



**I
N
A
O
E**

Meta-aprendizaje Auto-organizado Temporal para Predicción a Largo Plazo de Series de Tiempo Caóticas

Rigoberto Fonseca-Delgado, María del Pilar Gómez-Gil

Reporte Técnico No. CCC-14-003
6 de marzo de 2014

© Coordinación de Ciencias Computacionales
INAOE

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.



Resumen

La predicción a largo plazo de series de tiempo de sistemas caóticos es una tarea difícil pero requerida en diversos dominios como economía, medicina, astronomía, etc. En los últimos años, varias investigaciones han puesto su atención en reutilizar el conocimiento previo de cómo combinar modelos para mejorar la predicción. El proceso de aprender a partir de tareas y modelos que mejor funcionan es denominado meta-aprendizaje. La estrategia de meta-aprendizaje permite generar automáticamente un experto capaz de recomendar una combinación de modelos apropiada para predecir una serie de tiempo en particular. Sin embargo, el proceso de meta-aprendizaje en series de tiempo impone retos no triviales tales como: extraer características de los datos, buscar la mejor combinación de modelos para cada serie, e incluso generar nuevos métodos de meta-aprendizaje que consideren las variaciones de desempeño de los modelos en el tiempo. Esta investigación propone atacar el problema de meta-aprendizaje de combinación de modelos, utilizando auto-organización y estrategias basadas en ventanas de predicción. La meta es mejorar la predicción a largo plazo alcanzada por los algoritmos del estado del arte en series de tiempo caóticas. Con esta finalidad, proponemos utilizar auto-organización para generar las mejores combinaciones y dividir el horizonte de predicción en tres partes: corto, mediano y largo horizonte. Obteniendo automáticamente conocimiento acerca de cuál combinación de modelos tiene el mejor desempeño en cada horizonte, esperamos mejorar la eficacia de predicción. Sin embargo, al realizar esta combinación, el espacio de búsqueda incrementa en tamaño y dificultad, ya que los modelos pueden combinarse de muchas maneras no efectivas. Para evitar esto, proponemos el uso de auto-organización, fomentando la colaboración entre los modelos. En este documento, se presentan las ideas principales de esta propuesta, el plan de trabajo y algunos resultados preliminares obtenidos al combinar modelos en diferentes horizontes de predicción.

Palabras Clave: *Meta-aprendizaje, auto-organización, combinación de modelos, predicción a largo plazo de series de tiempo, series de tiempo caóticas.*

Índice general

Capítulo 1	Introducción.....	1
Capítulo 2	Marco teórico.....	5
2.1.	Introducción.....	5
2.2.	Predicción de series de tiempo a largo plazo.....	6
2.3.	Sistemas caóticos y modelos no-lineales.....	6
2.3.1.	Sistemas caóticos	6
2.3.2.	Modelos no-lineales	7
2.4.	Evaluación de la predicción	9
2.5.	Modelos de descomposición estructural.....	11
2.6.	Estrategias del ajuste de estacionalidad.....	12
2.7.	Modelos ARIMA.....	13
2.8.	Redes neuronales.....	13
2.8.1.	Redes neuronales NAR	14
2.9.	Máquinas de soporte vectorial.....	15
2.10.	Enfoques de ensambles	15
2.11.	Redes modulares	16
2.12.	Meta-aprendizaje.....	17
2.12.1.	Auto-organización y las redes SOM	17
2.13.	Resumen	19
Capítulo 3	Trabajo relacionado	21
3.1.	Predicción de series de tiempo a largo plazo.....	21

3.2. Ensamblados de clasificadores	22
3.2.1. Método <i>stacking</i>	22
3.2.2. <i>Stacking</i> con distribuciones de probabilidad	23
3.3. Auto-organización en la combinación de modelos.....	24
3.3.1. Algoritmo constructivo para entrenar ensamblados de redes neuronales cooperativas .	24
3.3.2. Auto-organización de modelos supervisados	26
3.4. Meta-aprendizaje en predicción	27
3.5. Resumen	28
Capítulo 4 Propuesta	29
4.1. Problemática	29
4.2. Preguntas de investigación e hipótesis	30
4.3. Objetivo general	31
4.4. Objetivos específicos.....	32
4.5. Contribuciones.....	32
Capítulo 5 Metodología.....	33
5.1. Plan de trabajo	36
Capítulo 6 Resultados preliminares.....	37
6.1. Diagrama del método general.....	37
6.2. Recopilación de datos y modelos	38
6.2.1. Pre-procesamiento.....	38
6.2.2. Características de las series de tiempo y de los modelos de predicción.....	39
6.2.3. Método de validación.....	41
6.3. Método extractor de meta-datos	42
6.4. Combinación temporal de modelos	43
6.5. Descripción de datos para la evaluación empírica.....	44
6.6. Caso de estudio y análisis de resultados.....	45
6.7. Avances sobre meta-aprendizaje auto-organizado	47

6.8. Comentarios finales.....	49
Capítulo 7 Conclusiones generales.....	50
Referencias.....	52
Anexos.....	57
Anexo I. Sistemas dinámicos.....	57
Anexo II. Características utilizadas en el análisis de series de tiempo	59

Capítulo 1

Introducción

Las series de tiempo caóticas están catalogadas como no predecibles a largo plazo, debido a su alta sensibilidad a las condiciones iniciales (Kantz & Schreiber, 2003). A pesar de ello, muchas aplicaciones que manejan sistemas caóticos requieren una estimación razonable de los valores futuros de las series de tiempo involucradas. Por esta razón, varias investigaciones buscan mejorar la eficacia de los modelos de predicción aplicados en diferentes ámbitos, por ejemplo modelos financieros (Pavlidis, et al., 2003), de previsión de consumo de carga eléctrica (Matijas, et al., 2013), de demanda de turismo (Andrawis, et al., 2011), de velocidad del viento en la superficie (Cao, et al., 2012) o de cardiología (Strauss, et al., 2011). Todos los ejemplos mencionados anteriormente trabajan analizando series de tiempo caóticas.

A pesar de los esfuerzos por mejorar la eficacia en la predicción, el problema de predecir múltiples pasos adelante partiendo de los datos del pasado es todavía un área de investigación abierta (De-Gooijer & Hyndman, 2006). Hay una gran cantidad de trabajos que han tratado este problema, principalmente utilizando modelos estadísticos y modelos basados en inteligencia computacional. Ejemplos de algoritmos estadísticos que proponen soluciones simples incluyen los basados en caminatas aleatorias y suavizado exponencial. Los sistemas complejos, comúnmente diseñados por expertos en estadística, están basados en modelos tipo ARIMA *Autoregressive Integrated Moving Average* (Box & Pierce, 1970). Ejemplos de modelos basados en inteligencia computacional incluyen el uso de algoritmos genéticos, lógica difusa, redes neuronales y máquinas de soporte vectorial (Sapankevych & Sankar, 2009), (Lemke & Gabrys, 2010), entre otros. Asimismo, los modelos de predicción han sido utilizados solos o en combinación con varias estrategias (Makridakis & Hibon, 2000).

Varios autores han buscado determinar los mejores algoritmos de predicción. Las principales conclusiones a las que han llegado son las siguientes:

- Los mejores resultados en dominios reales han sido obtenidos por combinaciones de modelos (Crone, et al., 2011), (Makridakis & Hibon, 2000).
- Las características de los datos determinan el desempeño entre modelos (Wang, et al., 2009) (Crone, et al., 2008), (Makridakis & Hibon, 2000).
- La precisión del modelo depende del número de pasos a predecir, valor conocido como “*horizonte de predicción*” (Crone, et al., 2008) (Makridakis & Hibon, 2000).
- Algunos de los mejores métodos de predicción siguen la estrategia “divide y vencerás” por ejemplo, el trabajo (Adeodato, et al., 2008). En este tipo de modelos, la serie de tiempo es descompuesta en varios *componentes estructurales*, tales como tendencia, estacionalidad e irregularidades. Después, un modelo es entrenado con cada componente. La predicción final es obtenida combinando las predicciones de todos los modelos.
- La colaboración automática entre modelos obtiene buenos resultados en el ámbito de clasificación (Kordik & Cerny, 2011), (Garcia-Pedrajas, et al., 2005). Sin embargo, a la fecha esta estrategia no ha sido probada en predicción a largo plazo.

El proceso de selección de un modelo adecuado de predicción, seguido por los expertos humanos, inicia revisando los datos. Luego los expertos, de acuerdo a su experiencia, deciden los modelos a utilizar y los ajustan. El alto costo en dinero y tiempo del análisis de expertos ha motivado la búsqueda de enfoques automáticos de esta estrategia. En los últimos años varios trabajos han sido publicados relacionados con este punto, por ejemplo (Lemke & Gabrys, 2010), (Wang, et al., 2009), (Matijas, et al., 2013). El trabajo que presenta mejores resultados a la fecha en este contexto es el de Lemke y Gabris (Lemke & Gabrys, 2010), donde, los autores superaron el desempeño de métodos individuales y el obtenido por diferentes combinaciones de todos los métodos involucrados en sus experimentos. Dicho trabajo parte de un conjunto de series de tiempo, donde se extraen características propias de las series llamadas “datos de los datos” o “*meta-datos*”. Posteriormente, se entrenan varios algoritmos de aprendizaje usando estos meta-datos, proceso conocido como “*meta-aprendizaje*” (Castiello & Fanelli, 2011). El proceso de meta-aprendizaje automáticamente induce un *meta-modelo*, al cual llamaremos “*experto*”. Dada una nueva tarea de predicción, este experto es capaz de determinar la mejor combinación de modelos capaces de predecir esta nueva serie.

La presente investigación se enfoca en el problema de decidir automáticamente qué modelos deben ser combinados y cómo deben combinarse, en el contexto de predicción a largo plazo de series de tiempo caóticas. Consideramos que los modelos existentes pueden mejorarse, diseñando expertos basados en ensambles que consideren auto-organización y la inclusión de un manejo de división del horizonte de predicción en diferentes ventanas. A esta división la referiremos en el

presente documento como “*combinación temporal*”. Por lo tanto, se propone en esta investigación el desarrollar un algoritmo de meta-aprendizaje capaz de construir combinaciones de modelos, de manera auto-organizada, considerando diferentes horizontes de predicción en series de tiempo caóticas. Las principales contribuciones esperadas de esta investigación son:

1. Una nueva estrategia para combinar modelos de predicción, considerando auto-organización de los modelos y diferentes horizontes de predicción. La combinación propuesta obtendrá mejor eficacia en la predicción a largo plazo que la reportada en el trabajo más cercano a éste, encontrado al momento de escribir este documento, reportado por (Lemke & Gabrys, 2010).
2. Un algoritmo auto-organizado de meta-aprendizaje aplicado en el contexto de predicción a largo plazo de series de tiempo caóticas.

Siguiendo la metodología descrita en el capítulo cinco, a la fecha se han logrado los siguientes avances:

1. Una recopilación y análisis de series de tiempo, que representan diferentes ámbitos de aplicación.
2. La determinación de funciones de evaluación de modelos de predicción, métodos de pre-procesamiento de datos, métodos de validación de modelos de predicción y modelos de predicción, apropiados al contexto de la investigación.
3. La implementación de un extractor de meta-datos, el cual permite buscar el mejor modelo para una serie de tiempo en particular en un espacio de búsqueda definido previamente.
4. La implementación de un algoritmo de combinación temporal de n -modelos.
5. La ejecución de un caso de estudio para probar la pertinencia de la combinación temporal, sin considerar aún auto-organización. Aquí se compararon los resultados obtenidos por la combinación de n -mejores modelos contra los obtenidos por combinación temporal, esto es, usando diferentes horizontes de predicción.

Los experimentos realizados a la fecha han mostrado que la combinación temporal obtiene, en promedio, mejor eficacia de predicción que la combinación de n -mejores modelos. Sin embargo, no se encontró una diferencia estadísticamente significativa entre estas dos estrategias. Dado que en algunos casos funcionó mejor la estrategia de combinación de n -mejores modelos. Un modelo que adapte automáticamente la estrategia de combinación de modelos de acuerdo a las características de las series obtendrá mejores resultados. La estrategia de combinación de modelos puede obtenerse de la auto-organización de los modelos a combinar. Actualmente se está trabajando con la inclusión de auto-organización y mejorando el método de combinación de modelos utilizado. Esto último se está llevando a cabo a través del uso de *stacking* con distribución de probabilidad, construyendo un

modelo inspirado en el propuesto por (Dzeroski & Zenko, 2004). Los detalles de estos avances se presentan en el capítulo seis.

