

Predicción de series de tiempo en sistemas dinámicos caóticos mediante redes generativas adversarias

Jesús Alejandro Salazar-González^{1,2}, Erik Molino-Minero-Re^{1,2}

¹ Posgrado en Ciencia e Ingeniería de la Computación, UNAM, Mérida, Yucatán,
México

² Unidad Académica del IIMAS, UNAM, en Yucatán, Mérida, Yucatán,
México

`alejandro.salazar9812@comunidad.unam.mx`

Resumen. La predicción de series de tiempo en sistemas dinámicos caóticos representa un reto debido a su alta sensibilidad a las condiciones iniciales y a las no linealidades que caracterizan su comportamiento. En este trabajo se propone evaluar el modelo *GAN-LSTM-Attention*, desarrollado para predecir índices financieros, con el objetivo de analizar su capacidad predictiva bajo condiciones altamente no lineales. La evaluación se realiza sobre un conjunto de datos sintéticos generados a partir de sistemas caóticos clásicos. Se analizan distintas configuraciones de tamaño de ventana de entrada y salida, así como esquemas de predicción autorregresivos y no autorregresivos. El enfoque no autorregresivo muestra un buen desempeño, alcanzando valores de coeficiente de determinación (R^2) y correlación de Pearson (r) superiores a 0.80 y 0.90, respectivamente. A pesar de eso, se encuentra que existe un deterioro significativo en la calidad de la predicción al incrementar la longitud de la ventana de predicción, lo cual se debe a la acumulación del error. Asimismo, se observó que un incremento del tamaño de la memoria no produce mejoras significativas en el desempeño.

Keywords: Redes neuronales, redes generativas adversarias, predicción, sistemas dinámicos, series de tiempo.

Time Series Prediction in Chaotic Dynamical Systems Using Adversarial Generative Networks

Abstract. Predicting time series in chaotic dynamical systems presents a challenge due to their high sensitivity to initial conditions and the nonlinearities that characterize their behavior. This paper proposes to evaluate the *GAN-LSTM-Attention* model, developed to predict financial indices, with the aim of analyzing its predictive capacity under highly

nonlinear conditions. The evaluation is performed on a set of synthetic data generated from classical chaotic systems. Different configurations of input and output window sizes are analyzed, as well as autoregressive and non-autoregressive prediction schemes. The non-autoregressive approach shows good performance, achieving coefficient of determination (R^2) and Pearson correlation (r) values greater than 0.80 and 0.90, respectively. Despite this, a significant deterioration in prediction quality is found when the prediction window length is increased, due to error accumulation. Furthermore, it was observed that increasing the memory size does not produce significant performance improvements.

Keywords: Neural networks, generative adversarial networks, prediction, dynamical systems, time series.

1. Introducción

La predicción de series de tiempo es un proceso importante para la toma de decisiones en un gran número de aplicaciones, que pueden ir desde la identificación temprana de una afección cardíaca, el pronóstico del clima e inclusive la detección de anomalías de maquinaria industrial. La manera tradicional de trabajar modelos en sistemas dinámicos consiste en estudiar el comportamiento físico del problema y en plantear ecuaciones diferenciales que lo describan. Sin embargo, en la práctica, la cantidad de variables que describan la realidad a menudo es desconocida, o ni siquiera podemos medirlas todas. También las condiciones de frontera introducen no linealidades que pueden elevar significativamente la dificultad de encontrar una solución analítica. Por ello, técnicas de aprendizaje automático (como lo son las redes neuronales) en los últimos años han tenido gran impacto en este campo debido a que no es necesario plantear un modelo analítico explícito y depende únicamente de los datos que tengamos del sistema a analizar.

Entre las variedades de técnicas del aprendizaje automático se encuentran las redes neuronales, que tienen la capacidad de reproducir características no lineales presentes en nuestro proceso de estudio. En el campo de los sistemas dinámicos, han demostrado ser eficaces para realizar predicciones de sistemas caóticos, tales como el sistema de Lorenz y el registro histórico del nivel del agua del Gran Lago Salado en Utah, Estados Unidos, ambos estudiados en [2]. En ese trabajo se utiliza una versión modificada del exponente de Lyapunov para comparar sus modelos, lo que permite estimar un horizonte de predicción en sistemas dinámicos caóticos.

