GREMLINS: etiquetado simbólico automático de series temporales mediante inferencia gramatical y motifs de longitud variable con límite de error

Christian Luna*[0009−0005−7502−0254], José María Luna[0000−0003−3537−2931], and Sebastián Ventura*[0000−0003−4216−6378]

Departamento de Ciencia de la Computación e Inteligencia Artificial,
Instituto Andaluz Interuniversitario en Ciencia de Datos e Inteligencia Computacional (DaSCI),
Universidad de Córdoba, 14071 Córdoba, España
{c.luna,jmluna,sventura}@uco.es

Resumen El etiquetado de datos constituye una pieza clave en aprendizaje automático, ya que permite dotar a los datos brutos de una estructura semántica que facilite su análisis e interpretación. En el caso de las series temporales, automatizar este proceso a nivel de punto sigue siendo una tarea especialmente exigente, debido a la necesidad de convertir señales continuas, complejas y a menudo ruidosas en representaciones discretas que sigan siendo comprensibles. Muchos de los métodos existentes, apoyados en técnicas de discretización y en el descubrimiento de *motifs*, dependen de elecciones de diseño delicadas (como la longitud de ventana, el tamaño del alfabeto o la exigencia de coincidencias exactas), lo que puede dificultar su generalización en escenarios heterogéneos. En este trabajo presentamos GREMLINS (*GRamática con Error acotado para descubrimiento de Motivos y etiquetado INterpretable en Series temporales*), un marco en dos etapas orientado al etiquetado automático de series temporales. En la primera etapa se lleva a cabo una discretización adaptativa guiada por un presupuesto de error relativo ϵ, definido por el usuario, con el objetivo de obtener una secuencia simbólica compacta sin perder la variabilidad local más relevante. A continuación, en la segunda etapa, se emplea un procedimiento de inferencia gramatical robusto al ruido para detectar patrones recurrentes de longitud variable y agrupar *motifs* con estructura similar mediante una tolerancia τ, calculada de forma determinista a partir de ϵ.
La evaluación experimental sobre 50 conjuntos de datos muestra que GREMLINS logra un equilibrio competitivo entre fidelidad y compresión frente a métodos de referencia en discretización e inferencia gramatical. Además, el método genera representaciones simbólicas organizadas e interpretables, lo que refuerza su interés como base para el etiquetado automático de series temporales.

Keywords: Etiquetado de Series Temporales, Representación Simbólica, *Motifs* Acotados por Error, *Motifs* de Longitud Variable, Inferencia Gramatical

* *Autores de correspondencias.*

1. Introducción

El etiquetado de datos consiste en asignar anotaciones semánticas a los datos brutos con el fin de aportarles contexto, estructura y significado [27]. Se trata de una etapa fundamental en aprendizaje automático, ya que la calidad de las etiquetas condiciona de forma directa el rendimiento de los modelos posteriores. Sin embargo, el etiquetado manual presenta limitaciones bien conocidas: requiere una inversión elevada de tiempo, esfuerzo y coste económico, y con frecuencia depende de expertos del dominio para garantizar anotaciones fiables [10]. Además, este proceso resulta difícil de escalar y está expuesto a errores humanos derivados de la fatiga o de la propia naturaleza repetitiva de la tarea.

Para mitigar estas limitaciones, se han explorado dos estrategias principales: la supervisión débil y el etiquetado débil. La supervisión débil [25] permite entrenar modelos con etiquetas ruidosas, incompletas o imprecisas, reduciendo así el esfuerzo manual a costa de una posible pérdida de precisión. Por su parte, el etiquetado débil [10] persigue una automatización más completa, generando anotaciones sobre grandes volúmenes de datos no etiquetados a partir de heurísticas o conocimiento externo. No obstante, cuando estas estrategias se aplican a series temporales, es habitual que asignen una única etiqueta a intervalos amplios, o incluso a la secuencia completa, en lugar de producir anotaciones finas a nivel de punto. En muchos escenarios reales, esta resolución resulta insuficiente, ya que es necesario describir simbólicamente la evolución de la señal a lo largo del tiempo para capturar eventos transitorios y cambios estructurales sutiles [13].

