_y}{\|[g_x, g_y]\|_2 + \epsilon} \tag{2}$$

Esta *Normalización Conjunta* escala las actualizaciones de acuerdo a la pendiente local del paisaje de pérdida y evita que una sola variable domine la corrección. El paso de refinamiento iterativo es

$$x \leftarrow x - \eta \cdot \hat{g}_x \cdot \mathbb{I}_{ruidoso}, \quad y \leftarrow y - \eta \cdot \hat{g}_y \cdot \mathbb{I}_{ruidoso} \tag{3}$$

En la práctica, esto se implementa iterativamente sobre el conjunto de datos: los gradientes se calculan mediante retropropagación y se aplican directamente en el espacio de entrada mientras los parámetros del modelo permanecen congelados. Los detalles completos de implementación del procedimiento tabular están disponibles en el repositorio oficial de CIRPEG en GitHub.

3.2. Extensión a Series Temporales: Estrategia de Consenso

Aplicar CIRPEG a series temporales introduce un desafío clave: la dependencia temporal. A diferencia de las instancias tabulares independientes, un punto en el tiempo t, denotado x_t, aparece en múltiples ventanas deslizantes superpuestas. Para una ventana de tamaño W, x_t puede ser el último elemento de la ventana que comienza en $t - W + 1$, el primer elemento de la ventana que comienza en t, y cualquier posición intermedia. Una actualización ingenua estilo tabular crearía por lo tanto conflictos, ya que los gradientes de diferentes ventanas podrían sugerir correcciones distintas para el mismo x_t.

Para resolver esto, CIRPEG utiliza una *Estrategia de Consenso mediante Acumulación de Gradientes*. En lugar de actualizar después de cada lote, acumula las correcciones propuestas para cada paso de tiempo a través de todas las ventanas que lo cubren. Sea G_t el gradiente acumulado para el paso de tiempo t, y C_t el número de ventanas contribuyentes.

En cada época, el marco de trabajo recorre todas las ventanas deslizantes, calcula los gradientes y los acumula en registros globales en los índices de tiempo correspondientes. Después del pase completo, la actualización para el paso de tiempo t se obtiene promediando los gradientes acumulados como $x_t^{nuevo} = x_t^{antiguo} - \eta \cdot \frac{G_t}{C_t}$.

Este mecanismo asegura la coherencia temporal: cada actualización refleja una dirección de consenso a través de todos los contextos en los que aparece la observación. La misma lógica se aplica al objetivo y, permitiendo que CIRPEG reduzca el ruido tanto en las características como en los objetivos en entornos secuenciales. Al utilizar modelos recurrentes como LSTMs o GRUs, los grafos computacionales de estados ocultos deben retenerse para preservar el flujo de la señal temporal durante la reducción de ruido.

4. Configuración Experimental

Evaluamos CIRPEG en diez conjuntos de datos reales (cinco tabulares y cinco temporales) y lo comparamos con siete métodos de referencia de las familias estadística, espectral y de aprendizaje profundo. Para evitar sesgo por arquitectura, medimos el efecto del refinamiento sobre un conjunto diverso de regresores

posteriores. Los experimentos se ejecutaron con aceleración GPU en un entorno estándar.

4.1. Estrategia de Evaluación

Para evaluar CIRPEG de forma rigurosa, utilizamos un protocolo de curación en diferido que mide tanto la ganancia predictiva posterior como la preservación de la distribución y características de los datos.

El objetivo del experimento no es la inferencia en línea sobre objetivos desconocidos, sino la utilidad de refinar particiones continuas disponibles de forma completa en problemas de regresión y pronóstico. Por ello, cada partición se procesa de manera independiente y sin mezclar información entre entrenamiento, validación y prueba.

Sean $\mathcal{D}_{train}$ y $\mathcal{D}_{test}$ las particiones originales ruidosas de entrenamiento y prueba. Entrenar y evaluar un regresor posterior en estas particiones en bruto produce el error base ECM_{base}. Después refinamos por separado cada partición, obteniendo $\hat{\mathcal{D}}_{train}$ y $\hat{\mathcal{D}}_{test}$ con CIRPEG o con los métodos competidores. Si el modelo entrenado con datos refinados logra un error menor que el base, la reducción de ruido ha tenido éxito. La integridad estructural del problema representado por los datos se evalúa por separado mediante divergencias estadísticas.