En el campo del aprendizaje automático, las redes generativas están diseñadas principalmente para crear datos realistas a partir de muestras representativas. A partir de esta perspectiva, el presente artículo explora el potencial de las redes generativas adversarias para la predicción de series de tiempo mediante la evaluación de una arquitectura *GAN-LSTM-Attention* aplicada a sistemas dinámicos caóticos. Dicha arquitectura, previamente utilizada en la predicción de datos financieros, integra módulos LSTM (*Long Short-Term Memory*) y

mecanismos de atención para capturar dependencias temporales relevantes en la señal. Se analizan distintos tamaños de ventana de entrada y salida (denominados tamaño de memoria y horizonte de predicción, respectivamente), además de esquemas de predicción autorregresivos y no autorregresivos.

2. Trabajo relacionado

Los sistemas caóticos representan un gran desafío en la predicción debido a la alta sensibilidad a las condiciones iniciales. Los métodos tradicionales de predicción incluyen sistemas determinísticos, que buscan una aproximación a la solución de ecuaciones que modelan la dinámica sistema. Un ejemplo de uso de aprendizaje automático en este ámbito es GenCast, empleada para la predicción del clima, un modelo de difusión probabilística que ha logrado superar en el 97.2% de las métricas evaluadas al sistema de predicción por conjuntos (ENS por sus siglas en inglés) del Centro Europeo de Predicciones Meteorológicas a Plazo Medio [10]. Con ello se destaca que los modelos de aprendizaje automático poseen un gran potencial para la predicción de sistemas dinámicos.

Debido a la naturaleza recurrente de los sistemas dinámicos, las redes LSTM han sido ampliamente utilizadas para modelar dependencias temporales en series de tiempo [12]. Las arquitecturas con LSTM se han utilizado en conjunto de transformaciones matemáticas o métodos estadísticos para modelar series de tiempo, pero se ha observado que estos enfoques no siempre se comportan bien con dinámicas altamente sensibles a las condiciones iniciales o caóticas [9].

En lo que respecta al análisis de series de tiempo, usualmente se añaden otros módulos que ayudan a extraer características locales y globales del sistema. Fan H. [4], en su artículo, propone el uso de un reservorio de cómputo que actualice esporádicamente el sistema real; con ello evita que los sistemas sensibles diverjan. Otra manera de extraer características es empleando la técnica de transformar las señales en imágenes, por ejemplo, estas pueden ser generadas por mapas de recurrencias, las cuales en conjunto de alguna red convolucional suelen ser útiles para la predicción [8].

En este contexto, también se han empleado auto-decodificadores variacionales, que transforman los datos a un espacio latente de menor dimensión, usualmente coherente con una distribución gaussiana. Esto permite extraer características relevantes e incluso logra la reducción del ruido. TimeVAE es un ejemplo de este tipo de red [3].

Por otro lado, las redes generativas adversarias (GAN) han cobrado relevancia en los últimos años. Estas emplean dos redes neuronales: la primera es un generador especializado en generar datos sintéticos a partir de ruido aleatorio, y la segunda es un discriminador que identifica si los datos de entrada son reales o sintéticos [1]. Al entrenar ambas redes en conjunto, se obtiene un modelo capaz de generar datos que parecen reales a partir de ruido aleatorio. Se han utilizado en la detección de anomalías en datos de telemetría espacial [11], en donde la idea principal es dejar que las redes aprendan la representación latente de datos normales; después de ello, se utilizará el discriminador para

identificar entre datos anómalos y normales. Asimismo, las GAN se han aplicado en modelos de hornos rotatorios; en estos se combinan varias herramientas que incluyen la reconstrucción de espacios de fases, la generación de mapas globales de recurrencia y el uso de LSTM para la obtención de características [5].