El etiquetado automático a nivel de punto en series temporales es especialmente complejo, porque exige transformar señales continuas y potencialmente ruidosas en representaciones discretas e interpretables [11]. En este contexto, la discretización desempeña un papel central. Los enfoques clásicos de binning, como equal-width (EWB)[7] o equal-frequency (EFB) [5], dependen fuertemente de la distribución de los datos y de la presencia de valores atípicos. Otros métodos ampliamente utilizados, como PAA [15], SAX [19] o PLA [17], resumen la serie mediante segmentaciones de tamaño fijo. Aunque estas estrategias han demostrado utilidad en numerosos problemas, siguen presentando limitaciones importantes cuando los patrones de interés tienen longitud variable o aparecen de forma inconsistente a lo largo de la señal.

En particular, continúan abiertos varios retos relevantes. En primer lugar, muchos métodos de segmentación dependen de conocimiento previo del dominio para identificar regiones de interés, lo que dificulta su generalización [26]. En segundo lugar, los enfoques basados en ventanas fijas no se adaptan bien a motivos de longitud variable [2]. En tercer lugar, buena parte de los métodos existentes siguen apoyándose en coincidencias exactas entre patrones, lo que reduce su robustez frente a pequeñas variaciones estructurales o ruido, muy frecuentes en señales reales [12,13]. Como consecuencia, patrones esencialmente similares pueden terminar tratándose como diferentes, dificultando tanto el descubrimiento de regularidades como la obtención de etiquetados estables e interpretables.

En este trabajo se propone GREMLINS (*GRamática con Error acotado para descubrimiento de Motivos y etiquetado INterpretable en Series temporales*), un

método ligero en parámetros para el etiquetado automático de series temporales. La propuesta se apoya en un único parámetro principal, un presupuesto de error relativo ϵ, a partir del cual se deriva de forma determinista una tolerancia τ. A diferencia de otros enfoques que dependen de segmentaciones rígidas, tamaños de ventana prefijados o definiciones exactas de motivo, GREMLINS busca obtener representaciones simbólicas más realistas y robustas de la señal.

El método se articula en dos fases. En la primera, la serie temporal se segmenta de forma adaptativa mediante un criterio acotado por error, generando una secuencia simbólica compacta y fiel a la variabilidad local. En la segunda, se aplica inferencia gramatical tolerante al ruido para identificar patrones recurrentes y agrupar motivos estructuralmente similares bajo una representación simbólica común. De este modo, GREMLINS permite generar anotaciones interpretables a nivel de segmento sin recurrir a reglas *ad hoc* específicas de cada dominio.

Para validar la propuesta, se realiza una evaluación experimental amplia sobre 50 conjuntos de datos de distintos dominios, comparando GREMLINS con métodos representativos de discretización e inferencia gramatical. De forma resumida, las principales contribuciones de este trabajo son las siguientes:

- Un procedimiento de segmentación adaptativa acotada por error que evita fijar manualmente tamaños de ventana o granularidades simbólicas;
- Un esquema de inferencia gramatical tolerante al ruido que favorece el descubrimiento y agrupamiento de motivos de longitud variable;
- Una validación empírica en múltiples dominios que muestra la competitividad del método en términos de robustez, interpretabilidad y escalabilidad.

El resto del artículo se organiza como sigue. La Sección 2 presenta los conceptos de base y el planteamiento del problema. La Sección 3 describe la metodología propuesta. La Sección 4 resume el diseño experimental y los resultados. Por último, la Sección 5 recoge las principales conclusiones y futuras líneas de trabajo.

2. Preliminares

Una serie temporal T de longitud n es una secuencia ordenada de observaciones reales $T = \{t_1, t_2, \ldots, t_n\}$, donde cada $t_i \in \mathbb{R}$ representa el valor registrado en el instante i. Este tipo de datos aparece en numerosos dominios y suele presentar tendencias, cambios abruptos y patrones recurrentes de interés para tareas como clasificación [1], predicción [14] o detección de anomalías [4].

En este trabajo se considera el problema del etiquetado simbólico de series temporales a nivel de punto o de segmento. Dada una serie temporal $T = (t_1, \ldots, t_n)$, el etiquetado a nivel de punto consiste en definir una función $\lambda : \{1, \ldots, n\} \to \Sigma$, donde $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ es un alfabeto finito, de modo que se obtiene una secuencia de etiquetas $L = (\ell_1, \ell_2, \ldots, \ell_n)$ con $\ell_i = \lambda(i)$. De forma análoga, el etiquetado a nivel de segmento asigna un símbolo a cada intervalo

de una partición temporal, induciendo posteriormente una secuencia de etiquetas a nivel de punto por expansión. En ambos casos, el objetivo es transformar la señal continua original en una representación simbólica más compacta e interpretable, capaz de reflejar estados, motivos o cambios relevantes a lo largo del tiempo. Dado que la propuesta presentada en este artículo combina discretización e inferencia gramatical, a continuación se resumen brevemente ambos componentes.