Todos los conjuntos se dividieron en entrenamiento, validación y prueba (70 %/10 %/20 %). En series temporales, la partición fue estrictamente secuencial; en datos tabulares, se barajaron las muestras antes de dividir. Estandarizamos características y objetivo ($\mu = 0$, $\sigma = 1$) e inyectamos ruido gaussiano con $\sigma = 0{,}1$ en la batería principal.

4.2. Métricas

Utilizamos una estrategia de evaluación dual para capturar tanto la *ganancia predictiva* como la *integridad distribucional* después de la reducción de ruido.

Para cuantificar la mejora del rendimiento, utilizamos el Error Cuadrático Medio (ECM) y el Porcentaje de Mejora (PM) del error con el ruido reducido en relación con la línea base. Un PM positivo indica una mitigación exitosa del ruido:

$$\mathrm{ECM} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

$$\mathrm{PM} = \left(\frac{\mathrm{ECM}_{base} - \mathrm{ECM}_{deno}}{\mathrm{ECM}_{base}}\right) \times 100 \tag{5}$$

Para la integridad de los datos, medimos el cambio distribucional entre la distribución de entrada ruidosa P y la distribución de salida reducida en ruido Q con la distancia de Wasserstein por proyecciones (DWP), una aproximación eficiente en alta dimensión basada en proyecciones aleatorias 1D [7]. En la práctica, aproximamos la integral con 100 proyecciones aleatorias mediante la biblioteca POT [5]. Formalmente, $\mathrm{DWP}(P, Q) = \int_{\mathbb{S}^{d-1}} W_2(\theta_{\#}P, \theta_{\#}Q)\, d\theta$

Además, medimos $\bar{\rho} = \frac{1}{M}\sum_{j=1}^{M} \mathrm{corr}(X_{*,j}, \hat{X}_{*,j})$, la correlación de Pearson promedio entre cada columna de características en los conjuntos de datos ruidosos y aquellos con el ruido reducido. Valores cercanos a 1 indican que la forma y las tendencias de la señal se han preservado.

4.3. Conjuntos de Datos

El estudio utiliza diez conjuntos de datos del mundo real que abarcan finanzas, física, IoT y salud, cinco tabulares y cinco de series temporales. Por restricciones de espacio, la descripción detallada de cada conjunto de datos, incluida su fuente, dimensionalidad, tamaño y frecuencia, se proporciona en el repositorio oficial del proyecto: `https://github.com/ari-dasci/S-DenoGrad`.

4.4. Líneas Base de Comparación y Modelos Posteriores

Para validar CIRPEG, lo evaluamos frente a reductores de ruido establecidos de las tres familias introducidas anteriormente. Para los métodos estadísticos y del dominio temporal, incluimos Kalman Filter (KF), Moving Average (MA) y Principal Component Analysis (PCA). Para los métodos espectrales, incluimos Empirical Mode Decomposition (EMD) y Wavelet Thresholding Denoising (WTD). Para la reducción de ruido basada en aprendizaje profundo, comparamos contra Denoising Autoencoder (DAE) y DN-ResNet.

Para verificar que la eficacia de la reducción de ruido se generaliza a través de los sesgos inductivos, utilizamos un conjunto diverso de regresores posteriores (lineales, basados en vecinos, árboles y redes neuronales).

Para los datos de series temporales, evaluamos modelos que capturan las dependencias temporales a través de diferentes mecanismos: XGBoost, LSTM, xLSTM, CNN-LSTM y DLinear. Para mantener el manuscrito compacto, la configuración detallada de hiperparámetros se proporciona en el repositorio oficial de código.

5. Resultados y Análisis

Esta sección presenta una evaluación exhaustiva de CIRPEG. Primero comparamos frente a reductores de ruido de referencia para establecer la competitividad empírica. Luego, validamos la optimización conjunta de características y objetivos mediante una ablación, y finalmente analizamos el comportamiento a través de regresores posteriores, arquitecturas base, ajustes de hiperparámetros, niveles de ruido y compromisos computacionales.

5.1. Comparación con Reductores de Ruido de Vanguardia

El objetivo principal de un marco de trabajo de reducción de ruido es mejorar la precisión posterior sin dañar el problema subyacente asociado al conjunto de datos. La Tabla 1 muestra las magnitudes alcanzadas por CIRPEG y los demás