En [7] se discute un modelo denominado GAN-LSTM-Attention, aplicado a la predicción de precios de acciones consideradas representativas del mercado (S&P 500, Apple, AMD y Google). La predicción bursátil se caracteriza por su fuerte dinamismo, no lineal y volátil, lo que la convierte en una tarea compleja. El estudio demuestra que la combinación de mecanismos generativos, redes recurrentes y módulos de atención puede ofrecer mejoras en la precisión y la robustez de la predicción de datos nuevos.

En dicho modelo, el generador emplea una capa LSTM para capturar dependencias a largo plazo, un mecanismo de atención para asignar mayor peso a la información temporal más relevante, y finalmente capas densas para producir la salida. El discriminador utiliza capas convolucionales unidimensionales para extraer patrones locales en la secuencia, seguidas de capas densas que permiten clasificar entre datos reales y generados. Un aspecto importante de este enfoque es que únicamente utiliza datos temporales como entrada; no requiere representaciones visuales como mapas de recurrencia. Por lo tanto, la cantidad de datos necesaria para entrenar esta arquitectura es considerablemente menor que la del modelo GRP-lstmGAN, lo que representa una ventaja.

3. Metodología

Las fases principales de este trabajo se representan en el diagrama esquemático de la Figura 1. La primera fase consiste en seleccionar la base de datos y procesarla. Posteriormente, con el propósito de evaluar distintos horizontes de predicción y tamaños de memoria, se realizan siete experimentos.

El entrenamiento de la red generativa adversaria se basa en la arquitectura propuesta por [7], utilizada para la predicción de los precios de las acciones de la bolsa. Para la evaluación del desempeño del modelo, se emplea un conjunto de 4 métricas, entre las cuales se encuentran el coeficiente de determinación (R^2), el coeficiente de correlación de Pearson (r), el error cuadrático medio ($RMSE$) y la Deformación Temporal Dinámica (dynamic time warping, DTW). Finalmente, se analizan los resultados de los experimentos.

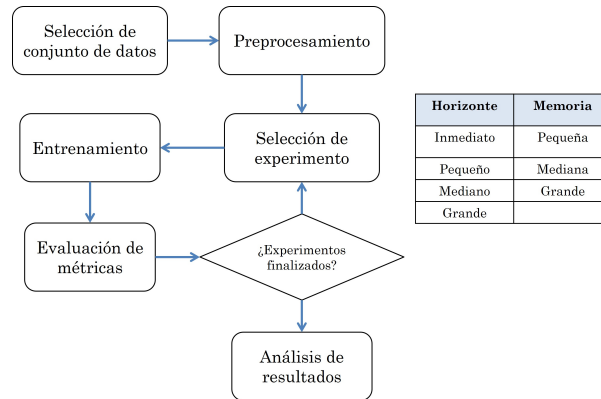


Fig. 1. Flujo de trabajo.

Selección de la base de datos y preprocesamiento. Un sistema dinámico se caracteriza por la evolución de su estado a lo largo del tiempo, de acuerdo con ciertas reglas. Se conoce una gran variedad de sistemas de este tipo, por ejemplo, la oscilación de un péndulo, el perfil de velocidades de un fluido e incluso el famoso sistema de Lorenz, por lo que la selección de un conjunto de datos que sea útil para el entrenamiento de nuestra red y pueda adaptarse fácilmente a sistemas no vistos es de suma importancia.

En el artículo [6] se emplea el modelo *PANDA*, entrenado con una base de datos sintética generada mediante algoritmos evolutivos. El proceso inicia con una población base de sistemas caóticos clásicos (como los de Lorenz, Rössler, Chua, entre otros), cuyos parámetros y condiciones iniciales han sido ajustados para garantizar comportamientos caóticos estables. A partir de esta población, se aplican operadores evolutivos, entre los cuales destacan la mutación, mediante la adición de ruido gaussiano a los parámetros, y la recombinación, en la que dos sistemas se acoplan mediante un producto sesgado aditivo (*skew-product*), produciendo un nuevo sistema híbrido.