2.1. Discretización de series temporales

La discretización transforma una señal continua en una representación simbólica o segmentada de menor complejidad, facilitando el análisis estructural posterior. En series temporales, este paso resulta especialmente relevante porque permite reducir dimensionalidad, mejorar la interpretabilidad y simplificar la búsqueda de patrones recurrentes.

Entre los enfoques más sencillos se encuentran los métodos basados en intervalos. Equal-Width Binning (EWB) [7] divide el rango de valores en intervalos de igual amplitud, mientras que Equal-Frequency Binning (EFB) [5] construye intervalos con el mismo número de observaciones. Aunque ambos métodos son simples y eficientes, su comportamiento depende en gran medida de la distribución de los datos y de la presencia de valores atípicos.

Otros métodos ampliamente utilizados se basan en segmentar el eje temporal. PAA [15] resume la serie mediante segmentos de igual longitud, sustituyendo cada uno por su valor medio. SAX [19] extiende esta idea transformando dichos promedios en símbolos a partir de puntos de corte estadísticos, lo que ha favorecido su uso en minería de series temporales. Por su parte, PLA [17] aproxima la señal mediante segmentos lineales, capturando mejor ciertas tendencias locales.

A pesar de su utilidad, estos enfoques presentan limitaciones importantes para el problema abordado en este trabajo. En general, dependen de ventanas o segmentaciones de tamaño fijo, lo que dificulta representar motivos de longitud variable. Además, pueden ser sensibles al ruido, a la elección de parámetros y a supuestos globales sobre la distribución de la señal. Como consecuencia, es posible perder estructura local relevante o introducir discretizaciones poco fieles cuando la serie presenta variabilidad heterogénea.

2.2. Inferencia gramatical

La inferencia gramatical busca aprender estructuras repetitivas a partir de secuencias simbólicas. Aunque su origen se encuentra en ámbitos como el procesamiento del lenguaje [8] o la compresión [23], también se ha aplicado al análisis de series temporales discretizadas con el fin de identificar patrones recurrentes de forma interpretable [24].

Entre los algoritmos más conocidos destacan Sequitur [22] y Re-Pair [18]. Sequitur construye una gramática de manera incremental sustituyendo pares repetidos por reglas, lo que permite comprimir la secuencia y revelar subestructuras recurrentes. Re-Pair sigue una estrategia voraz offline basada en frecuencia,

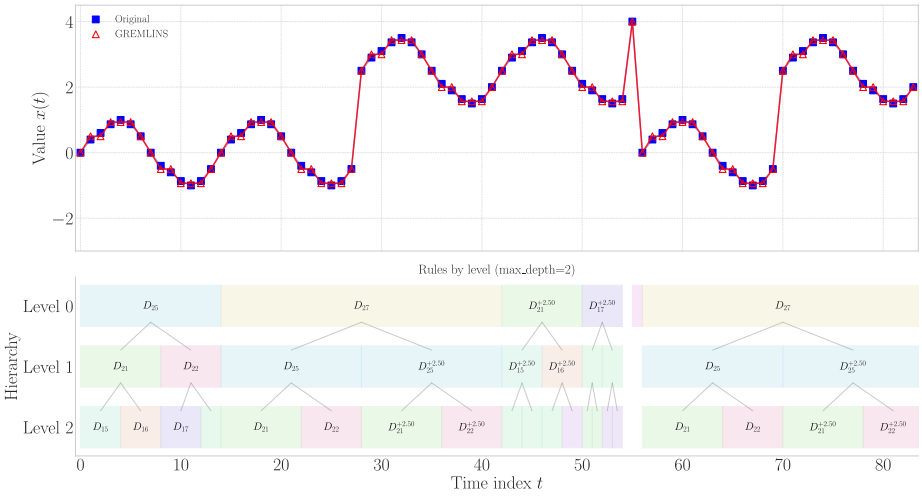


Figura 1. Resumen del flujo de GREMLINS. La serie temporal original se discretiza de forma adaptativa bajo un presupuesto de error ϵ, generando una secuencia simbólica compacta. Sobre esta representación, una fase de inferencia gramatical tolerante al ruido descubre y agrupa motivos recurrentes para producir etiquetas interpretables.

normalmente logrando gramáticas más compactas, aunque con mayor coste de memoria y de implementación. Ambos métodos han sido especialmente influyentes en marcos de descubrimiento de patrones simbólicos como GrammarViz [24].