Finalmente, es importante hacer notar que parte del trabajo reportado en este documento ha sido publicado y/o aceptado para publicación en los siguientes foros con arbitraje internacional:

- Fonseca R, Gómez-Gil P “*Temporal self-organized meta-learning for predicting chaotic time series,*” *Advances in Pattern Recognition, special issue, editors: Olvera-López JA et al., Research in Computing Science, Vol 61, ISSN: 1870-4069, 2013. Pp 13-22.* Este trabajo fue presentado en el “Mexican Congress on Pattern Recognition 2013, PhD student’s meeting.”
- Fonseca-Delgado R, Gomez-Gil, P. “*An assessment of ten-fold and Monte Carlo cross validations for time series forecasting.*” A publicarse en las memorias del “*10th International Conference on Electrical Engineering, Computing Science and Automatic Control CCE 2013*” Sept. 30 – Oct. 6, 2013.

En los siguientes capítulos se profundizan y apoyan los conceptos descritos anteriormente. En el capítulo 2 se describe detalladamente el marco teórico asociado a este proyecto; los principales trabajos relacionados a la investigación se encuentran en el capítulo 3. El capítulo 4 proporciona una descripción del problema a resolver, de las preguntas de investigación y de los objetivos de la tesis. El capítulo 5 explica la metodología a seguir, incluyendo el plan de trabajo; en el capítulo 6 se muestra la viabilidad de la propuesta a través del análisis de resultados empíricos obtenidos a la fecha. En el último capítulo se presentan las conclusiones generales de la propuesta.

Capítulo 2

Marco teórico

2.1. Introducción

La predicción de series de tiempo es un área muy activa desde los años 50's (De-Gooijer & Hyndman, 2006). Aunque buenos resultados han sido obtenidos en los últimos años, existe un grupo de series de tiempo que son difíciles de predecir por su alta sensibilidad a las condiciones iniciales; esto quiere decir que pequeños cambios en las condiciones de inicio implican un gran cambio en el comportamiento de toda la serie. Estas series son el resultado de sistemas caóticos. Ejemplos de este tipo de series pueden encontrarse en demanda de carga eléctrica (Matijas, et al., 2013), en retiros bancarios (Pavlidis, et al., 2003) y en la velocidad del viento en la superficie terrestre (Cao, et al., 2012), entre otros.

Persisten varios problemas abiertos en esta área. El primer problema a atacar, acordado por varios investigadores, es la necesidad de mejorar los métodos de combinación de predicciones que se aplican para conseguir predicciones más exactas (Hibon & Evgeniou, 2005). El segundo problema importante a atacar es el desarrollar procedimientos para seleccionar modelos que utilicen tanto los datos disponibles como el conocimiento previo sobre el medio ambiente (De-Gooijer & Hyndman, 2006).

Como se comentó anteriormente, esta investigación trata el problema de seleccionar los modelos a combinar y cómo combinarlos en el contexto de predicción de series de tiempo caóticas. La propuesta de solución descrita aquí se encuentra en la intersección de diferentes áreas del conocimiento: teoría de predicción de series de tiempo a largo plazo, redes neuronales artificiales, auto-organización, análisis de series de tiempo, meta-aprendizaje y teoría de ensamblados. En este capítulo se presentan los conceptos más importantes asociados a estas áreas y que son utilizados en el resto del documento.

2.2. Predicción de series de tiempo a largo plazo

En varias situaciones la información de cierto dominio es representada como una secuencia de medidas tomadas consecutivamente en el tiempo. Dichas secuencias se denominan series de tiempo. En la mayoría de casos prácticos se consideran series de tiempo discretas, donde el tiempo está indizado por valores enteros (Borovkova, 1998). La notación normalmente utilizada para las series de tiempo discretas es $\{y_i\}$, donde el índice $i \in \mathbb{N}$ representa el lugar del muestreo de la serie. Las series de tiempo utilizadas en esta investigación son de una dimensión, esto es $y_i \in \mathbb{R}$.

El objetivo de la predicción de series de tiempo es estimar algún valor futuro basado en los datos actuales y del pasado. En los trabajos relacionados a esta área se ha puesto mucha atención a la predicción de un solo valor en el futuro, por lo que la investigación de predecir múltiples valores futuros aún es limitada (Taieb, et al., 2012). La predicción de series de tiempo de múltiples pasos consiste en estimar los siguientes H valores $\{y_{N+1}, \dots, y_{N+H}\}$ de una serie de tiempo $\{y_1, \dots, y_N\}$ compuesta de N observaciones, donde $H > 1$ denota el horizonte de predicción. En la literatura revisada hasta el momento, no existe un acuerdo de cuándo H es considerada largo plazo; esto se debe principalmente a la dependencia de este criterio del dominio de aplicación. Varios trabajos tratan de estimar el mayor número de pasos a futuro posibles mientras el error acumulado de la predicción sea aceptable para cada dominio; ésta será la estrategia seguida en la presente investigación para considerar H un valor representativo del concepto “largo plazo”.

De manera general, existen dos estrategias para estimar los valores futuros que forman un horizonte de predicción. La primera estrategia estima los valores en un solo cálculo. La segunda estrategia predice un elemento a la vez y utiliza la última predicción para estimar el siguiente valor; éste proceso es repetido hasta alcanzar el horizonte de predicción deseado (Crone, et al., 2011). Está investigación se enfoca en las estrategias de predecir un paso a la vez iterativamente, ya que ésta es más flexible para la combinación de predicciones en diferentes horizontes de predicción.

2.3. Sistemas caóticos y modelos no-lineales

2.3.1. Sistemas caóticos

Kaplan y Glass sugieren que caos se define como “dinámicas acotadas y aperiódicas en un sistema determinista, con dependencia sensible a las condiciones iniciales” (Kaplan & Glass, 1995). Abarbanel y Gollub proponen que caos es “la evolución determinista de un sistema no-lineal, la cual está entre el comportamiento regular y el comportamiento estocástico o ruido” (Abarbanel &

Gollub, 1996). Este comportamiento de sistemas no-lineales es un poco predecible, no-periódico y presentan cambios exponenciales en respuesta a cambios en las condiciones iniciales.

Las características de un sistema caótico según (Dhanya & Kumar, 2010) están listadas a continuación:

1. Son deterministas, es decir, existe alguna ecuación determinista (probablemente desconocida) guiando su comportamiento.
2. Son sensibles a las condiciones iniciales (un pequeño cambio en el punto de inicio puede llevar a una salida considerablemente diferente).
3. No son aleatorios ni desordenados. Los sistemas caóticos tienen un sentido del orden y patrones, incluso cuando estos patrones no se repitan.

En esta investigación se entiende por serie de tiempo caótica a la secuencia registrada de la salida de un sistema caótico. Estas series de tiempo son difíciles de predecir por las siguientes razones:

- Una variación infinitamente pequeña en el estado presente puede implicar una diferencia finita en el estado del sistema en un tiempo finito, la condición del sistema se dice que es inestable y vuelve imposible la predicción exacta de los eventos futuros (Campbell, 1882).
- Dada la sensibilidad a las condiciones iniciales, en 1898 el matemático Jacques Hadamard resalta que una pequeña diferencia o error en las condiciones iniciales puede hacer la predicción a largo plazo de un sistema caótico imposible (Cartwright, 1965). Por ejemplo, Edward Lorenz desarrolló un modelo determinista para predecir el clima. Él encontró que pequeños cambios en las condiciones iniciales causaban una diferencia drástica del resultado final y el error crecía exponencialmente (Lorenz, 1963). Él concluyó que la sensibilidad a las condiciones iniciales de estos sistemas amplifica un pequeño error de manera exponencial causando consecuencias enormes.

2.3.2. Modelos no-lineales

Sea la serie de tiempo $\{y_n\}, n \in \mathbb{N}$ obtenida por la regla definida en la ecuación (2.1), donde $f: \mathbb{R} \rightarrow \mathbb{R}$ es una función no-lineal y $y_0 \in \mathbb{R}$ es un valor inicial.

$$y_{n+1} = f(y_n) \tag{2.1}$$

Incluso en este caso simple, determinista y de una dimensión, la función no-lineal f puede causar un comportamiento complejo de la serie de tiempo resultante tanto que parezca aleatoria.

Uno de los ejemplos más nombrados de este fenómeno es el del mapa logístico (May, 1976) $f: [0,1] \rightarrow [0,1]$ dado por

$$f(x) = \lambda x(1 - x) \tag{2.2}$$

Es conocido que esta función exhibe un comportamiento caótico para ciertos parámetros, por ejemplo para $\lambda = 4$. El comportamiento de (2.2) es caótico según (May, 1976) dado que existe un número infinito de puntos fijos con diferentes periodicidades, y un número infinito de ciclos de diferentes períodos. También hay un número incontable de puntos iniciales los cuales dan trayectorias totalmente aperiódicas, aunque acotadas; sin importar cuán larga sea la serie de tiempo generada, los patrones nunca se repiten. Por ejemplo, la serie de tiempo mostrada en la Figura 2.1 (a) fue generada por el mapa logístico de la ecuación (2.3).

$$y_{n+1} = 4y_n(1 - y_n) \tag{2.3}$$

La Figura 2.1 (b) muestra en el eje de las ordenadas el valor de y en el tiempo n mientras que en el eje de las abscisas se representa el valor de y en el tiempo $n + 1$. Esta gráfica, conocida como *mapa de retorno* (Takens, 1981), muestra si existe una relación entre un y_n y algún valor de la misma serie adelantado en el tiempo. Los mapas de retorno son una herramienta de análisis de sistemas dinámicos. Una breve explicación de otros conceptos involucrados en el mapa de retorno se incluye en el Anexo I. Con la finalidad de ejemplificar una serie de tiempo resultante de un generador de números aleatorios, en la Figura 2.2 (a) se muestra una serie aleatoria y en la Figura 2.2 (b) se muestra su respectivo mapa de retorno. Nótese que el mapa de retorno de la serie aleatoria no muestra ninguna forma particular.

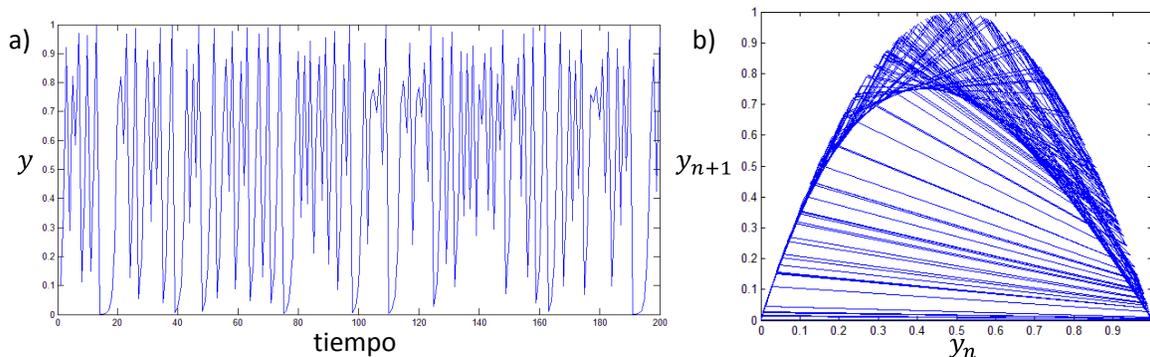


Figura 2.1. a) Serie de tiempo no-lineal conocida como ‘mapa logístico’, con un comportamiento que parece aleatorio. b) Mapa de retorno del mismo ‘mapa logístico’ que muestra una relación entre los tiempos n y $n + 1$.

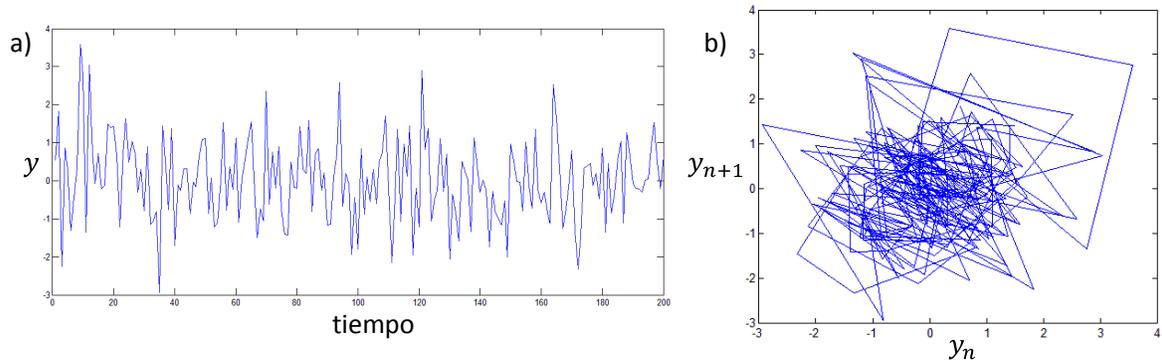


Figura 2.2. a) Serie de tiempo resultado de un generador de números aleatorios con distribución normal. b) Mapa de retorno de la serie de tiempo aleatoria, no se observa una relación entre los tiempos n y $n + 1$.