Para el entrenamiento, se seleccionan aleatoriamente 100 series de tiempo tridimensionales de 4086 pasos, provenientes del conjunto de datos original de [6]. Dependiendo del experimento, se aplica una ventana deslizante para generar un conjunto de datos con un tamaño de entrada (memoria) y un horizonte de predicción definidos. A cada serie se le aplica un proceso de estandarización con $\mu = 0$ y $\sigma = 1$. Asimismo, se construye el conjunto de datos de prueba a partir de 20 series de tiempo no vistas por los modelos.

Experimentos. Para evaluar los resultados, se realizan experimentos sobre el horizonte de predicción y el tamaño de memoria. En el primero de estos se selecciona una ventana de entrada de 100 pasos y se varía el tamaño de salida entre los descritos en la tabla 1. En el caso del tamaño de memoria, se utilizan

las características descritas en la tabla 2, en la que el horizonte de predicción se fija en 10 muestras a futuro.

Tabla 1. Características de experimentos para predicción. Se fija un tamaño de memoria de 100 pasos y se varía el número de pasos para predecir.

Abreviatura	Horizonte de predicción	Pasos a futuro	Porcentaje de memoria
HI	Inmediato	1	1 %
HS	Pequeño	10	10 %
HM	Mediano	25	25 %
HL	Largo	50	50 %

Tabla 2. Características de los experimentos de tamaño de memoria. Se fija un horizonte de predicción a 10 pasos futuros y se varía el tamaño de memoria.

Abreviatura	Tamaño de memoria	Pasos en el tiempo	Porcentaje de serie
MS	Pequeña	100	2.5 %
MM	Mediana	200	5 %
MG	Grande	400	10 %

Entrenamiento. El entrenamiento se realiza tomando como base el modelo propuesto en [7]. Este cuenta con dos fases alternadas: en la primera se entrena al discriminador y se congela al generador, con el fin de que el discriminador sea capaz de diferenciar datos reales y sintéticos. En la segunda fase, se entrena al generador mientras el discriminador se congela, para que el generador pueda aproximarse cada vez más a la distribución de datos reales. Las funciones de pérdida incluyen la entropía cruzada binaria para el discriminador, mientras que el generador incorpora una pérdida proveniente del error cuadrático medio, con la adición de una pérdida adversarial. La arquitectura del generador está conformada por:

1. **Una capa de entrada lineal:** que toma el tamaño de la memoria seleccionada.
2. **Dos capas LSTM:** la primera con 128 unidades y la segunda con 64, ambas con retorno de secuencias, encargadas de capturar dependencias temporales de distintas profundidades en la serie de entrada.
3. **Mecanismo de atención:** se calcula un puntaje mediante una capa densa y se normaliza con una función softmax para obtener los pesos de atención. Estos pesos se aplican a las salidas de las LSTM y se obtiene un vector de contexto mediante una suma ponderada.

4. **Capas densas:** una capa con 128 neuronas y activación ReLU para transformar el contexto y extraer representaciones no lineales.
5. **Capa de salida:** una capa densa que realiza la predicción; su tamaño depende del horizonte de predicción seleccionado.

Mientras que la arquitectura del discriminador está conformada por:

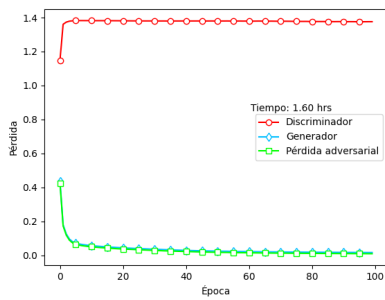
1. **Una capa de entrada lineal:** que toma el tamaño de la memoria seleccionada.
2. **Tres capas convolucionales 1D:** con 64, 128 y 256 filtros respectivamente, todas con activación ReLU, encargadas de extraer patrones locales en las secuencias predichas.
3. **Capa de aplanado:** convierte las características extraídas en un vector unidimensional.
4. **Capas densas:** una con 128 neuronas y otra con 64, ambas con activación ReLU, para reducir la dimensionalidad y aprender representaciones más abstractas.
5. **Capa de salida sigmoide:** produce una probabilidad que indica si la secuencia de entrada corresponde a datos reales o generados.