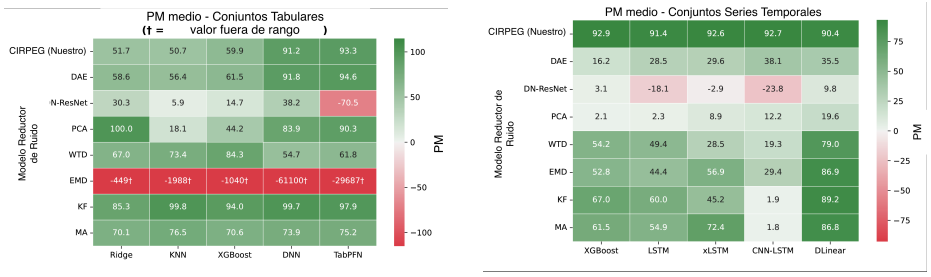
métodos de reducción de ruido en los diez conjuntos. En CIRPEG, el patrón más fuerte aparece en series temporales, con mejoras superiores al 86 % en todos los casos, mientras que en el dominio tabular mantiene fidelidad estructural casi perfecta. La ganancia predictiva bruta de CIRPEG compite siempre por el

Cuadro 1: Evaluación exhaustiva de los métodos de reducción de ruido en todos los conjuntos de datos. Para cada conjunto de datos, reportamos PM ↑, DWP ↓, y $\bar{\rho}$ ↑. Los mejores resultados por métrica están resaltados en **negrita**.

Conjunto de Datos	Métrica	CIRPEG	DAE	DN-ResNet	PCA	WTD	EMD	KF	MA
Conjuntos de Datos Tabulares Estáticos									
House Prices	PM ↑	77.1 %	76.2 %	17.6 %	89.9 %	64.0 %	-83.2 %	**98.5 %**	64.1 %
	DWP ↓	**0.0068**	0.0746	0.0137	0.0486	0.6630	0.3252	0.7967	0.5432
	$\bar{\rho}$ ↑	**0.9994**	0.9845	0.9990	0.9775	0.5324	0.5510	0.2677	0.4715
Lattice Physics	PM ↑	48.5 %	83.8 %	36.0 %	62.2 %	**99.6 %**	-552.5 %	99.1 %	76.8 %
	DWP ↓	**0.0019**	0.3321	0.0166	0.0344	0.8800	0.4570	0.8424	0.5563
	$\bar{\rho}$ ↑	**1.0000**	0.8126	0.9997	0.9757	0.1247	0.5311	0.2184	0.4467
Parkinsons	PM ↑	72.2 %	53.8 %	21.6 %	61.2 %	70.9 %	49.6 %	**82.8 %**	63.2 %
	DWP ↓	**0.0060**	0.0907	0.0270	0.0674	0.2179	0.1411	0.3113	0.2076
	$\bar{\rho}$ ↑	0.9981	0.9828	**0.9985**	0.9784	0.8676	0.8426	0.7456	0.8284
RT IOT 2022	PM ↑	82.0 %	76.3 %	52.1 %	57.1 %	24.2 %	-93414.3 %	**98.0 %**	84.3 %
	DWP ↓	**0.0015**	0.6481	0.1494	0.0818	0.1065	0.9651	0.6021	0.4369
	$\bar{\rho}$ ↑	**1.0000**	0.9598	0.9795	0.9765	0.9822	0.5576	0.6116	0.7161
Support2	PM ↑	67.1 %	78.0 %	16.6 %	66.4 %	98.7 %	35.8 %	**99.1 %**	79.1 %
	DWP ↓	**0.0134**	0.1205	0.0175	0.0357	0.5662	0.2930	0.7766	0.5300
	$\bar{\rho}$ ↑	**0.9993**	0.9558	0.9985	0.9803	0.6236	0.5453	0.2645	0.4713
Conjuntos de Datos de Series Temporales									
Daily Climate	PM ↑	87.2 %	21.3 %	-15.3 %	1.4 %	91.4 %	70.1 %	**95.9 %**	81.0 %
	DWP ↓	0.0330	0.0704	0.3547	$\mathbf{3{,}1 \times 10^{-16}}$	0.2970	0.4255	0.5477	0.4335
	$\bar{\rho}$ ↑	0.9898	0.9958	0.9528	**1.0000**	0.8688	0.7464	0.6907	0.8020
ECL	PM ↑	**98.4 %**	65.1 %	-5.1 %	23.0 %	23.5 %	34.0 %	43.9 %	26.1 %
	DWP ↓	**0.0120**	0.1160	0.0437	0.0415	0.2910	0.2529	0.5820	0.1237
	$\bar{\rho}$ ↑	**0.9995**	0.9506	0.9904	0.9746	0.9197	0.9761	0.4855	0.9619
ETT	PM ↑	90.9 %	15.2 %	0.8 %	8.2 %	89.9 %	76.3 %	**98.9 %**	91.6 %
	DWP ↓	**0.0073**	0.0619	0.0325	0.0183	0.1634	0.1061	0.2947	0.0708
	$\bar{\rho}$ ↑	0.9983	0.9946	**0.9989**	0.9970	0.9271	0.9562	0.7872	0.9689
Microsoft Stock	PM ↑	**97.6 %**	47.7 %	16.8 %	21.4 %	25.0 %	28.4 %	38.4 %	30.4 %
	DWP ↓	0.1288	0.0949	0.0472	**0.0169**	0.1040	0.6485	0.1733	0.0741
	$\bar{\rho}$ ↑	0.9583	0.9954	**0.9980**	0.9968	0.9739	0.7915	0.9398	0.9791
WTH	PM ↑	86.9 %	16.1 %	3.7 %	1.1 %	49.1 %	97.2 %	**100.0 %**	94.7 %
	DWP ↓	0.0375	0.0633	**0.0146**	0.0550	0.1807	0.0646	0.2202	0.0941
	$\bar{\rho}$ ↑	0.9820	0.9946	**0.9989**	0.9796	0.9142	0.9521	0.8574	0.9480