Sin embargo, tanto Sequitur como Re-Pair dependen de coincidencias exactas entre símbolos o pares de símbolos. Esta característica limita su robustez cuando la secuencia de entrada contiene ruido o pequeñas variaciones estructurales, una situación frecuente en series temporales reales. Por ello, aunque la inferencia gramatical constituye una herramienta prometedora para descubrir patrones repetitivos, su efectividad depende en gran medida de la calidad de la discretización previa y de la capacidad del método para tolerar variabilidad.

3. Metodología propuesta

La Figura 1 resume el funcionamiento general de GREMLINS, un método para la extracción automática de patrones y el etiquetado simbólico de series temporales. La propuesta se articula en dos fases complementarias. En la primera, la serie temporal original se transforma en una representación simbólica compacta mediante una discretización adaptativa acotada por error. En la segunda, dicha secuencia se procesa mediante inferencia gramatical tolerante al ruido para descubrir y agrupar patrones recurrentes, generando una estructura jerárquica interpretable.

Frente a los enfoques clásicos de discretización, que suelen depender de ventanas de tamaño fijo o de supuestos globales sobre la distribución de los datos,

GREMLINS adopta una estrategia adaptativa guiada por error. Asimismo, frente a los métodos tradicionales de inferencia gramatical, basados en repeticiones exactas, el método incorpora tolerancia a variaciones estructurales y ruido, lo que favorece el descubrimiento de motivos de longitud variable en señales reales.

Con el fin de facilitar su aplicación práctica, GREMLINS es ligero en parámetros. La Fase 1 se controla mediante un único presupuesto de error relativo $\epsilon \in (0, 1)$, que regula el grado de compresión permitido. A partir de este parámetro se deriva de forma determinista una tolerancia τ, utilizada en la Fase 2 para agrupar patrones similares. En términos generales, valores pequeños de ϵ producen representaciones más finas y detalladas, mientras que valores mayores favorecen una discretización más compacta.

3.1. Fase 1: discretización adaptativa de la serie temporal

La primera fase transforma la serie temporal original en una secuencia simbólica compacta que preserve la estructura relevante de la señal. Su objetivo no es únicamente reducir la dimensionalidad, sino también generar una representación más robusta al ruido y más adecuada para la fase posterior de inferencia gramatical.

Para ello, GREMLINS parte del conjunto de valores distintos presentes en la serie, los ordena y los divide en segmentos iniciales mediante umbrales basados en cuantiles. Dentro de cada segmento, los valores se agrupan secuencialmente utilizando un umbral de aproximación local definido a partir de ϵ y del rango de valores del propio segmento. Cada grupo resultante se representa mediante un valor representativo, concretamente su mediana, que sustituye a los valores originales correspondientes en la serie.

Este procedimiento permite construir una discretización adaptativa al contexto local de la señal, evitando imponer ventanas temporales fijas o granularidades simbólicas prefijadas. Como resultado, la serie transformada reduce su complejidad simbólica sin perder las diferencias estructurales más relevantes entre regiones de la señal.

3.2. Fase 2: inferencia gramatical tolerante al ruido

A partir de la secuencia simbólica generada en la fase anterior, la segunda fase busca descubrir patrones recurrentes y organizarlos jerárquicamente mediante inferencia gramatical. La idea central consiste en identificar pares adyacentes de símbolos frecuentes, denominados digramas, y sustituirlos iterativamente por nuevos símbolos no terminales, generando así una representación más compacta y estructurada de la secuencia.

A diferencia de algoritmos clásicos como Sequitur o Re-Pair, que dependen de coincidencias exactas, GREMLINS incorpora un mecanismo de tolerancia para evitar que pequeñas fluctuaciones de discretización den lugar a reglas redundantes o casi duplicadas. En cada iteración, tras detectar los digramas frecuentes no solapados, se crean nuevas reglas gramaticales y se construye una secuencia comprimida. A continuación, los nuevos digramas generados se analizan mediante un

procedimiento de agrupamiento basado en densidad usando DBSCAN [9], con el fin de fusionar aquellos que representen transiciones estructuralmente similares.