En general, un sistema caótico es no-lineal, sin embargo, no todo sistema no-lineal es caótico (Grigorenko, 2006). Un método numérico para determinar si una serie tiene un comportamiento caótico consiste en calcular el máximo exponente de Lyapunov (Kantz, 1994). El exponente de Lyapunov de un sistema dinámico es una cantidad que caracteriza el grado de separación de dos trayectorias infinitesimalmente cercanas. Un exponente positivo indica que la serie tiene un comportamiento caótico (Kantz & Schreiber, 2003).

2.4. Evaluación de la predicción

La eficacia es uno de los criterios más importantes de los utilizados en la evaluación de sistemas de predicción (Armstrong, 1985), por lo que será utilizado en esta investigación. Existen varios métodos de evaluación de la predicción realizada por un modelo, cada uno con diferentes inclinaciones. En este trabajo se utilizarán varios métodos de evaluación con la finalidad de explicar los resultados desde diferentes puntos de vista, que son descritos a continuación.

En general una función de evaluación de predicción recibe la serie estimada por el modelo y la serie esperada. La salida de la función es un valor real cuyos límites dependen del método de evaluación. La notación utilizada en los diferentes métodos es:

- A indica el valor esperado,
- F indica la estimación obtenida por el modelo,
- n indica el índice de tiempo,
- H es el tamaño del horizonte de predicción

Raíz del promedio del error al cuadrado.- RMSE (por las siglas de *Root Mean Square Error*), es una métrica de eficacia invariante al sesgo del error (Armstrong, 1985). Usando esta función, el

costo asociado con un error se incrementa al cuadrado del error. RMSE es sensible a valores inesperados (grandes mediciones de error), y es calculado como:

$$RMSE = \sqrt{\frac{\sum_{n=1}^H (A_n - F_n)^2}{H}} \quad (2.4)$$

Un valor de 0 en RMSE implica una predicción correcta; entre mayor sea el valor del RMSE mayor es el error de la predicción.

Promedio de error absoluto.- MAE (por las siglas de *Mean Absolute Error*), esta función mide el promedio de magnitudes de los errores en un conjunto de predicciones sin considerar sus direcciones. Esta función no distingue entre la varianza y el sesgo. MAE es apropiado cuando el costo de los errores de predicción sigue una función lineal. MAE se define como (Armstrong, 1985):

$$MAE = \frac{\sum_{n=1}^H |A_n - F_n|}{H} \quad (2.5)$$

Promedio del porcentaje de error absoluto.- MAPE (por las siglas de *Mean Absolute Percentage Error*) no tiene una unidad; esto la vuelve útil para comparación de resultados y evaluación de modelos. Un valor de 0 en MAPE implica una predicción correcta, mientras que, valores que tienden a 100 están más alejados de la predicción correcta; MAPE tiende a favorecer bajas estimaciones (Armstrong, 1985) y se calcula como:

$$MAPE = \frac{\sum_{n=1}^H \frac{|A_n - F_n|}{A_n}}{H} \cdot 100 \quad (2.6)$$

Promedio ajustado del porcentaje de error absoluto.- Es representada por \overline{MAPE} , y es similar a MAPE, pero no favorece bajas estimaciones. Sus posibles valores van de 0 para una predicción perfecta a 200 para una predicción infinitamente mala. Esta métrica se calcula como:

$$\overline{MAPE} = \frac{\sum_{n=1}^H \frac{|A_n - F_n|}{(A_n + F_n)/2}}{H} \cdot 100 \quad (2.7)$$

Con la finalidad de evitar problemas con predicciones negativas, \overline{MAPE} puede adaptarse, definiéndose SMAPE (promedio simétrico del porcentaje del error absoluto), el cual se calcula como:

$$SMAPE = \frac{\sum_{n=1}^H \frac{|A_n - F_n|}{(|A_n| + |F_n|)/2}}{H} \cdot 100 \quad (2.8)$$

La función SMAPE fue utilizada para evaluar modelos en varios torneos de predicción internacionales como por ejemplo el NN3 y el NN5 (Crone, 2008).

2.5. Modelos de descomposición estructural

La descomposición estructural consiste en extraer las características más importantes de una serie de tiempo a partir de los datos (ABS, 2005). Un enfoque simple de descomposición es asumir que cualquier serie de tiempo puede ser descompuesta en tres componentes básicos (ABS, 2005):

1. La tendencia, o el nivel fundamental de la serie original. El componente tendencia se define como el movimiento a largo plazo en una serie.
2. La estacionalidad, o la porción de variación con respecto a la tendencia atribuible a factores que se repiten sistemáticamente una o más veces.
3. La irregularidad, o variación residual, es el remanente después de que los componentes de tendencia y estacionalidad son removidos de la serie original.

El modelo de descomposición de series de tiempo no es único, un problema importante es definir y estimar los componentes que se ajusten a un marco de trabajo teórico. En la selección de un modelo de descomposición, se desea escoger el modelo que genere el componente estacional más estable, especialmente al final de las series. Los métodos de descomposición estructural más utilizados son el multiplicativo, el aditivo y el mixto (combinación de multiplicativos y aditivos) (ABS, 2005).

La Tabla 2.1 ilustra los métodos de descomposición más conocidos. La segunda columna de la tabla muestra la descomposición asumiendo que solo existen los componentes tendencia, estacionalidad e irregular. La tercera columna asume un componente adicional, cómo podría ser por ejemplo, el día de la semana en series capturadas diariamente. En la tabla, SA_t representa la estacionalidad ajustada y SI_t es el componente de irregularidad ajustado.

Existen otros modelos para el análisis de una señal, en particular aquellos basados en el dominio de la frecuencia, tales como son las transformadas Wavelet y las transformadas de Fourier.

Descripciones de ejemplos de estos modelos, aplicados a la predicción de series de tiempo, pueden encontrarse en (García Pedrero & Gómez Gil, 2009) y (Gómez-Gil, 1998) respectivamente.

Los modelos de descomposición estructural generalmente siguen una estrategia de ajuste de estacionalidad. Hay diferentes estrategias, las cuales se describen a continuación.

Tabla 2.1. Comparación de los modelos de descomposición multiplicativo, aditivo y mixto.

	Modelo de tres componentes	Modelo de cuatro componentes
Componentes	$O_t =$ original	$O_t =$ original
	$T_t =$ tendencia	$T_t =$ tendencia
	$S_t =$ estacionalidad	$S_t =$ estacionalidad
	$I_t =$ irregularidad	$I_t =$ irregularidad
	$SA_t =$ estacionalidad ajustada	$D_t =$ día de comercio
	$SI_t =$ irregularidad ajustada	
Multiplicativo	$O_t = T_t \times S_t \times I_t$	$O_t = T_t \times S_t \times D_t \times I_t$
	$SA_t = O_t/S_t = T_t \times I_t$	$SA_t = O_t/(S_t \times D_t) = T_t \times I_t$
	$SI_t = O_t/T_t = S_t \times I_t$	$SI_t = O_t/(T_t \times D_t) = S_t \times I_t$
Aditivo	$O_t = T_t + S_t + I_t$	$O_t = T_t + S_t + D_t + I_t$
	$SA_t = O_t - S_t$	$SA_t = O_t - S_t - D_t$
	$SI_t = S_t + I_t$	$SI_t = S_t + I_t$
Mixtos	$O_t = T_t + (S_t + I_t - 1)$	$O_t = T_t \times D_t \times (S_t + I_t - 1)$
	$SA_t = O_t - T_t \times (S_t - 1)$	$SA_t = (O_t/D_t) - T_t \times (S_t - 1)$
	$SI_t = S_t + I_t - 1$	$SI_t = S_t + I_t - 1$

2.6. Estrategias del ajuste de estacionalidad

Según Sutcliffe (1999) las dos estrategias principales para el ajuste de estacionalidad son el método basado en el modelo y el método basado en filtros. La estrategia basada en el modelo asume que el modelo generador del sistema puede separarse en sub-modelos, cada sub-modelo genera los componentes como pueden ser tendencia, estacionalidad e irregularidad. Los métodos basados en el modelo asumen que el componente de irregularidad es “ruido blanco” es decir la irregularidad no tiene estructura, tiene un promedio de cero y una varianza constante (Wu & Huang, 2004).

Los métodos basados en filtros aplican un conjunto de filtros definidos para descomponer la serie de tiempo en componentes de tendencia, estacionalidad e irregularidad. Generalmente, estos

métodos aplican filtros lineales simétricos en la parte media de la serie, y filtros lineales asimétricos son aplicados en los extremos de la serie. Dada una posición actual de la serie de tiempo, un filtro simétrico utiliza el mismo número de elementos de posiciones anteriores como posteriores al punto actual para procesar el valor de la posición actual. Un filtro asimétrico utiliza un número distinto de elementos en posiciones anteriores y posteriores. Una familia de filtros que ha presentado muy buenos resultados es el promedio de movimientos Henderson (Trewin, 2003).

2.7. Modelos ARIMA

Los modelos Auto-Regresivos Integrados de Medias Móviles (ARIMA por sus siglas en inglés) proveen una gran herramienta para describir series de tiempo estacionarias y no estacionarias. Box y colaboradores (Box, et al., 1994) definen un modelo de orden (p, q) como:

$$w_t = \phi_1 w_{t-1} + \dots + \phi_p w_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.9)$$

Donde ϕ_i son constantes que determinan la dependencia del valor actual de los p valores previos, mientras que θ_j son constantes que relacionan los q errores aleatorios a_t del pasado con la predicción. El proceso ARIMA puede ser generado sumando o “integrando” el proceso estacionario auto-regresivo de medias móviles w_t (ARMA por sus siglas en inglés) d veces. En esta investigación los modelos ARIMA son utilizados por sus buenos resultados en el área de predicción. Una descripción más detallada de los modelos ARIMA puede verse en (Box, et al., 1994).

Un grupo de modelos ampliamente utilizados en el área de predicción de series de tiempo por sus buenos resultados son las redes neuronales. La siguiente sección da una introducción breve de este tipo de modelos.

2.8. Redes neuronales

Las redes neuronales están inspiradas en el funcionamiento del cerebro humano, principalmente en la plasticidad o capacidad de adaptarse. Una red neuronal es una máquina inspirada en la manera en la cual el cerebro realiza una tarea particular o función de interés. Las redes neuronales usualmente están implementadas utilizando componentes electrónicos o son simuladas por software. Para alcanzar un buen desempeño, las redes neuronales emplean una gran cantidad de interconexiones de celdas de cómputo simples conocidas como “neuronas” o “unidades de procesamiento”. Descripciones de los tipos de redes neuronales más populares pueden encontrarse en (Haykin, 1999). Gracias a su capacidad de adaptación las redes neuronales han sido ampliamente

utilizadas en el problema de predicción de series de tiempo (Crone, et al., 2011). Enseguida se describen los modelos de redes neuronales utilizados en esta investigación.

2.8.1. Redes neuronales NAR

Las redes neuronales auto-regresivas no-lineales (NAR por sus siglas en inglés), son un tipo de red neuronal con una o varias capas ocultas (Leontaritis & Billings, 1985). La característica principal es una conexión recurrente entre la capa de salida y la capa de entrada. Los componentes básicos de una modelo NAR son mostrados en la Figura 2.3. Segmentos de la serie de entrada son almacenados en el bloque de retraso de tiempo (TDL por sus siglas en inglés). El bloque TDL es utilizado como entrada para un procesamiento alimentado hacia adelante. El elemento devuelto por la capa de salida es formado en una cola de espera en un segundo TDL, usado para evaluar la siguiente predicción. Los parámetros que generalmente se varían en las redes NAR son el número de elementos de retraso, el número de neuronas en la capa oculta y el algoritmo de entrenamiento. Más información de los modelos NAR puede encontrarse en (Zemouri, et al., 2010).