primer puesto, con mejoras notables como 98,4 % en ECL y 97,6 % en Microsoft Stock, pero su principal ventaja aparece en la preservación de los datos. Para validar esta comparación en todos los conjuntos y métodos sin suposiciones de normalidad, aplicamos el test no paramétrico de Friedman seguida del test post-hoc de Nemenyi con $\alpha = 0{,}05$. Siguiendo a Demšar [4], este protocolo es el estándar para evaluaciones comparativas de múltiples algoritmos y conjuntos de datos porque compara la consistencia de los *rankings* en lugar de las medias sensibles a los valores atípicos.

La prueba de Friedman seguida de Nemenyi ($\alpha = 0{,}05$) confirma la ventaja global de CIRPEG en el compromiso rendimiento-fidelidad. En términos de rangos medios, el método lidera en la distancia de Wasserstein por proyecciones y mantiene posicionamiento competitivo en mejora predictiva, lo que respalda un comportamiento estable entre dominios y arquitecturas posteriores.



(a) Dominio tabular. (b) Series temporales.

Figura 4: Mapas de calor del porcentaje de mejora (PM) de los reductores de ruido a través de distintos modelos posteriores.

5.2. Estudio de Ablación: Optimización Conjunta de Características y Objetivo

Analizamos una decisión central del método: actualizar conjuntamente características X y objetivo y. En regresión y pronóstico, mantener y congelada en estado ruidoso sesga las correcciones de X; al actualizar ambos, cada par (x, y) se desplaza de forma coherente hacia regiones de menor discrepancia inducida por el modelo.

La ablación confirma este efecto: la mejora media pasa de 14,41 % (solo X) a 49,17 % (optimización conjunta $X{+}y$), con mediana de 6,30 % frente a 45,08 %. En paralelo, la variante conjunta mantiene alta fidelidad distribucional y correlacional. El contraste de Wilcoxon sobre PM pareado reporta $W = 4{,}0$ y $p = 0{,}0068$, con significación para $\alpha = 0{,}05$ y $\alpha = 0{,}01$.

En resumen, la ablación respalda fuertemente la optimización conjunta. Refinar simultáneamente las características y los objetivos elimina un sesgo de gradiente sistemático que penaliza gravemente las tareas de series temporales, produciendo un incremento de 3,4× estadísticamente significativo en la mejora predictiva. Esta ganancia se logra sin sacrificar la estructura: la variante conjunta mejora la fidelidad distribucional y preserva correlaciones casi perfectas.

Análisis complementario. CIRPEG mantiene buen comportamiento al cambiar de regresor posterior, lo que sugiere que la mejora proviene del refinamiento del dato y no de una arquitectura concreta. La Figura 4 resume este patrón en dominios tabulares y temporales. La red neuronal base influye en la calidad de los gradientes, pero el patrón general se mantiene: tasas moderadas-altas ($\eta \in [0{,}01, 0{,}1]$), umbral $\tau \approx 0{,}1$ y alrededor de 200 iteraciones ofrecen un compromiso robusto entre mejora predictiva y preservación estructural. En conjunto, el coste adicional frente a filtros clásicos sigue siendo razonable para una etapa en diferido de curación de datos.