La tolerancia de este agrupamiento no se introduce como un hiperparámetro adicional, sino que se calcula de forma determinista a partir de ϵ y de la escala empírica de la secuencia discretizada T', de modo que

$$\tau = \max\left(\tau_{\min}, \epsilon \cdot (\max(T') - \min(T'))\right).$$

De esta forma, el grado de tolerancia empleado en la inferencia gramatical queda alineado con el nivel de aproximación permitido en la discretización.

Cada grupo de digramas similares se normaliza mediante la selección de un representante, que se utiliza para reescribir el resto de elementos del grupo. Este proceso reduce la proliferación de reglas casi equivalentes y favorece una representación más estable e interpretable. La iteración continúa hasta que no quedan digramas frecuentes susceptibles de ser reemplazados. El resultado final de esta fase es una gramática jerárquica, una secuencia comprimida y un conjunto de grupos de motivos que permiten generar etiquetas simbólicas interpretables sobre la serie original.

En conjunto, GREMLINS combina una discretización adaptativa acotada por error con una inferencia gramatical robusta a pequeñas variaciones estructurales. La primera fase produce una representación simbólica compacta y fiel a la variabilidad local de la señal, mientras que la segunda explota dicha representación para descubrir y agrupar motivos recurrentes de longitud variable. Esta combinación permite obtener anotaciones simbólicas más flexibles e interpretables que las derivadas de enfoques basados en ventanas fijas o en repeticiones exactas.

4. Experimentación

Esta sección evalúa la efectividad de GREMLINS desde dos perspectivas. En primer lugar, se analiza la calidad de la discretización generada en la Fase 1, comparándola con métodos clásicos en términos de fidelidad de reconstrucción y compacidad simbólica. En segundo lugar, se estudia el comportamiento de la Fase 2 de inferencia gramatical frente a métodos establecidos de compresión y descubrimiento de patrones simbólicos. Además, se consideran tiempo de ejecución y uso de memoria para ofrecer una visión global del coste computacional.

4.1. Configuración experimental

La evaluación se realizó sobre 50 conjuntos de datos: 5 series sintéticas, diseñadas para probar propiedades estructurales, y 45 series reales de repositorios públicos. Los experimentos se ejecutaron en una máquina de 64 bits con Intel Xeon E5-2620 v2 a 2.10 GHz y 64 GB de RAM. Cada configuración se repitió 10 veces por conjunto y se reportaron valores medios.

En la evaluación de discretización se comparó GREMLINS con varios métodos representativos: EWB [7], EFB [5], PAA [15], SAX [19], PLA [17] con

segmentos y PLA-SWAB [16] , empleando implementaciones propias y de la librería *aeon* [21]. Como métricas principales se utilizaron el error cuadrático medio normalizado (NRMSE), para medir la fidelidad entre la serie original y la reconstruida, y el *Unique Ratio*, para cuantificar la compacidad simbólica.

En la evaluación de inferencia gramatical se comparó GREMLINS con dos métodos clásicos: Sequitur [22] y Re-Pair [18]. Estos algoritmos se aplicaron sobre las secuencias simbólicas generadas por los dos mejores discretizadores obtenidos en la etapa anterior.

Los hiperparámetros de los métodos se calibraron previamente mediante NSGA-II [3,6] sobre un subconjunto representativo de 10 conjuntos de datos, obteniendo una única configuración global por método. Posteriormente, estas configuraciones se mantuvieron fijas en todo el benchmark para evitar sobreajuste específico por conjunto de datos y garantizar una comparación más justa y reproducible.

Para el análisis estadístico se utilizó la librería StaTDS [20]. En primer lugar, se aplicó la prueba de Shapiro–Wilk para estudiar la normalidad de las distribuciones. Al rechazarse normalidad en las métricas analizadas, se recurrió a pruebas no paramétricas, empleando Friedman para detectar diferencias globales y Nemenyi para las comparaciones *post hoc*.

4.2. Evaluación de la discretización

La primera etapa del análisis estudia la capacidad de los distintos métodos de discretización para preservar la estructura de la serie original y, al mismo tiempo, producir representaciones simbólicas compactas. Para ello se consideraron conjuntamente NRMSE y *Unique Ratio*, ya que ambas métricas reflejan objetivos en tensión: una menor pérdida de información suele implicar representaciones menos compactas, mientras que una compresión más agresiva puede deteriorar la fidelidad de reconstrucción.