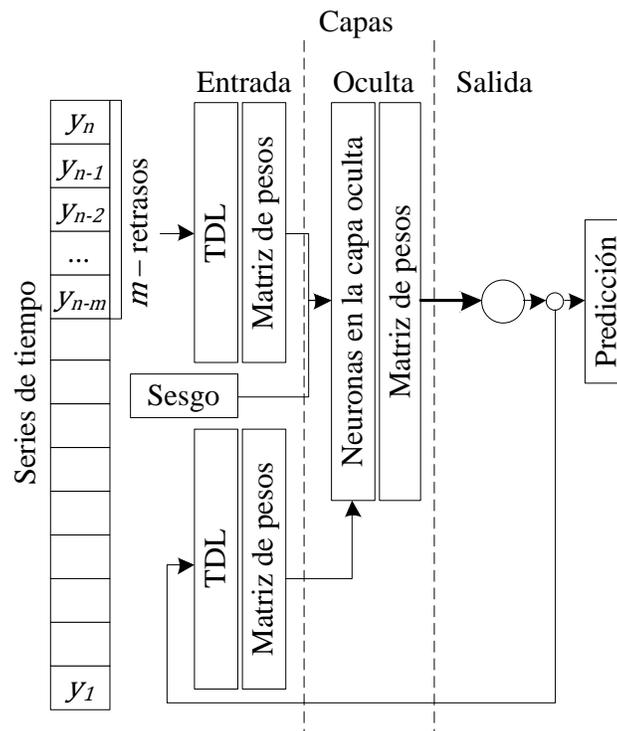


Figura 2.3. Estructura de una red neuronal NAR con un TDL de m neuronas de retraso, una capa oculta con varias neuronas, y una neurona en la capa de salida, imagen basada en (Beale, et al., 2012).

Otros modelos neuronales recurrentes, aplicados en el problema de predicción, que han presentado muy buenos resultados son los basados en descomposiciones en el dominio de la

frecuencia. Ejemplos de estos modelos incluyen a la “Red Neuronal Híbrida compleja” (HCNN por sus siglas en inglés) (Gómez-Gil, 1998), al modelo “Híbrido basado en reconstrucciones wavelet” (HWNN por sus siglas en inglés) (García Pedrero & Gómez Gil, 2009) y a las redes neuronales *Echo State* (Koryakin, et al., 2012). Estos modelos se consideran como modelos base para la construcción del método propuesto en esta investigación.

2.9. Máquinas de soporte vectorial

Las máquinas de soporte vectorial mapean de manera no-lineal vectores de entrada a un espacio de características de alta dimensión. En este espacio de características una superficie de decisión lineal es construida. La propiedad de esta superficie de decisión es asegurar una alta habilidad de generalización (Cortes & Vapnik, 1995). Las máquinas de soporte vectorial han sido muy exitosas en la predicción de series de tiempo (Sapankevych & Sankar, 2009).

Una extensión de las máquinas de soporte vectorial son las máquinas de regresión de soporte vectorial (SVR por sus siglas en inglés) (Drucker, et al., 1997). Las SVRs buscan la función que mejor se ajuste a la serie de entrenamiento. Más información del funcionamiento de las SVRs puede encontrarse en (Drucker, et al., 1997). En esta investigación vamos a extender el conjunto de modelos base utilizando SVRs.

2.10. Enfoques de ensambles

Un ensamble combina predicciones de la misma serie de tiempo provenientes de diferentes modelos. La finalidad de un ensamble es alcanzar un desempeño mejor que el alcanzado por el mejor de sus modelos componentes (Bates & Granger, 1969) (Dzeroski & Zenko, 2004).

En el error de un modelo de predicción existe un compromiso entre el sesgo y la varianza. El sesgo de modelo de predicción indica cuanto se aleja la predicción del valor esperado, mientras la varianza muestra cuantas veces se aleja la predicción del valor esperado. Generalmente, cuando se pretende disminuir el sesgo aumenta la varianza y viceversa (Sharkey, 1999). La mejora en el desempeño alcanzada por un ensamble, usualmente, es el resultado de una reducción de la varianza, en lugar de una reducción del sesgo, dado que el efecto usual de combinar por promedio es reducir la varianza de un conjunto de modelos de predicción. Por esta razón se busca combinar modelos cuyos errores tengan baja correlación (Sharkey, 1999). Esta investigación utiliza la teoría de ensambles para mejorar la combinación de modelos de predicción.

2.11. Redes modulares

En la literatura existe una gran cantidad de arquitecturas neuronales que funcionan bien cuando el número de entradas es relativamente pequeño, pero cuando la complejidad del problema crece o el número de entradas aumenta, el desempeño decrece muy rápido. Por esta razón, ha habido trabajo de investigación compensando de alguna manera los problemas en aprendizaje de una única red neuronal sobre un espacio de alta dimensión (Melin, et al., 2007).

Sharkey (1999) describe el uso de múltiples sistemas neuronales (redes-múltiples). Se afirma que las redes múltiples tienen mejor desempeño o incluso resuelven problemas que una sola red neuronal (red monolítica) no es capaz de resolver. También se afirma que las redes múltiples o sistemas modulares tienen además la ventaja de ser fáciles de entender o modificar, si es necesario.

Las redes modulares se basan en la idea de que una tarea puede ser descompuesta en sub-tareas. Diferentes redes son entrenadas para realizar cada sub-tarea, esto es un módulo. Finalmente, la solución completa requiere la contribución de todos los módulos (Sharkey, 1999). Las redes modulares se basan en el principio de divide y vencerás (Brassard & Pa, 1996). En este enfoque se pueden encontrar redes que utilizan arquitecturas fuertemente separadas. Cada red neuronal trabaja independientemente en su propio dominio y cada red neuronal es construida y entrenada para una sub-tarea específica. La decisión final se basa en los resultados de las redes individuales, llamadas agentes o expertos. (Melin, et al., 2007)

Una lista de las ventajas de las redes modulares, tomada de (Melin, et al., 2007), es presentada a continuación:

- Mejoran significativamente las capacidades de aprendizaje sobre las redes neuronales monolíticas, debido a las restricciones impuestas por la arquitectura modular.
- Permiten modelar comportamientos complejos, utilizando diferentes tipos de conocimiento.
- La modularidad puede implicar una reducción del número de parámetros, lo cual permite mejorar la capacidad de generalización, adicionalmente, se reduce el tiempo de procesamiento requerido.
- Ayudan a determinar la actividad que está siendo realizada en cada parte del sistema, ayudando a entender el papel que juega cada red dentro del sistema completo.
- Si existen cambios en el ambiente, las redes modulares permiten realizar cambios fácilmente, dado que no es necesario modificar el sistema completo, sino únicamente los módulos afectados por el cambio.

Los elementos que componen una red modular son (Ronco & Gawthrop, 1995):

- La descomposición del problema principal en sub-tareas.

- La organización de la arquitectura modular, tomando en cuenta la naturaleza de cada sub-tarea.
- La comunicación entre los módulos, es muy importante considerar la entrada del sistema y la integración de la respuesta.

En esta investigación consideramos que la combinación de modelos puede seguir el enfoque de las redes modulares, el cual divide la tarea en sub-tareas con la finalidad de dividir la complejidad del problema.

2.12. Meta-aprendizaje

Castiello y Fenalli (Castiello & Fanelli, 2011) dan la siguiente definición de meta-aprendizaje.

Sea A un conjunto de algoritmos de aprendizaje y T un conjunto de tareas. Sea $a_A(t)$ el mejor algoritmo en A aplicable a la tarea específica t , para cada tarea $t \in T$, y $c(t)$ una caracterización de la tarea escogida t . Entonces un proceso de meta-aprendizaje es un mecanismo automático que, a partir del conjunto de meta-entrenamiento:

$$\{\langle c(t), a_A(t) \rangle : t \in T\} \quad (2.10)$$

induce un meta-modelo. Dada una nueva tarea, el meta-modelo estima el mejor modelo en A para la nueva tarea. El modelo devuelto por el meta-modelo es seleccionado para ser ajustado con la nueva tarea.

Una alternativa para seleccionar un modelo es realizar una búsqueda evaluando los modelos disponibles en A en la tarea específica t (Kohavi & John, 1997). Las dos estrategias principales para seleccionar modelos pueden trabajar juntas. En el primer paso del proceso de meta-aprendizaje es necesario buscar los mejores modelos en cada tarea para construir el conjunto de meta-entrenamiento. Por otro lado, la salida del proceso de meta-aprendizaje puede modificarse ligeramente para devolver un conjunto de modelos posibles como mejores y entre ellos buscar el modelo que mejor resultado devuelva en la nueva tarea. En esta investigación guiaremos la selección y combinación de modelos con meta-datos.

2.12.1. Auto-organización y las redes SOM

Desai y Kapral definen que las estructuras auto-organizadas surgen de un estado inicialmente sin estructura sin la intervención de un agente que predetermine el comportamiento del sistema. Las estructuras auto-organizadas emergen de interacciones cooperativas entre los componentes que constituyen el sistema y a menudo exhiben propiedades distintas a las de los componentes individuales (Desai & Kapral, 2009), (Haken, 2006).

Un sistema complejo está compuesto por varias partes posiblemente del mismo tipo o de diferente tipo. El sistema complejo en su conjunto puede presentar un comportamiento caótico (Helbing, 2012). El comportamiento colectivo de todo el sistema no puede ser simplemente inferido del comportamiento individual de los componentes (Haken, 2006), es necesario poner atención en las interacciones de los componentes del sistema (Bertelle, et al., 2009). La coordinación y cooperación en un sistema complejo aparecerán por si solas, si las interacciones entre los elementos del sistema son apropiadas (Helbing, 2012). Las interacciones permitidas entre los componentes de un sistema pueden estar definidas; por ejemplo, en la naturaleza, las leyes de fluctuación y conservación juegan un rol importante guiando la evolución del sistema en el tiempo. Estas leyes o reglas de interacción dan lugar a estructuras auto-organizadas (Desai & Kapral, 2009). En esta investigación definirá la interacción de los modelos de predicción con la finalidad de mejorar el comportamiento de los modelos en su conjunto. Modelos computacionales auto-organizados guiados por reglas simples son las redes neuronales SOM (Kohonen, 1990).

Los *mapas auto-organizados* (SOM, *Self-Organizing Maps*) (Kohonen, 1990) son un tipo de red neuronal compuesta por una capa de entrada y una o varias capas de salida. La principal ventaja de la red SOM es su capacidad de transformar un vector de entrada de una dimensión arbitraria en un mapa discreto de una o dos dimensiones. El mapa de salida está compuesto por vectores prototipos y cada vector de entrada se conecta con todos los prototipos de salida. La Figura 2.4 muestra en un ejemplo las partes de una red neuronal SOM (Kohonen, 1990).

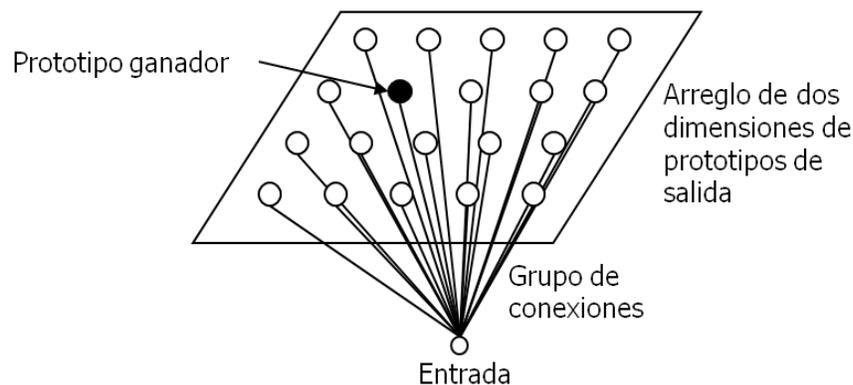


Figura 2.4. Ejemplo de un mapa auto-organizado (Gómez Gil, 2010).

El algoritmo de entrenamiento de las redes SOM es no supervisado. El proceso comienza inicializando los pesos sinápticos en la red, lo cual se puede realizar asignando valores aleatorios pequeños (entre 0 y 1) a los prototipos. A continuación, se siguen tres procesos: competitivo, cooperativo y adaptación sináptica, que se explican brevemente a continuación.

Dado un número definido de iteraciones sobre el conjunto de entrenamiento, el algoritmo adapta los pesos que conectan con la neurona ganadora. Esto lo hace en un proceso de *competencia* entre

todas las neuronas del nivel de salida. Entonces, la neurona ganadora se define como el prototipo de menor distancia al ejemplo y se obtiene calculando la distancia que hay entre el vector de entrada x y cada uno de los pesos de cada neurona en el nivel de salida w_j como lo muestra la ecuación:

$$i(x) = \underset{j}{\operatorname{argmin}} \|x - w_j\|, j = 1, 2, \dots, l \quad (2.11)$$

donde l es el número de neuronas en el nivel de salida.

En el proceso de *cooperación* se adaptan los pesos del vecindario de influencia de la neurona ganadora, la función de vecindario $h(i(x), j)$ devuelve la influencia que tiene la neurona ganadora i con la neurona vecina j . El coeficiente de aprendizaje $\eta(n)$ disminuye conforme transcurren las n iteraciones, la función recomendada por (Haykin, 2005) para actualizar el coeficiente de aprendizaje es la siguiente:

$$\eta(n + 1) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), \quad n = 0, 1, 2, \dots \quad (2.12)$$

donde τ_2 es una constante de tiempo y η_0 el coeficiente inicial. La *adaptación* de los pesos de la red neuronal SOM está dada por la ecuación:

$$w_j(n + 1) = w_j(n) + \eta(n)h(i(x), j) (x - w_j(n)) \quad (2.13)$$

En esta investigación proponemos guiar la selección y combinación de modelos utilizando las reglas simples de auto-organización de las redes neuronales SOM.