Evaluación de métricas. Una vez finalizado el entrenamiento, se realizan las predicciones bajo dos esquemas, no autorregresivo y autorregresivo. En el primero de ellos, el modelo recibe únicamente datos reales de entrada, mientras que el segundo se alimenta de sus propias predicciones. Para predecir las series de tiempo se emplea una ventana deslizante, debido a que en la mayoría de los experimentos el horizonte de predicción produce múltiples pasos hacia adelante; algunas predicciones se solapan. El valor de predicción será el promedio de las predicciones solapadas. Una vez generada la serie de predicción, se determinan las métricas R^2 , r , $RMSE$ y DTW .

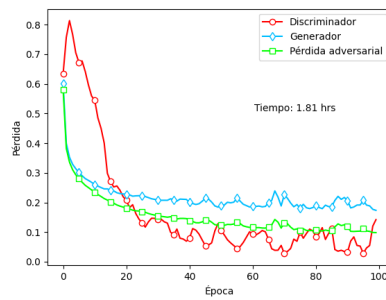
4. Resultados y discusión

Funciones de pérdida. Se grafican los resultados obtenidos durante el entrenamiento de los 7 modelos propuestos. En las figuras 2 y 3 se muestran los resultados de los experimentos de horizonte de predicción y de memoria, respectivamente. En el caso del horizonte de predicción, podemos notar en la figura 2 (b) que, en un principio, el discriminador presenta una pérdida demasiado alta, pero al continuar con el entrenamiento esta decrece, lo que indica que el modelo está aprendiendo a reconocer muestras reales de las falsas. En torno a la época 40 se observa que esta pérdida oscila, lo cual puede interpretarse como que el generador ha llegado a un punto en el que es capaz de engañar al discriminador con mayor facilidad.

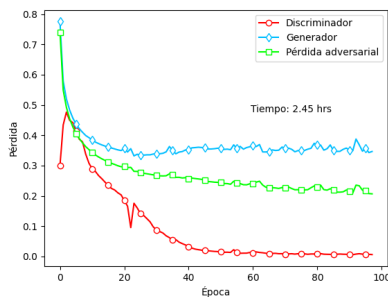
Por otro lado, en las figuras 2 (a) se observa que el discriminador no es capaz de distinguir muestras reales de las falsas, y en 2 (c) y (d) este obtiene una dominancia muy fuerte, lo que provoca que el generador no sea capaz de mejorar a partir de cierto punto.



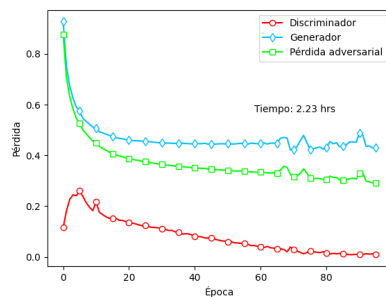
(a) HI



(b) HS



(c) HM



(d) HL

Fig. 2. Curvas de pérdida de horizonte de predicción.

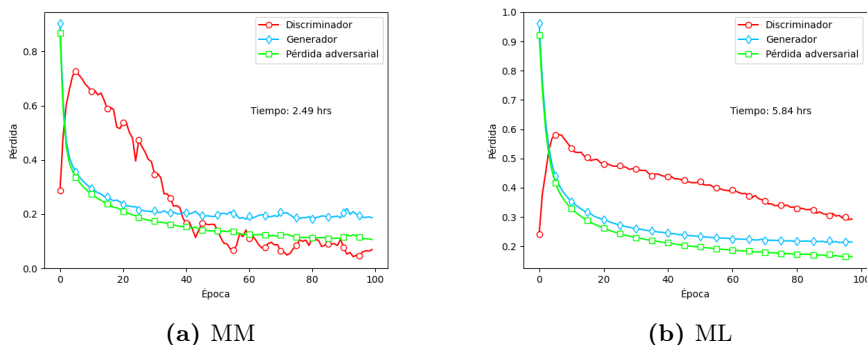


Fig. 3. Curvas de pérdida de tamaño de memoria.