La Tabla 1 resume los resultados medios obtenidos. GREMLINS alcanza el mejor valor medio de NRMSE, lo que indica una reconstrucción especialmente fiel de la señal original, mientras que EWB obtiene también un comportamiento competitivo. Por su parte, SAX logra una compresión simbólica más fuerte, pero a costa de un deterioro notable en fidelidad. En conjunto, estos resultados sugieren que GREMLINS ofrece un equilibrio más sólido entre preservación estructural y reducción de complejidad simbólica.

Desde el punto de vista estadístico, el test de Friedman confirmó la existencia de diferencias significativas entre métodos ($p = 1{,}31 \times 10^{-10}$). El análisis *post hoc* de Nemenyi mostró que GREMLINS y EWB forman el grupo mejor posicionado cuando NRMSE y *Unique Ratio* se consideran de forma conjunta, sin diferencias significativas entre ambos. En cambio, SAX, EFB y las variantes basadas en PLA quedaron situados en grupos estadísticamente inferiores. En consecuencia, para la segunda parte del estudio se seleccionaron GREMLINS y EWB como discretizadores de referencia.

Tabla 1. Resultados medios y desviaciones típicas en la evaluación de discretización.

Algoritmo	NRMSE	Unique Ratio	Tiempo (s)	RAM (MB)
GREMLINS	**0.007 ± 0.003**	0.130 ± 0.266	0.171 ± 0.349	17.718 ± 37.257
EWB	0.021 ± 0.047	0.116 ± 0.051	0.095 ± 0.195	17.532 ± 37.199
EFB	0.024 ± 0.059	0.184 ± 0.047	0.163 ± 0.330	17.753 ± 37.264
PAA	0.110 ± 0.094	0.219 ± 0.486	**0.029 ± 0.054**	17.446 ± 37.561
SAX	0.195 ± 0.373	**0.072 ± 0.176**	0.083 ± 0.073	17.386 ± 37.182
PLA-Segm	0.031 ± 0.044	675.055 ± 2207.291	0.300 ± 0.264	**17.328 ± 37.320**
PLA-SWAB*	0.106 ± 0.073	10.189 ± 21.861	19.031 ± 28.672	0.776 ± 1.165

Nota: * El método superó el tiempo límite en uno o más conjuntos de datos. Las métricas se calcularon solo sobre las ejecuciones completadas.

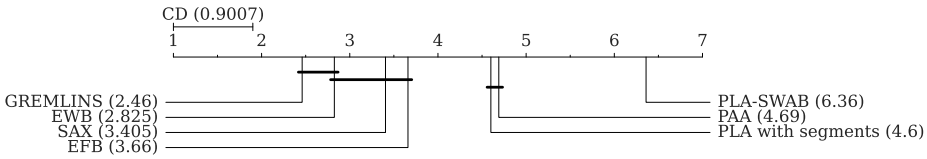


Figura 2. Diferencias críticas según Friedman–Nemenyi para la evaluación conjunta de NRMSE y *Unique Ratio*.

4.3. Evaluación de la inferencia gramatical

La segunda etapa estudia la capacidad de los métodos de inferencia gramatical para producir representaciones compactas e interpretables a partir de las secuencias discretizadas. Para ello se comparó GREMLINS con Sequitur y Re-Pair empleando como entrada las representaciones generadas por GREMLINS y EWB. Las métricas consideradas fueron la tasa de compresión, el tiempo total de ejecución y el uso máximo de memoria.

La Tabla 2 resume los resultados obtenidos. GREMLINS alcanzó la mejor tasa media de compresión y también el mejor tiempo de ejecución, mostrando un comportamiento especialmente competitivo cuando se aplica sobre su propia discretización. Sequitur presentó resultados razonables, aunque con mayor coste temporal y de memoria. Re-Pair, por su parte, fue el método menos favorable en términos globales, especialmente en tiempo de ejecución, llegando incluso a superar el límite temporal en algunos conjuntos de datos.

El análisis estadístico confirmó diferencias significativas en las tres métricas evaluadas: compresión ($p = 5{,}56 \times 10^{-11}$, ver Figura 3), memoria ($p = 3{,}92 \times 10^{-11}$, ver Figura 5) y tiempo ($p = 7{,}39 \times 10^{-11}$, ver Figura 4). Los diagramas de diferencias críticas muestran que GREMLINS ocupa la mejor posición en compresión y tiempo, mientras que Re-Pair queda sistemáticamente relegado a posiciones inferiores. En conjunto, estos resultados indican que la combinación de la discretización adaptativa de la Fase 1 y la inferencia gramatical tolerante de la Fase 2 proporciona representaciones más compactas y eficientes que las obtenidas con alternativas clásicas.