2.13. Resumen

La predicción iterativa de series de tiempo inicia con los valores conocidos y estima un valor a la vez. Este último valor es utilizado para predecir el siguiente valor. El proceso itera hasta alcanzar el horizonte de predicción. Un horizonte es considerado largo plazo dependiendo del dominio de la serie de tiempo. Las series de tiempo caóticas provienen de sistemas complejos sensibles a cambios en las condiciones iniciales. Es decir, las series de tiempo caóticas son difíciles de predecir porque un pequeño cambio en las condiciones iniciales implica un cambio exponencial en el comportamiento futuro del sistema.

Existen varias funciones para evaluar modelos de predicción. Este capítulo presentó las funciones RMSE, MAE y SMAPE.

Una herramienta poderosa en la predicción de series de tiempo es combinar modelos, una estrategia altamente utilizada es el uso de ensamblados. Un ensamblado combina varios modelos para realizar una tarea completa. Una alternativa para combinar modelos es utilizar redes modulares. La ventaja de las redes modulares es seguir la estrategia divide y vencerás, así una tarea compleja es dividida en varios componentes, se resuelven los componentes y se integra la solución de la tarea. En el caso de predicción de series de tiempo, la serie de tiempo puede ser descompuesta en componentes estructurales cada uno con sus características propias. Entonces, la tarea de predecir toda la serie se divide en predecir los diferentes componentes estructurales. Diferentes modelos pueden encargarse de cada componente, sin embargo el espacio de búsqueda de los modelos apropiados para predecir cada componente puede ser muy grande. El proceso de meta-aprendizaje puede ser útil para seleccionar un subconjunto de modelos a combinar para tareas particulares, el cual es de menor tamaño que todo el espacio de búsqueda original.

Auto-organización es alcanzar estructuras organizadas guiadas por las interacciones entre los componentes del sistema. Un ejemplo exitoso de auto-organización son los mapas auto-organizados (SOM), los cuales siguen reglas simples de competir y cooperar. Esta investigación busca mejorar la predicción a largo plazo de series de tiempo caóticas. En esta investigación proponemos utilizar las reglas simples de las redes SOM para seleccionar y combinar modelos de predicción. En el siguiente capítulo se presentarán los enfoques de investigaciones actuales con relación directa a esta investigación.

Capítulo 3

Trabajo relacionado

La presente investigación busca mejorar la eficacia de predicción a largo plazo en series de tiempo caóticas. El camino seleccionado está ubicado en la intersección de varias áreas como son: redes neuronales artificiales, teoría de ensambles, teoría del caos, teoría de auto-organización y análisis de series de tiempo. En esta sección son presentados los trabajos más relacionados y su implicación con la presente investigación.

3.1. Predicción de series de tiempo a largo plazo

La necesidad de mejorar las predicciones a largo plazo ha impulsado una extensa investigación por varios años (De-Gooijer & Hyndman, 2006). Varios autores han llegado a conclusiones importantes que abren nuevas oportunidades de investigación. Las conclusiones que guían esta investigación son listadas a continuación:

1. Las combinaciones de modelos obtienen los mejores resultados (Crone, et al., 2008), (Makridakis & Hibon, 2000), (Deutsch, et al., 1994).
2. Las características de los datos determinan el desempeño entre modelos (Matijas, et al., 2013), (Lemke & Gabrys, 2010), (Wang, et al., 2009).
3. La eficacia del modelo depende del horizonte de predicción (Wong, et al., 2010), (Crone, et al., 2008), (Makridakis & Hibon, 2000).

Las conclusiones motivan las siguientes ideas:

1. La combinación de diversos métodos puede mejorarse con teoría de ensambles.
2. La combinación de métodos podría realizarse considerando diferentes horizontes de predicción.
3. La mejor combinación puede elegirse a través de meta-aprendizaje a partir de las características de las diferentes series de tiempo.

Guiada por las conclusiones de (Crone, et al., 2011), (Makridakis & Hibon, 2000), (De-Gooijer & Hyndman, 2006), la revisión de los trabajos relacionados continúa con una descripción breve de los trabajos de ensambles de clasificadores. El objetivo es tomar ideas de estos trabajos como base para generar nuevas ideas y llevarlas al problema de predicción a largo plazo de series de tiempo caóticas.

3.2. Ensamblés de clasificadores

La combinación de clasificadores busca mejorar la eficacia en la decisión de la clasificación con el costo de incrementar la complejidad (Kuncheva, 2004). Los ensambles pueden agruparse por la manera de construcción del ensamble. Según (Kuncheva, 2004) los cuatro grupos principales son:

- Nivel de datos, utilizando diferentes subconjuntos de datos.
- Nivel de características, utilizando diferentes subconjuntos de características.
- Nivel de clasificador, utilizando clasificadores de diferente tipo¹.
- Nivel de combinación, diseñan diferentes métodos de combinación.

La mayoría de ensambles de clasificadores son generados utilizando un solo algoritmo de aprendizaje (Dietterich, 2000). Los clasificadores que forman parte del ensamble son generados por ejemplo manipulando las características de entrada, manipulando las salidas deseadas o insertando aleatoriedad en el aprendizaje. Los clasificadores generados son entonces combinados típicamente por voto mayoritario o pesado (Dzeroski & Zenko, 2004).

Otro enfoque de construcción de ensambles consiste en generar clasificadores aplicando diferentes algoritmos de aprendizaje (con esquemas diferentes de representación) utilizando un mismo conjunto de datos (ver (Merz, 1999)). El método *Stacking* (Wolpert, 1992) es utilizado con frecuencia para aprender un método de combinación además del ensamble de clasificadores. Este método se explica a continuación.

3.2.1. Método *stacking*

Stacking tiene como objetivo minimizar la tasa de error de generalización de uno o más clasificadores (Wolpert, 1992). Los N clasificadores son generados con diferentes algoritmos de aprendizaje L_1, \dots, L_N , en un único conjunto de datos S , el cual consiste de n ejemplos $s_i = (x_i, y_i)$, que son pares de vectores de características x_i y sus clasificaciones y_i . La primera fase consiste en generar un conjunto de clasificadores base C_1, C_2, \dots, C_N , donde $C_i = L_i(S)$. En la segunda fase, un

¹ Un clasificador de diferente tipo también se conoce como clasificador de diferente base.

clasificador de meta nivel es entrenado para combinar las salidas de los clasificadores base (Dzeroski & Zenko, 2004).

El clasificador de meta nivel requiere meta datos. La generación del conjunto de meta datos está basada en el mecanismo de validación cruzada dejando un elemento fuera (LOO-CV por sus siglas en inglés). LOO-CV aplica cada uno de los algoritmos de entrenamiento de nivel base a casi todo el conjunto de datos, dejando un ejemplo para prueba $C_k^i = L_k(S - s_i) | \forall i = 1, \dots, n | \forall k = 1, \dots, N$. Luego el modelo entrenado genera la estimación del ejemplo s_i , el resultado es $\hat{y}_i^k = C_k^i(x_i)$. El símbolo \hat{y}_i^k representa la salida del clasificador k para el ejemplo i . El conjunto de datos de meta nivel consiste de ejemplos de la forma $((\hat{y}_i^1, \dots, \hat{y}_i^n), y_i)$, donde las características corresponden a las estimaciones de los clasificadores de nivel base y la clase y_i es la clase correcta del ejemplo procesado (Dzeroski & Zenko, 2004).

Por su forma de operar, el método de *stacking* puede considerarse parte del área de meta-aprendizaje. En contraste, ningún aprendizaje tiene lugar a meta nivel cuando las estimaciones de los modelos son combinadas por un esquema de votación (Dzeroski & Zenko, 2004).

3.2.2. *Stacking* con distribuciones de probabilidad

En este método propuesto en (Ting & Witten, 1999) cada clasificador nivel base estima una distribución de probabilidad (PD por sus siglas en inglés) de los posibles valores de la clase. La respuesta del clasificador de nivel base C aplicado al ejemplo x es una PD:

$$p^C(x) = (p^C(c_1|x), p^C(c_2|x), \dots, p^C(c_m|x)) \quad (3.1)$$

donde $\{c_1, c_2, \dots, c_m\}$ es el conjunto de los posibles valores de clase y $p^C(c_i|x)$ denota la probabilidad de que el ejemplo x pertenezca a la clase c_i como estimado del clasificador C . La clase c_j con la más alta probabilidad $p^C(c_j|x)$ es devuelta por el clasificador C . Los atributos de meta nivel (Ting & Witten, 1999) son las probabilidades devueltas para cada una de las m clases posibles por cada uno de los clasificadores de nivel base $p^{C_j}(c_i|x)$ para $i = 1, \dots, m$ y $j = 1, \dots, N$.

En el trabajo de Dzeroski y Zengo (2004), los autores aumentaron dos atributos a meta nivel. El primer atributo fue la PD multiplicada por la máxima probabilidad:

$$P_{C_j} = p^{C_j}(c_i|x) \times M_{C_j(x)} = p^{C_j}(c_i|x) \times \max_{i=1, \dots, m} (p^{C_j}(c_i|x)) \quad (3.2)$$

El segundo atributo aumentado fue la entropía de la PD:

$$E_{C_j}(x) = - \sum_{i=1}^m p^{C_j}(c_i|x) \cdot \log_2 p^{C_j}(c_i|x) \quad (3.3)$$

como una medida de falta de certeza, es decir a mayor entropía menor certeza de la estimación. De igual manera la máxima probabilidad M_{C_j} en la PD estimada también contiene información de la certeza de la estimación, un valor de M_{C_j} alto significa que se tiene una estimación con alta certeza, y un valor bajo de M_{C_j} significa que se tiene una estimación incierta.

Los resultados de la combinación de modelos propuesta por (Dzeroski & Zenko, 2004) superaron a los resultados obtenidos por la selección del mejor modelo dentro de los modelos base en la tarea de clasificación. El método de Dzeroski y Zenko requiere modelos que devuelvan un indicador de certeza asociado con el resultado. En el área de predicción la certeza de la predicción disminuye con el horizonte de predicción. Esta investigación propone extender el método de *stacking* con distribución de probabilidades para considerar predicción de múltiples pasos a futuro y diferentes horizontes de predicción. El elemento necesario es un modelo de predicción que devuelva posibles valores futuros con probabilidades de confianza asociadas con cada valor futuro.

3.3. Auto-organización en la combinación de modelos

En la naturaleza algunos sistemas complejos presentan comportamiento caótico; este comportamiento es el resultado de las interacciones entre los componentes del sistema (Haken, 2006). La auto-organización es el proceso que siguen los componentes de un sistema para alcanzar el comportamiento del sistema completo (Rocha, 2013). A continuación se presenta dos trabajos que utilizan el concepto de auto-organización para combinar modelos y los resultados que obtienen. El primer trabajo busca mejorar la cooperación entre los modelos del ensamble. El segundo trabajo define reglas para fomentar la auto-organización en diferentes partes del proceso completo.

3.3.1. Algoritmo constructivo para entrenar ensambles de redes neuronales cooperativas

El algoritmo propuesto en (Islam, et al., 2003) diseña arquitecturas de redes neuronales (NNs). Los componentes de la arquitectura neuronal son miembros de un ensamble, los cuales son construidos utilizando entrenamiento cooperativo. Este algoritmo fue evaluado en varios problemas de clasificación y predicción de series de tiempo. Entre las series utilizadas está la serie Mackey-Glass conocida por tener comportamiento caótico.

Varios autores consideran que la diversidad es una característica muy importante en la combinación de clasificadores, siguiendo el supuesto de mejorar las debilidades de un modelo con las fortalezas de otro (Kuncheva & Whitaker, 2003). El objetivo del aprendizaje guiado por correlación negativa es producir redes neuronales individuales en un ensamble. Para promover la

diversidad, el aprendizaje guiado por correlación negativa introduce un término de penalización a la correlación en la función de error de cada red neuronal individual del ensamble. Todas las redes neuronales en el ensamble son entrenadas simultáneamente e iterativamente en el mismo conjunto de datos. La función de error E_i para la red neuronal i es definida por la siguiente ecuación:

$$E_i = \frac{1}{N} \sum_{j=1}^N E_i(j) = \frac{1}{N} \sum_{j=1}^N \left(\frac{1}{2} (F_i(j) - d(j))^2 + \lambda p_i(j) \right) \quad (3.4)$$

donde N es el tamaño del conjunto de entrenamiento, F_i es el valor devuelto por la red neuronal i , $d(j)$ es el valor esperado para la entrada j , p_i es la función de penalización por correlación de errores de diferentes redes neuronales. El parámetro $0 \leq \lambda \leq 1$ ajusta la fuerza de la penalización. La función de penalización p_i puede utilizar la siguiente ecuación:

$$p_i(j) = (F_i(j) - F(j)) \sum_{k \neq i} (F_k(j) - F(j)) \quad (3.5)$$

donde $F(j)$ es el valor devuelto por el ensamble.