Predicciones. En la figura 4 se encuentran los resultados obtenidos para los esquemas autorregresivos y no autorregresivos respectivamente. En el caso no autorregresivo, R^2 obtiene valores positivos, alcanzando un máximo de 0.816 en el modelo HI, lo que indica que el modelo sí logra capturar la dinámica del sistema. En este mismo caso se observa un valor alto del coeficiente de correlación de Pearson, alcanzando un 0.904, indicando una buena correspondencia entre datos predichos y reales.

En el caso autorregresivo, los valores de R^2 son negativos en la mayoría de los casos, lo que indica que el modelo no logra captar la dinámica del sistema, lo cual es de esperarse en sistemas caóticos, donde cambios pequeños en las condiciones acumulan un error rápidamente, provocando que las predicciones se vayan deteriorando. Asimismo, se observa que el coeficiente de Pearson es cercano a 0, lo que indica una baja correlación entre la predicción y los datos reales. Algo que destaca es que la métrica DTW parece encontrar un mínimo en el modelo HM, lo que sugiere que un horizonte de predicción intermedio logra capturar mejor la dinámica temporal de las series.

En lo que respecta al tamaño de la memoria, en el esquema autorregresivo el tamaño no mejora significativamente los resultados en todas las métricas; por lo tanto, en este caso, la mejor opción es escoger un tamaño de memoria pequeño, pues el tiempo de entrenamiento es menor (1.81 vs. 5.84 hrs).

Los resultados anteriores muestran que el esquema no autorregresivo es claramente superior, especialmente cuando trabajamos con horizontes cortos. El esquema autorregresivo, aunque es interesante para simular a largo plazo, sufre de una acumulación de errores que no logra capturar la dinámica de los sistemas caóticos. Una forma de mejorar estos resultados podría ser emplear una estrategia en la que el sistema reciba una actualización esporádica de los datos, similar a la mencionada por [4].

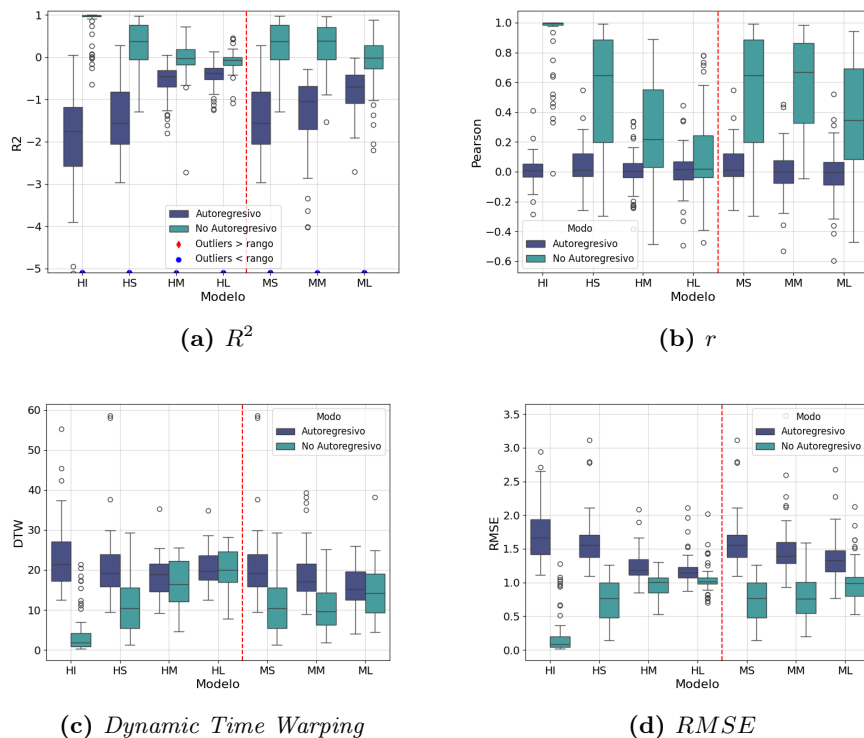


Fig. 4. Diagramas de caja de las métricas obtenidas en el conjunto de prueba para los modelos entrenados. Se muestran ambos esquemas de predicción. La línea roja punteada separa las pruebas de horizonte de predicción y de memoria. Para fines de visualización se señalan los valores atípicos que están fuera del rango de la imagen.