Tabla 2. Resultados medios y desviaciones típicas en la evaluación de inferencia gramatical.

Algoritmo	Compresión	Tiempo (s)	RAM (MB)
GREMLINS	**0.213 ± 0.205**	**12.520 ± 26.400**	39.997 ± 80.665
Sequitur + GREMLINS (Fase 1)	0.222 ± 0.208	39.976 ± 86.681	94.475 ± 194.478
Re-Pair* + GREMLINS (Fase 1)	0.299 ± 0.208	50.754 ± 200.167	**32.051 ± 127.323**
GREMLINS (Fase 2) + EWB	0.306 ± 0.270	31.256 ± 70.599	126.202 ± 483.693
Sequitur + EWB	0.317 ± 0.272	40.735 ± 88.751	129.364 ± 246.136
Re-Pair* + EWB	0.406 ± 0.264	133.291 ± 574.557	121.613 ± 304.634

Nota: * El método superó el tiempo límite en uno o más conjuntos de datos. Las métricas se calcularon solo sobre las ejecuciones completadas.

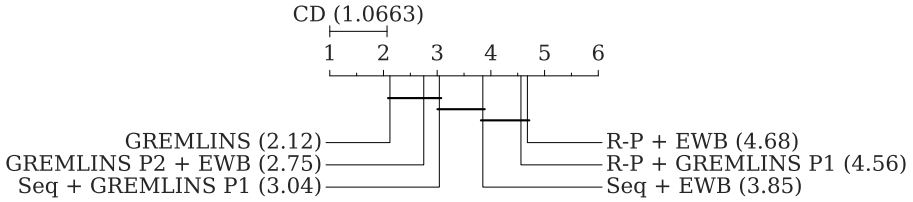


Figura 3. Diferencias críticas según Friedman–Nemenyi para la tasa de compresión.

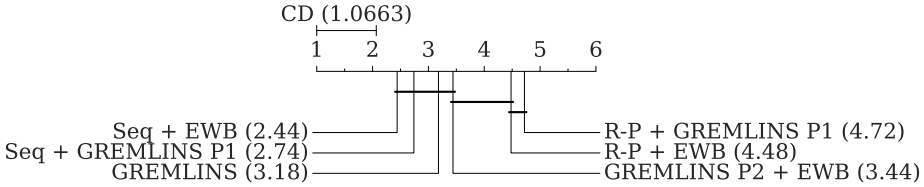


Figura 4. Diferencias críticas según Friedman–Nemenyi para el tiempo.

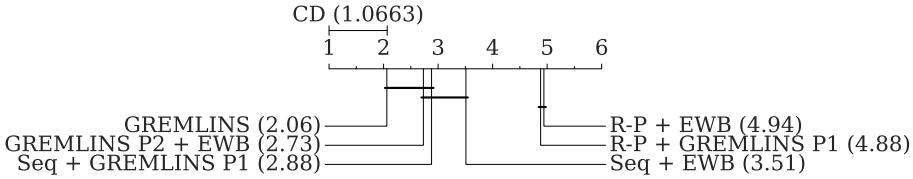


Figura 5. Diferencias críticas según Friedman–Nemenyi para la memoria pico.

5. Conclusiones

En este trabajo se ha presentado GREMLINS, un método para el etiquetado automático de series temporales que combina una discretización simbólica adaptativa acotada por error con una fase de inferencia gramatical tolerante a pequeñas variaciones estructurales. La propuesta evita el uso de ventanas fijas y

se controla mediante un único parámetro principal, ϵ, del que se deriva de forma determinista la tolerancia empleada en la fase de agrupamiento.

Los resultados experimentales muestran que GREMLINS proporciona un equilibrio competitivo entre fidelidad, compacidad y eficiencia, superando en conjunto a alternativas clásicas de discretización e inferencia gramatical. En particular, los resultados confirman que la discretización adaptativa no actúa solo como preprocesado, sino como un componente esencial para mejorar la calidad de la representación simbólica y favorecer el descubrimiento de motivos recurrentes.