En la construcción de las NNs individuales, el número de nodos ocultos es determinado en varias iteraciones. Las NNs miembros del ensamble no son entrenadas por separado. El entrenamiento es iterativo entre todos los miembros del ensamble. Si una NN del ensamble tiene un error alto en el entrenamiento este valor afecta al error de las otras NNs del ensamble. Específicamente, el error con el que se ajustan los pesos de una NN (E_i) cambia según p_i . La función p_i depende de la correlación de los errores de la NN actual con el error de las otras NNs, en la misma iteración.

El algoritmo constructivo (Islam, et al., 2003), superó a diversos métodos de combinación de modelos. Sus experimentos utilizaron siete conjuntos de clasificación como por ejemplo el problema australiano de asignación de tarjeta de crédito, también experimentaron con el problema predicción de múltiples pasos (500) de la serie de tiempo caótica Mackey Glass. Los resultados del algoritmo constructivo muestran evidencia empírica de la relación positiva entre la diversidad y la eficacia del ensamble. La correlación negativa mostró ser una buena estrategia para buscar diversidad entre los modelos. Motivar la cooperación entre modelos diversos en el proceso de entrenamiento también ayudó a mejorar los resultados. Esta estrategia de buscar diversidad y favorecer la cooperación entre modelos será aprovechada en esta investigación, en particular extendiéndola para que considere diferentes tipos de modelos.

3.3.2. Auto-organización de modelos supervisados

El método Especialización-Generalización (SpecGen por sus siglas en inglés), propuesto por (Kordik & Cerny, 2011), combina modelos de manera jerárquica, utiliza diferentes técnicas de ensambles ya conocidas como son *Bagging*, *Boosting*, *Stacking*, etc. El método SpecGen sigue una estrategia evolutiva que combina diferentes técnicas de ensambles y modelos base en una jerarquía tipo árbol. La estructura jerárquica es evolucionada para cada conjunto de datos por medio de programación genética. Los ensambles jerárquicos evolucionados demostraron versatilidad y además incrementaron la eficacia para todos los conjuntos de datos utilizados en los experimentos.

En SpecGen los modelos base fueron neuronas con diferentes funciones de activación. Los autores buscan un balance entre diversidad y precisión. La diversidad en el nivel bajo de combinación puede ser regulado por (Kordik & Cerny, 2011):

- Manipulación de datos
 - Características de entrada utilizadas,
 - Conjunto de instancias utilizadas en el entrenamiento;
- Manipulación de modelos
 - Distancia entre los parámetros de los modelos base,
 - Tipo y complejidad de la función de transferencia.

El método SpecGen crea varios modelos, los cuales son combinados con diferentes técnicas de ensambles. Los ensambles vuelven a ser combinados en el siguiente nivel de la jerarquía. El espacio de búsqueda para encontrar la mejor jerarquía es exponencial según el número de niveles a considerar. La combinación SpecGen puede ser optimizada por un algoritmo diseñado para evolucionar estructuras tipo árbol. Éste consiste de dos partes principales: la primera parte es el algoritmo evolutivo que es independiente del problema. La segunda parte se denomina Contexto y representa las operaciones dependientes del problema (por ejemplo pre-procesamiento de datos y cálculo de la aptitud para resolver el problema).

Los experimentos realizados por Kordik y Cerny incluyen diferentes conjuntos de datos de problemas de clasificación y regresión. Los resultados de SpecGen superaron las técnicas tradicionales para la construcción de ensambles. SpecGen tienen una complejidad mayor dado que prueba diversas técnicas de generación de ensambles y las combinar varias veces hasta encontrar una jerarquía adecuada para cada problema. Por otro lado, SpecGen solo trabaja con modelos de regresión simples. En esta investigación proponemos extender el método SpecGen para considerar modelos de predicción diferentes y el problema de predecir varios valores a futuro. Una primera estrategia es considerar diferentes horizontes de predicción en la construcción de la jerarquía.

3.4. Meta-aprendizaje en predicción

La teoría de “*No-free-lunch*” establece en esencia que, dados todos los escenarios posibles, todos los modelos tendrían el mismo desempeño. En otras palabras, no es posible que un solo modelo sea el mejor en todos los escenarios. Lemke y Gabrys (Lemke & Gabrys, 2010) proponen realizar una selección de un modelo o combinaciones de modelos para atacar el problema de predicción de múltiples pasos de series de tiempo. Los modelos de base que consideraron dichos autores en su investigación son en total 13 modelos estadísticos y 3 redes neuronales. Las estrategias de combinación que evaluaron son: promedio simple, promedio simple con podas, mejor desempeño, basado en la varianza, dos versiones de *ranking* y regresión.

Los meta-datos de las series de tiempo fueron extraídos considerando un grupo de 23 características, estas características están listadas en el Anexo II. En la primera parte de su trabajo utilizaron un algoritmo generador de reglas para generar el sistema experto. A continuación, utilizaron una estrategia de *ranking* y un agrupamiento utilizando *k-means* para definir los modelos a combinar. El *ranking* se basa en una función de distancia utilizando las características de las series de tiempo. Cuando una serie a predecir llega es agrupada. Cada serie tiene asociado el modelo de mejor desempeño. El modelo ganador es elegido por el número de veces que ha sido mejor que los otros modelos del grupo (*ranking*). Posteriormente, generaron tres expertos utilizando tres algoritmos de aprendizaje sobre el conjunto de meta-datos. Los algoritmos de aprendizaje utilizados son: red neuronal alimentada hacia adelante con 30 nodos en la capa oculta, árbol de decisión y SVM. Los expertos entrenados fueron evaluados en la tarea de predicción. Las series de tiempo utilizadas fueron tomadas de los torneos de predicción NN3 y NN5. Los tres expertos fueron comparados con la estrategia de selección de modelos basada en *ranking*. Los mejores resultados fueron obtenidos con su estrategia basada en *ranking*.

En el trabajo descrito en el párrafo anterior los autores llegaron a dos conclusiones: las características que funcionaron bien con un problema no necesariamente funcionan bien con otro. Probar agrupamiento de series de tiempo con mapas auto-organizados podría brindar nuevas oportunidades de mejorar los resultados. Por otro lado, en este trabajo se consideraron pocos modelos de inteligencia computacional. Aumentar el número de modelos y una búsqueda de mejores parámetros sobre los modelos a combinar incrementaría la posibilidad de obtener mejores resultados. La selección de nuevos modelos para el problema debe incluir un análisis previo, es decir, el desempeño de los nuevos modelos debe ser validado antes de incluirlos en la combinación de modelos (Kuncheva & Whitaker, 2003).

3.5. Resumen

Varios autores coinciden en que las combinaciones de modelos obtienen mejores resultados que el mejor de los modelos solo (Deutsch, et al., 1994), (Crone, et al., 2008), (Makridakis & Hibon, 2000). Sin embargo la pregunta de cuáles modelos combinar y cómo combinarlos sigue abierta. El compromiso a tener en cuenta es la diversidad entre los modelos y el desempeño individual de los mismos. Hay muchos trabajos en la combinación de métodos de clasificación, pero ellos no consideran el problema de manejar el tiempo, donde el error crece exponencialmente con el tamaño del horizonte de predicción. En esta sección varias estrategias para combinar modelos fueron presentadas. Las características principales de los trabajos mostrados son:

- Utilizar meta-características de desempeño para buscar diversidad en los modelos teniendo presente el desempeño individual.
- Guiar la selección de los modelos a combinar y la manera de combinarlos sin intervención del exterior (auto-organización). Generalmente, métodos de optimización como algoritmos evolutivos son utilizados para seleccionar los modelos y seleccionar la manera de combinarlos.
- Una estructura de árbol con diferentes tipos de modelos a combinar, diferentes tipos de combinaciones y diferentes horizontes de predicción podría adaptarse (auto-organizarse) mejor a las características de los series de tiempo caóticas que modelos adaptivos o métodos clásicos de combinación de modelos.

La siguiente sección reúne las características de los trabajos relacionados para combinar sus fortalezas y atacar sus debilidades.

Capítulo 4

Propuesta

Este capítulo presenta el objetivo principal de esta investigación, iniciando con un resumen del problema general tratado en esta investigación. Las siguientes secciones muestran las preguntas de investigación, las hipótesis y los objetivos planteados. El capítulo termina con las contribuciones esperadas al finalizar esta investigación.

4.1. Problemática

Varios dominios importantes para la sociedad trabajan con sistemas caóticos y necesitan mejorar sus estimaciones a largo plazo (ver Capítulo 1). En el área de predicción de series de tiempo, varios autores coinciden en que la combinación de modelos permite obtener los mejores resultados (Makridakis & Hibon, 2000), (Crone, et al., 2008), (Dzeroski & Zenko, 2004). Sin embargo, seleccionar los modelos de predicción a combinar y determinar la manera de combinarlos es un problema abierto. Esta investigación ataca este problema con la finalidad de mejorar la predicción a largo plazo de series de tiempo caóticas. La solución propuesta conjuga parte del conocimiento de las siguientes áreas: redes neuronales artificiales, teoría de ensambles, teoría de auto-organización, teoría del caos y análisis de series de tiempo. La presente investigación tomará las fortalezas de diversos trabajos en las áreas mencionadas para atacar sus debilidades y mejorar la predicción a largo plazo.

Los problemas que están asociados al problema principal, encontrados en la revisión del trabajo relacionado, son los siguientes:

- Las series de tiempo caóticas tienen un comportamiento sensible a cambios en las condiciones iniciales. Esto implica que para poder predecirlas indefinidamente sería necesario conocer todos los valores del pasado y que cada operación sea realizada con precisión infinita, lo cual no es posible. Entonces, son necesarios modelos que puedan

aprender el comportamiento del sistema y estimar valores a futuro utilizando los valores conocidos y operaciones computables.

- No existe un modelo único que tenga el mejor desempeño con todas las series de tiempo. El desempeño de los modelos de predicción varía principalmente según las características de la series de tiempo y del horizonte de predicción. Se requiere un método que permita seleccionar modelos de predicción de acuerdo a las características de la serie y del desempeño esperado de cada modelo.
- En la combinación de modelos es necesario considerar simultáneamente: la diversidad de los modelos a combinar, el desempeño de cada modelo y la forma en la que están siendo combinados los modelos. Desarrollar un método que considere estos tres factores eficazmente no es una tarea trivial.

En la naturaleza se observa que los componentes simples de un sistema pueden interactuar (organizarse) y presentar comportamientos caóticos. Por otra parte, la teoría de la auto-organización estudia las interacciones entre los componentes de sistemas complejos (Haken, 2006). En esta investigación utilizaremos esta teoría para guiar las interacciones de los modelos de predicción con la finalidad de que se acerquen al comportamiento caótico de la serie a predecir.

4.2. Preguntas de investigación e hipótesis

A partir de los problemas encontrados en la revisión de la literatura y de los resultados obtenidos a la fecha en las áreas de interés involucradas en esta investigación, se plantea encontrar respuestas a las siguientes preguntas:

- ¿Cómo se pueden encontrar automáticamente modelos de predicción adecuados a combinar y cuál es la manera correcta de combinarlos, con la finalidad de mejorar la predicción a largo plazo en series de tiempo caóticas?
- ¿Cómo se puede extraer y aprovechar el conocimiento previo de modelos de predicción que funcionan mejor en diferentes horizontes que en otros?
- ¿Puede la auto-organización mejorar el resultado global obtenido por la combinación de modelos de predicción?

Derivadas de estas preguntas, planteamos las siguientes tres hipótesis a probar:

- Una combinación eficaz de modelos de predicción puede definirse a partir de las características de las series de tiempo involucradas, utilizando conocimiento previo.
- Las características de desempeño en diferentes ventanas de tiempo pueden guiar la selección y combinación de modelos de predicción.

- Dado que la auto-organización de componentes de un sistema puede generar comportamientos caóticos y siguiendo reglas simples cómo competir y cooperar, es posible ordenar la interacción entre modelos de predicción para mejorar la predicción a largo plazo de series de tiempo caóticas.