En la figura 5 se muestra cómo se superponen las gráficas de predicción con la serie real de 3 series provenientes del conjunto de pruebas (se muestra una serie distinta por fila). En el lado izquierdo de la figura se muestra un esquema no autorregresivo y en el derecho el autorregresivo, donde se aprecia con claridad la diferencia en la calidad de predicción.

5. Conclusiones

Este trabajo aborda la predicción de series de tiempo en sistemas dinámicos caóticos mediante el uso de redes generativas adversarias, en particular a través de un modelo basado en *GAN-LSTM-Attention*. El objetivo es evaluar el potencial de las redes generativas para la predicción en sistemas altamente no lineales y sensibles a las condiciones iniciales. Dicha arquitectura ha sido previamente

empleada en la predicción de precios de acciones, caracterizados por su comportamiento volátil. En este trabajo, se explora su uso en sistemas caóticos para estudiar su capacidad de modelar dinámicas complejas en escenarios distintos al financiero.

Para evaluar la arquitectura se recurrió a un conjunto de datos empleado para entrenar un algoritmo denominado *PANDA*, que contiene datos sintéticos generados mediante algoritmos evolutivos, cuya población inicial estuvo compuesta por sistemas caóticos clásicos. Se analizan diversas configuraciones de horizonte y memoria bajo dos esquemas de predicción distintos: no autorregresivo y autorregresivo.