Como trabajo futuro, se plantea extender el método a escenarios en línea e integrarlo con técnicas de aprendizaje automático para tareas interpretables sobre series temporales.

Agradecimientos

Esta investigación ha sido financiada parcialmente por los proyectos PID2023-148396NB-I00 del Ministerio de Ciencia e Innovación y el Fondo Europeo de Desarrollo Regional, por el proyecto ProyExcel-0069 de la Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía, y por una ayuda predoctoral FPU23/02732 del Ministerio de Universidades.

Referencias

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data mining and knowledge discovery **31**, 606–660 (2017)
2. Blalock, D.W., Guttag, J.V.: Extract: Strong examples from weakly-labeled sensor data. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). pp. 799–804. IEEE (2016)
3. Blank, J., Deb, K.: Pymoo: Multi-objective optimization in python. Ieee access **8**, 89497–89509 (2020)
4. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A review on outlier/anomaly detection in time series data. ACM computing surveys (CSUR) **54**(3), 1–33 (2021)
5. Chaudhari, P., Rana, D.P., Mehta, R.G., Mistry, N.J., Raghuwanshi, M.M.: Discretization of temporal data: a survey. arXiv preprint arXiv:1402.4283 (2014)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
7. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Machine learning proceedings 1995, pp. 194–202. Elsevier (1995)
8. D'Ulizia, A., Ferri, F., Grifoni, P.: A survey of grammatical inference methods for natural language learning. Artificial Intelligence Review **36**, 1–27 (2011)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)

10. Fredriksson, T., Mattos, D.I., Bosch, J., Olsson, H.H.: Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In: International conference on product-focused software process improvement. pp. 202–216. Springer (2020)
11. García-Treviño, E.S., Barria, J.A.: Structural generative descriptions for time series classification. IEEE transactions on cybernetics **44**(10), 1978–1991 (2014)
12. Guo, H., Wang, L., Liu, X., Pedrycz, W.: Trend-based granular representation of time series and its application in clustering. IEEE Transactions on Cybernetics **52**(9), 9101–9110 (2021)
13. Gupta, P., Nandakumar, S., Gupta, M., Panda, G.: Data programming enabled weak supervised labeling for ecg time series. Biomedical Signal Processing and Control **87**, 105540 (2024)
14. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
15. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems **3**, 263–286 (2001)
16. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: Proceedings 2001 IEEE international conference on data mining. pp. 289–296. IEEE (2001)
17. Keogh, E.J., Pazzani, M.J.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Kdd. vol. 98, pp. 239–243 (1998)
18. Larsson, N.J., Moffat, A.: Off-line dictionary-based compression. Proceedings of the IEEE **88**(11), 1722–1732 (2000)
19. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Mining and knowledge discovery **15**, 107–144 (2007)
20. Luna, C., Moya, A.R., Luna, J.M., Ventura, S.: Statds library: Statistical tests for data science. Neurocomputing **595**, 127877 (2024)
21. Middlehurst, M., Ismail-Fawaz, A., Guillaume, A., Holder, C., Guijo-Rubio, D., Bulatova, G., Tsaprounis, L., Mentel, L., Walter, M., Schäfer, P., Bagnall, A.: aeon: a python toolkit for learning from time series. Journal of Machine Learning Research **25**(289), 1–10 (2024), `http://jmlr.org/papers/v25/23-1444.html`
22. Nevill-Manning, C.G., Witten, I.H.: Identifying hierarchical structure in sequences: A linear-time algorithm. Journal of Artificial Intelligence Research **7**, 67–82 (1997)
23. Sakamoto, H.: Grammar compression: Grammatical inference by compression and its application to real data. In: International Conference on Grammatical Inference. pp. 3–20. PMLR (2014)
24. Senin, P., Lin, J., Wang, X., Oates, T., Gandhi, S., Boedihardjo, A.P., Chen, C., Frankenstein, S.: Grammarviz 3.0: Interactive discovery of variable-length time series patterns. ACM Transactions on Knowledge Discovery from Data (TKDD) **12**(1), 1–28 (2018)
25. Tok, W.H., Bahree, A., Filipi, S.: Practical Weak Supervision. .º'Reilly Media, Inc."(2021)
26. Wilson, G., Doppa, J.R., Cook, D.J.: Multi-source deep domain adaptation with weak supervision for time-series sensor data. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1768–1778 (2020)
27. Zhou, Z.H.: A brief introduction to weakly supervised learning. National science review **5**(1), 44–53 (2018)