4.3. Objetivo general

El objetivo principal de esta investigación es *desarrollar un método de meta-aprendizaje capaz de seleccionar un conjunto de modelos y combinarlos, de manera auto-organizada, considerando diferentes horizontes de predicción en series de tiempo caóticas.*

El método desarrollado deberá obtener en promedio un porcentaje de predicción a largo plazo mejor que el alcanzado por los métodos de combinación presentados en el estado del arte. El desempeño será medido utilizando las métricas SMAPE (ecuación (2.8)), MAE (ecuación (2.5)) y RMSE (ecuación (2.4)). El tamaño del horizonte de predicción (largo plazo) se definirá de acuerdo al dominio de origen de las series de tiempo involucradas en los casos de prueba. El desempeño del método propuesto se comparará con el trabajo actual más cercano a esta investigación (Lemke & Gabrys, 2010). También se analizará el desempeño del método propuesto con respecto al de otros modelos especializados en predicción a largo plazo: el modelo propuesto por García y Gómez (García Pedrero & Gómez Gil, 2009), basado en osciladores harmónicos y análisis espectral, el de (Andrawis, et al., 2011) basado en estadística y el de (Alarcon-Aquino & Barria, 2006), basado en redes wavelets. Se espera obtener una diferencia estadísticamente significativa en la eficacia de la predicción a largo plazo utilizando las series caóticas NN3, NN5, niveles de glucosa en la sangre de personas con diabetes², series tomadas de sensores de temperatura³, electroencefalogramas⁴, electrocardiogramas⁵, precipitaciones diarias⁶, y audios digitales⁷. Con respecto a los modelos involucrados en el diseño del algoritmo, se utilizarán los mapas auto-organizados SOM (Kohonen, 1990) para combinar modelos a través de las reglas de competir y cooperar. Asimismo, se usarán modelos de diferente tipo aplicados para diferentes ventanas de tiempo en el horizonte de predicción. Finalmente, se seguirá la estrategia de divide y vencerás dividiendo a la serie de tiempo en componentes estructurales, considerando que los componentes de la serie pueden ser más fáciles

² Series base de datos UCI disponibles en <http://archive.ics.uci.edu/ml/datasets/Diabetes>

³ Series de sensores disponibles en <http://www.spotseven.de/gecco-challenge/gecco-challenge-2013/>

⁴ Series base de datos UCI disponibles en <http://archive.ics.uci.edu/ml/datasets/EEG+Database>

⁵ Series base de datos Physionet disponibles en <http://www.physionet.org/physiobank/database/>

⁶ Series base de datos NASA disponibles en <http://pmm.nasa.gov/node/289>

⁷ Música generada por sistemas caóticos http://www.digital-daydreams.com/tech/show_tech.php?id=24

de aprender que la serie entera. Los modelos aprenderán cada componente y después las predicciones de los modelos serán combinadas.

4.4. Objetivos específicos

Los objetivos específicos planteados para alcanzar el objetivo general son:

1. Establecer un método de selección de los mejores modelos en diferentes ventanas de predicción, a fin de mejorar la eficacia en la predicción a largo plazo con respecto a otros modelos existentes.
2. Construir un algoritmo de meta-aprendizaje capaz de extraer características de las series de tiempo y del comportamiento de los modelos de predicción, a fin de mejorar la combinación de los modelos.
3. Desarrollar un método de auto-organización de modelos, promoviendo la colaboración entre ellos, a fin de mejorar la predicción a largo plazo de series de tiempo caóticas.

4.5. Contribuciones

Las principales contribuciones esperadas de esta investigación son:

1. Un nuevo método para combinar modelos de predicción, considerando auto-organización de los modelos y diferentes horizontes de predicción. El método propuesto obtendrá mejor eficacia en la predicción a largo plazo que el reportado en el trabajo más cercano a éste, encontrado al momento de escribir este documento, realizado por (Lemke & Gabrys, 2010).
2. Un algoritmo auto-organizado de meta-aprendizaje aplicado en el contexto de predicción a largo plazo de series de tiempo caóticas.

Al momento de publicarse este documento, no se ha encontrado en la revisión del estado del arte un trabajo que implemente el uso de auto-organización, división de horizontes de predicción y meta-aprendizaje para mejorar la predicción a largo plazo en series de tiempo caóticas.

Capítulo 5

Metodología

En este capítulo se describen el camino a realizarse a fin de demostrar o refutar las hipótesis de esta investigación, siguiendo un enfoque sistemático. Cada tarea incluye un proceso de validación interna (con datos conocidos) y validación externa (con datos que no fueron mostrados en el entrenamiento) antes de declararla terminada. Algunas de estas tareas ya han sido realizadas y otras están en proceso de realización actualmente, según se detalla en el capítulo 5. Estas actividades son:

1. Recopilar un conjunto de datos y modelos. Esta tarea implica la selección de un conjunto de series de tiempo caóticas y un conjunto de modelos de predicción adecuados para entrenar y probar el modelo. En esta tarea también se incluye:
 - a. Establecer un método de limpieza y pre-procesamiento de datos. Se debe considerar el eliminar ruido, principalmente *outliers*, y estimar valores faltantes en las series.
 - b. Definir características representativas de las series de tiempo. Algunos ejemplos de estas características son la desviación estándar, tendencia, inclinación y máximo exponente de Lyapunov (Casdagli, 1992). Ver el apéndice II para un listado detallado.
 - c. Identificar los parámetros de los modelos de predicción seleccionados y rango de valores recomendados, como son: número de retrasos, número de neuronas ocultas y algoritmos de entrenamiento.
 - d. Documentar un conjunto de estrategias básicas de combinación de modelos. Estas estrategias incluyen: promedio simple, *stacking* con distribución de probabilidad (Yang, et al., 2001), y redes modulares (Melin, et al., 2007)).
 - e. Seleccionar funciones de evaluación. Implica la definición un conjunto de funciones para evaluar modelos en la tarea de predicción de múltiples pasos. Las funciones más comunes para valorar la predicción se presentan en el Capítulo 2.
 - f. Seleccionar los tipos de validación a realizar, con cada uno determinar el método de validación.

2. Construir un extractor automático de meta datos. Los datos a extraer se dividen en dos grupos: características de las series y desempeño del modelo. El método debe cumplir los siguientes requerimientos:
 - Buscar el modelo más eficaz para predecir una determinada serie de tiempo en un espacio de búsqueda previamente definido. Se consideran modelos diferentes a los que provienen de distintos tipos de modelos, así como los modelos del mismo tipo pero con diferentes parámetros.
 - Permitir evaluar los modelos en diferentes horizontes de predicción.
 - Validar la eficacia del modelo con cada serie.
 - Permitir aumentar nuevos modelos de predicción.
 - Extraer características de las series de tiempo.

En esta tarea también se incluyen los siguientes sub-tareas:

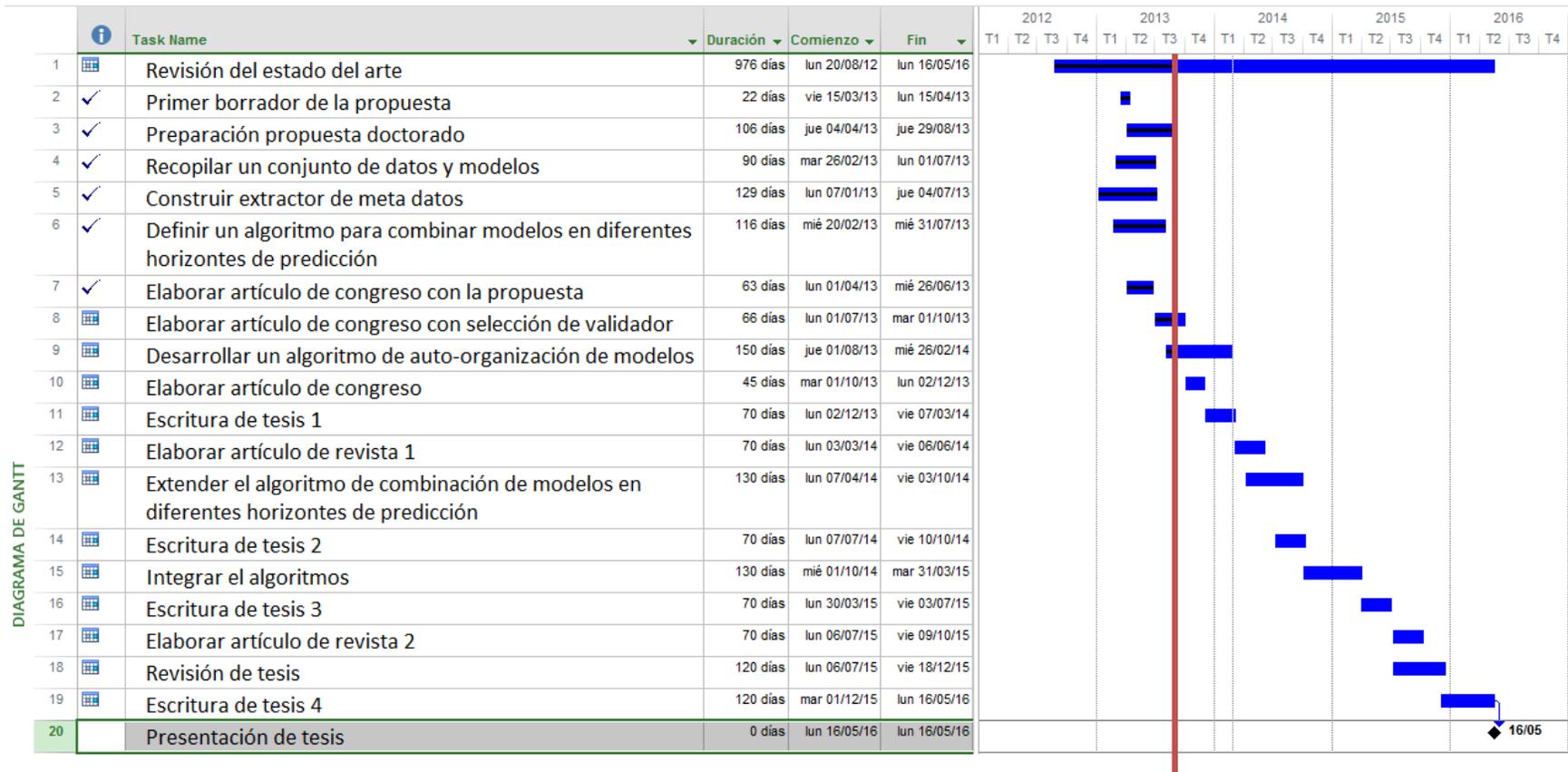
- a. Revisar métodos de búsqueda en la literatura que sean flexibles y permita aumentar nuevos modelos.
 - b. Extender el método de búsqueda seleccionado considerando los requerimientos de esta tarea.
 - c. Evaluar la eficacia del método desarrollado evaluando los modelos que devuelve para cada serie de tiempo. Las predicciones de los modelos seleccionados serán comparadas con los resultados obtenidos por (García Pedrero & Gómez Gil, 2009) y (Alarcon-Aquino & Barria, 2006). Si los resultados no son satisfactorios se regresará a la sub-tarea (b).
3. Definir un algoritmo para combinar modelos en diferentes horizontes de predicción. Los pasos a realizar en esta tarea son:
 - a. Evaluar las estrategias de combinación de modelos de predicción existentes. Decidir cuál de ellas, si existe, permite combinar modelos eficazmente y explotar el mejor desempeño de los diferentes modelos en diferentes horizontes de predicción.
 - b. Proponer una estrategia de combinación utilizando diferentes horizontes de predicción.
 - c. Evaluar el algoritmo propuesto comparándolo con las estrategias de combinación de modelos encontradas en la revisión del trabajo relacionado.
 4. Desarrollar un algoritmo de auto-organización de modelos, motivando la cooperación entre ellos en un proceso de meta-aprendizaje. Los pasos a realizar en esta tarea son los siguientes:
 - a. Analizar en la literatura las estrategias de auto-organización, en particular las enfocadas en la construcción de combinaciones de modelos.