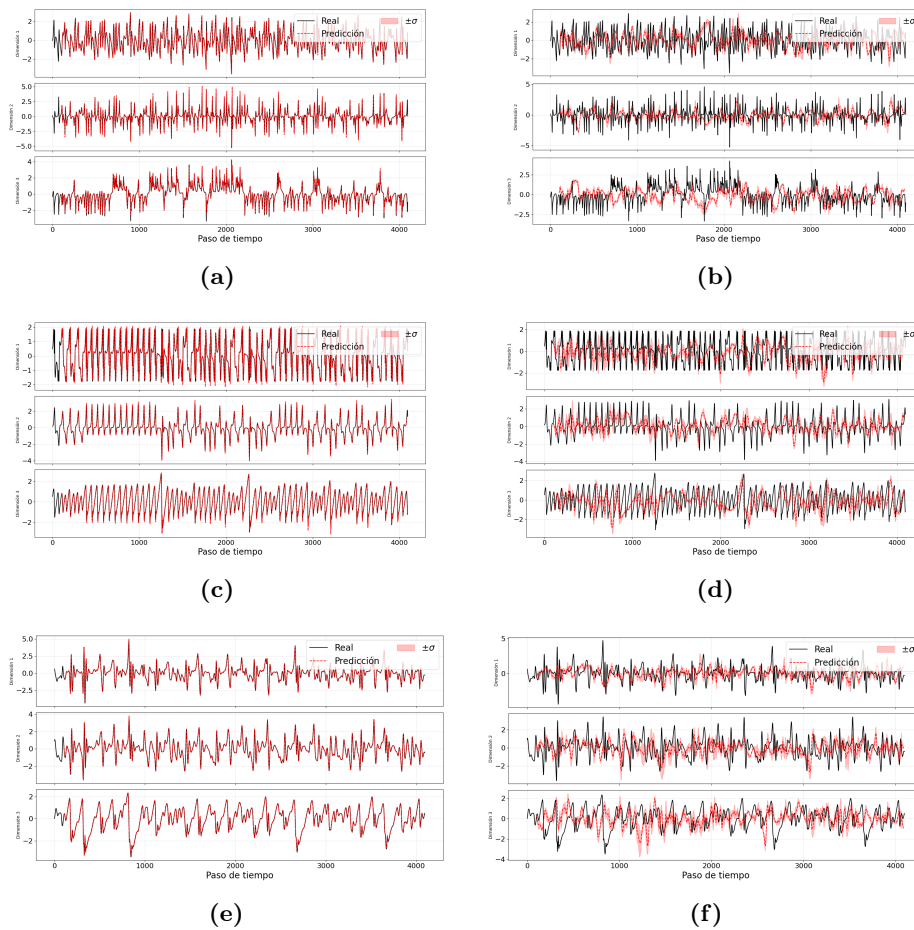


Fig. 5. Comparación entre los ambos esquemas de predicción, paneles izquierdos muestran esquema no autorregresivo mientras que los derechos el autorregresivo. (a), (c) y (e) muestran el modelo HI mientras que (b), (d) y (f) emplean el modelo HM.

Los resultados muestran que, bajo el esquema no autorregresivo, los valores R^2 y r se encuentran por encima de 0.80 y 0.90 respectivamente, indicando buen desempeño predictivo. A pesar de esto, al incrementar el horizonte de predicción, la calidad de los resultados disminuye rápidamente debido a la acumulación de errores, lo que constituye un punto crítico en un sistema sensible a las condiciones iniciales. En lo que respecta al incremento de la memoria, no se observa un aumento significativo en la calidad de la predicción, por lo que se sugiere utilizar configuraciones más pequeñas para reducir el costo computacional. Como trabajo a futuro, se propone explorar enfoques basados en reservorios de cómputo que permitan agregar información de las observaciones en tiempo real, con el objetivo de minimizar la propagación del error y mejorar la precisión de las predicciones a largo plazo.

Agradecimientos. Este proyecto fue apoyado por el programa UNAM-PAEP IG101725. JASG agradece el apoyo económico de SECIHTI.

Referencias

1. Abdel-Basset, M., Moustafa, N., Hawash, H.: Generative adversarial networks (gans). In: *Deep Learning Approaches for Security Threats in IoT Environments*, pp. 271–285. Wiley (2022)
2. Alfaro, M., Fuertes, G., Vargas, M., Sepúlveda, J., Veloso-Poblete, M.: Forecast of chaotic series in a horizon superior to the inverse of the maximum lyapunov exponent. *Complexity* 2018, 1452683 (2018)
3. Desai, A., Freeman, C., Wang, Z., Beaver, I.: Timevae: A variational auto-encoder for multivariate time series generation (2021)
4. Fan, H., Jiang, J., Zhang, C., Wang, X., Lai, Y.C.: Long-term prediction of chaotic systems with machine learning. *Physical Review Research* 2(1), 012080 (2020)
5. Gao, Y., Sosnowski, M., Grabowska, K., Skrobek, D., Moeen Uddin, G., Kulakowska, A., Zylka, A., El Fil, B., Hu, W., Mao, Z.: Forecasting for chaotic time series based on grp-lstmgan model: Application to temperature series of rotary kiln. *Entropy* 25(1), 52 (2023)
6. Lai, J., Bao, A., Gilpin, W.: Panda: A pretrained forecast model for chaotic dynamics (2025)
7. Li, P., Wei, Y., Yin, L.: Research on stock price prediction method based on the gan-lstm-attention model. *Computers, Materials and Continua* 82(1), 609–625 (2025)
8. Li, X., Kang, Y., Li, F.: Forecasting with time series imaging. *Expert Systems with Applications* 160, 113680 (2020)
9. Mahmoudi, A.: Investigating LSTM-based time series prediction using dynamic systems measures. *Evolving Systems* (2025), <https://link.springer.com/article/10.1007/s12530-025-09703-y>
10. Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T.R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P., Lam, R., Willson, M.: Probabilistic weather forecasting with machine learning. *Nature* 637(8044), 84–90 (2024)
11. Song, Y., Yu, J., Tang, D., Han, D., Wang, S.: Telemetry data-based spacecraft anomaly detection using generative adversarial networks. In: *ICSMMD 2020*. pp. 297–301 (2020)

12. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation* 31(7), 1235–1270 (2019)