5.3. Discusión e Implicaciones más Amplias

El rendimiento de CIRPEG depende de que el modelo neuronal base capture una aproximación razonable de la estructura latente. Si el modelo está subajustado o existe cambio de distribución severo, las correcciones pueden degradarse; además, el marco está especialmente orientado a ruido estocástico cercano a media cero. Como guía práctica, recomendamos su uso como etapa en diferido con parámetros moderados ($\eta \in [0{,}01, 0{,}1]$, $\tau \approx 0{,}1$, 100–200 iteraciones). En este punto, CIRPEG funciona como alternativa intermedia frente a filtros clásicos y reductores de ruido de alto coste: reutiliza un predictor existente y aplica optimización de entradas para corregir estructura sin entrenamiento adicional [3, 6, 9, 10, 13].

6. Conclusiones y Trabajo Futuro

En este trabajo se presenta CIRPEG, un marco independiente del modelo que reformula la eliminación de ruido como un problema de optimización de las entradas. Para ello, se congela una red base previamente entrenada y se retropropagan correcciones directamente sobre el espacio de características y objetivos, permitiendo refinar observaciones corruptas tanto en entrenamiento como en datos no vistos, sin necesidad de disponer de referencias limpias. Este diseño reutiliza el conocimiento representacional aprendido por el modelo para guiar las muestras ruidosas hacia la variedad de datos aprendida.

La evaluación muestra mejoras consistentes en el rendimiento posterior de los modelos, manteniendo una elevada fidelidad distribucional y topológica frente a métodos estadísticos, espectrales y neuronales. Los estudios de ablación confirman que refinar conjuntamente características y objetivos supera al refinamiento únicamente de las características, respaldando la decisión metodológica central. Asimismo, CIRPEG actúa como un regularizador intrínseco de los datos, reduciendo ruido latente e inconsistencias entre variables, incluso en conjuntos nominalmente no corruptos.

No obstante, sus beneficios dependen de la calidad de la red base y de la naturaleza del ruido. El método ha sido diseñado y validado principalmente para perturbaciones estocásticas próximas a ruido gaussiano de media cero, por lo que ante sesgos sistemáticos intensos o desplazamientos distribucionales significativos los gradientes podrían seguir distorsiones aprendidas por el modelo en lugar de recuperar la señal real. Como trabajo futuro, se plantea explorar estrategias adaptativas, integrar redes fundacionales de mayor escala para mejorar la robustez frente al cambio de distribución, y extender la corrección basada en gradientes a tareas de clasificación y procesos de curación de datos a gran escala.

Agradecimientos. Esta investigación es parte del Proyecto TSI-100927-2023-1, financiado por Plan de Recuperación, Transformación y Resiliencia Europeo, de la Unión Europea-Next Generation EU a través del Ministerio para la Transformación Digital y de la Función Pública y está parcialmente financiada por el Ministerio de Ciencia y Tecnología de España en el marco del proyecto PID2023-150070NB-I00, financiado por MICIU/AEI/10.13039/501100011033 y por FEDER, UE.

Declaración de intereses. Los autores declaran que no tienen conflicto de intereses con respecto a la publicación de este artículo.

Referencias

1. Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y.: A closer look at memorization in deep networks. In: International Conference on Machine Learning (ICML). pp. 233–242. PMLR (2017)
2. Bishop, C.M.: Training with noise is equivalent to tikhonov regularization. Neural computation **7**(1), 108–116 (1995)
3. Cui, W., Pan, Z., Li, X., Tang, Y., Sun, S.: Physical imaging model-guided deep variational despeckling framework for ultrasound images. Knowledge-Based Systems **329** (2025)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006)
5. Flamary, R., Courty, N., Gramfort, A., Alaya, M.Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., et al.: Pot: Python optimal transport. Journal of Machine Learning Research **22**(78), 1–8 (2021)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
7. Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., Rohde, G.: Generalized sliced wasserstein distances. Advances in neural information processing systems **32** (2019)
8. Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 20331–20342 (2020)
9. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5188–5196 (2015)
10. Milanfar, P., Delbracio, M.: Denoising: a powerful building block for imaging, inverse problems and machine learning. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences **383** (2024)
11. Northcutt, C.G., Athalye, A., Mueller, J.: Pervasive label errors in test sets destabilize machine learning benchmarks. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021)
12. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning (ICML). pp. 5301–5310. PMLR (2019)
13. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
14. Szeghalmy, S., Fazekas, A.: A comparative study on noise filtering of imbalanced data sets. Knowledge-Based Systems **301** (2024)
15. Zha, D., Bhat, Z.P., Lai, K.H., Yang, F., Jiang, Z., Zhong, S., Hu, X.: Data-centric artificial intelligence: A survey. ACM Computing Surveys **57**(5), 1–42 (2025)