- b. Definir un algoritmo de meta-aprendizaje utilizando la estrategia de auto-organización seleccionada.
 - c. Comparar el método de meta-aprendizaje auto-organizado diseñado con el método de meta-aprendizaje de Lemke y Gabrys (Lemke & Gabrys, 2010).
5. Extender el algoritmo de combinación de modelos en diferentes horizontes de predicción con la finalidad de mejorar la eficacia. Adicionalmente, considerar el impacto en la eficiencia del algoritmo desarrollado. Los pasos a realizar en esta actividad son:
- a. Revisar las estrategias de combinación de modelos de clasificación del estado del arte, considerando su aplicabilidad en la combinación de modelos de predicción de series de tiempo.
 - b. Evaluar las estrategias de combinación de modelos en el contexto de predicción de series de tiempo.
 - c. Extender el algoritmo de combinación de modelos propuesto en la tarea 3 con una de las estrategias de combinación de modelos del estado del arte, con la finalidad de mejorar la eficacia. Considerar la descomposición de la tarea de predecir la serie de tiempo completa, en las sub-tareas de predecir los componentes de las series de tiempo estructurales y en el dominio de la frecuencia. Al final, los resultados de las sub-tareas son integrados para dar una predicción final.
 - d. Comparar la eficacia y la eficiencia del algoritmo extendido en esta tarea y el algoritmo propuesto en la tarea 3.
6. Integrar el algoritmo de combinación en diferentes horizontes de predicción con el algoritmo de auto-organización de modelos en un proceso de meta-aprendizaje. Las sub-tareas relacionadas son las siguientes:
- a. Analizar los puntos en los que los algoritmos pueden ser integrados.
 - b. Analizar la complejidad de los algoritmos desarrollados para identificar los posibles cuellos de botella de los algoritmos.
 - c. Integrar los algoritmos considerando un equilibrio entre eficacia y eficiencia.
 - d. Evaluar el algoritmo final y compararlos con los trabajos del estado del arte.

En el siguiente capítulo se describen los resultados preliminares obtenidos a la fecha, los cuales han sido desarrollados siguiendo las tareas anteriormente descritas. En la siguiente sección se presenta el calendario de trabajo desglosado por trimestres. Se planea defender la tesis doctoral al final del abril de 2016

5.1. Plan de trabajo

El cronograma de actividades propuesto se muestra a continuación. Las tareas corresponden a los puntos principales de la metodología propuesta. La segunda columna señala las tareas completadas. Las iniciales Q1 Q4 representan bloques de tres meses de tiempo-calendario. La línea roja corresponde a la fecha de presentación de este documento.



Capítulo 6

Resultados preliminares

Este capítulo detalla los resultados obtenidos a la fecha. Con el fin de clarificar al lector las funciones de cada uno de los algoritmos involucrados y sus relaciones, se inicia con una descripción general del sistema de predicción propuesto. En el resto de las secciones se describen: el proceso de recopilación de datos y modelos, el detalle de cada uno de los algoritmos realizados a la fecha, los datos utilizados para la obtención de resultados experimentales y una discusión de dichos resultados. Asimismo, se incluye una descripción de algunas de las estrategias que se plantean para resolver las tareas pendientes a realizarse.

6.1. Diagrama del método general

El esquema general de la solución propuesta en esta investigación se muestra en la Figura 6.2. En el primer paso, un proceso de *meta-aprendizaje* genera un *meta-modelo*. El meta-modelo devuelve la combinación de modelos recomendada para predecir una nueva serie de tiempo. El proceso de meta-aprendizaje considera características de las series de tiempo para poder separarlas; también utiliza características de los modelos tales como el desempeño de los modelos y la correlación negativa entre modelos. La manera de combinar los modelos se apoya en la auto-organización, a fin de asignar pesos a los modelos y la combinación temporal utiliza la estrategia de “divide y vencerás”. La combinación puede realizarse utilizando redes modulares.

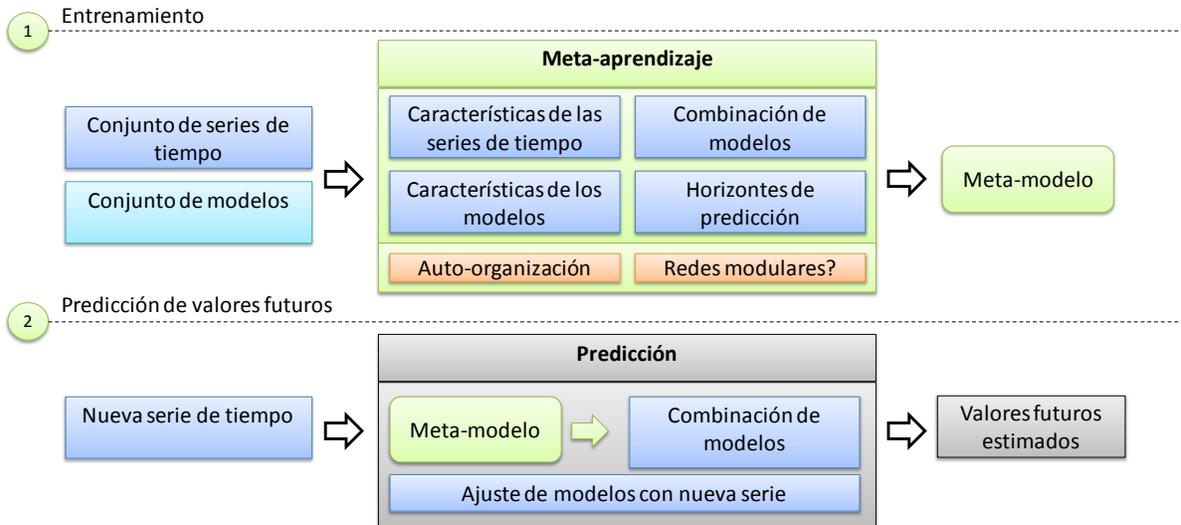


Figura 6.1. Esquema de la solución propuesta en esta investigación. El primer paso es un proceso de meta-aprendizaje del cual se obtiene un meta-modelo. El meta-modelo devuelve la combinación de modelos recomendada para la predicción de una serie de tiempo particular.

6.2. Recopilación de datos y modelos

Esta sección describe las series de tiempo utilizadas, el pre-procesamiento realizado. También incluye una descripción de los modelos base que fueron considerados.

6.2.1. Pre-procesamiento

Como primer paso se analizaron diferentes conjuntos de series de tiempo utilizadas en torneos de predicción y en evaluación de modelos de predicción reportados en la literatura. Algunas de las series escogidas presentaron datos faltantes y valores no esperados. Los valores no esperados son difíciles de eliminar, dado que el comportamiento de las series de tiempo caóticas es muy irregular, por lo que se optó por eliminar solo los valores que vayan en contra del dominio de la serie. Una estrategia simple y eficaz para completar valores faltantes es la regresión lineal entre los puntos extremos. Esta estrategia fue escogida por su simplicidad, aunque en investigaciones futuras podría evaluarse la mejora del desempeño utilizando otras estrategias para llenar los valores faltantes.

Los conjuntos reducidos de series de tiempo de los torneos NN3 y NN5 (Crone, 2008) fueron analizados para determinar si presentan un comportamiento caótico. El primer paso fue graficar los mapas de retorno de las series para realizar un análisis visual, el cual es un método comúnmente usado para identificar caos en series de tiempo (Gómez-Gil, 1998). Todas las series presentaron atractores extraños lo que es un claro indicio de comportamiento caótico. El segundo paso fue

calcular el máximo exponente de Lyapunov de cada serie. Todas las series presentaron un exponente de Lyapunov positivo, esto implica que las series tienen un comportamiento caótico.

6.2.2. Características de las series de tiempo y de los modelos de predicción

Una parte importante del proceso de meta-aprendizaje es el poder separar las series de tiempo en grupos significativos con respecto a ciertas características. Wang y colaboradores (Wang, et al., 2009) proponen un conjunto de características para agrupar series de tiempo, el cual fue enriquecido en el trabajo (Lemke & Gabrys, 2010). Utilizando ambos trabajos, se seleccionaron las características que se utilizarán en esta investigación, detalladas en el Anexo II.

Con respecto a modelos de predicción, NAR (Leontaritis & Billings, 1985) y ARIMA (Box, et al., 1994) fueron seleccionados ya que presentan buenos resultados. La Tabla 6.1 presenta los parámetros recomendados en la literatura (Box, et al., 1994) para los modelo ARIMA, los cuales fueron seleccionados en este trabajo. La primera columna de la tabla contiene el nombre del parámetro y la segunda el rango de valores permitido para dicho parámetro.

Tabla 6.1. Rango de valores recomendados en la literatura para modelos ARIMA

Parámetros	Rango de valores
Número de términos auto regresivos (p)	{0, 1, 2}
Número de diferencias no-estacionales (d)	{1, 2}
Número de errores de predicción del pasado (q)	{0, 1, 2}
Estacionalidad	{0, 7, 12}

Los modelos NAR tienen tres parámetros principales que determinan su comportamiento. El primero es el número de neuronas de retraso, el cual se determina en función del espacio embebido contenido en cada serie. Aún y cuando éste todavía es un problema abierto, en la teoría del caos puede encontrarse la definición de varios métodos para calcular el número mínimo de elementos de retardo requeridos (Mees, 2001). Tres métodos fueron analizados para determinar el rango de valores a probar como elementos de retardo:

- Método de Cao (Cao, 1997), basado en encontrar el número mínimo de observaciones antes de encontrar un cambio de trayectoria en el atractor.
- Método de Lei (Lei, et al., 2002), basado en geometría *Symplectic* y *eigen* valores.
- Método del falso vecino más cercano (Kennel, et al., 1992), basado en construcción geométrica.

El método Cao puede ser visto como una evolución del método del falso vecino más cercano; su principal ventaja es que descarta valores que no modelan apropiadamente el comportamiento de la serie de tiempo. La ventaja del método de Lei es que permite apreciar visualmente el comportamiento del sistema en función de los elementos de retardo. Ambos métodos fueron utilizados para determinar rangos de valores para las neuronas de retraso.

Es bien sabido que determinar el número de neuronas necesarias y suficientes en la capa oculta de una red neuronal multicapa continúa siendo un problema abierto (Haykin, 2005). En los resultados mostrados en este documento, los rangos de los valores posibles para el número de neuronas escondidas se determinaron en función de las recomendaciones encontradas en otros trabajos de predicción de series de tiempo y a través de una evaluación empírica del modelo NAR con los conjuntos reducidos NN3 y NN5.

La determinación del algoritmo de entrenamiento para NAR se realizó empíricamente, generando y probando varios modelos usando 11 series de tiempo del torneo NN3. Es importante hacer notar que los algoritmos de entrenamiento contienen un paso de inicialización aleatoria de pesos, por lo que pueden producir resultados diferentes en cada ejecución, aún y cuando se utilice el mismo conjunto de entrenamiento. Por esta razón, el experimento para selección de algoritmos de entrenamiento fue repetido 10 veces, a fin de disminuir el efecto de la inicialización de pesos aleatoria. De los 10 algoritmos de entrenamiento disponibles en MATLAB (Hudson Beale, et al., 2012), plataforma en la que se llevaron a cabo los experimentos, los algoritmos que en promedio fueron mejores para la tarea de predicción de series de tiempo caóticas son presentados en la Tabla 6.2. En la misma tabla también se muestran los rangos de valores escogidos para los parámetros involucrados en los modelos NAR.

Tabla 6.2. Rango de valores seleccionados para los modelos NAR

Parámetros	Rango de valores
Número de elementos de retardo	{3, 10, 15, 20, 25}
Número de neuronas en la capa oculta	{10, 20, 30}
Algoritmo de entrenamiento	<i>Bayesian regulation back-propagation (BP), Conjugate gradient BP with Fletcher-Reeves updates, Levenberg-Marquardt BP</i>

6.2.3. Método de validación

La selección de un modelo para una tarea específica requiere de un método de validación apropiado, a fin de determinar que opción se comporta mejor desde un punto de vista estadístico. En la validación de modelos de predicción, los dos métodos más eficaces son la validación cruzada de diez pliegues y la validación cruzada Monte Carlo.

A fin de determinar cuál de los métodos de evaluación de modelos es el más apropiado para sistemas de predicción a largo plazo en series caótica, se realizó un experimento comparativo, el cual ha sido aceptado para publicarse en un congreso con arbitraje internacional (Fonseca & Gómez Gil, 2013). La eficacia de los métodos de validación fue evaluada con las series del torneo de predicción NN3. Los resultados obtenidos, que se muestran en la Tabla 6.3, fueron analizados y no se encontró una diferencia estadísticamente significativa entre ambos métodos. Sin embargo, la desviación estándar de los errores de estimación utilizando la validación cruzada Monte Carlo es menor que el obtenido por validación cruzada de 10 pliegues. Dado que la desviación estándar es un índice de la precisión, se concluyó que la validación cruzada Monte Carlo es más precisa que su contraparte de 10 pliegues, para la tarea de seleccionar un modelo de predicción para una serie de tiempo. Asimismo, se encontró que en la tarea de evaluar modelos con diferentes series de tiempo la precisión es mejor utilizando la validación cruzada de diez pliegues que utilizando Monte Carlo (Fonseca & Gómez Gil, 2013).

Tabla 6.3. Promedio de MSE por series y modelos para la validación cruzada de diez pliegues y la validación cruzada Monte Carlo (MCCV) con las series de tiempo del torneo NN3

Validación cruzada	Promedio de MSE	Desviación estándar por series	Desviación estándar por modelos
Diez pliegues	618.19	757.98	455.53
MCCV	655.80	641.72	568.65

Considerando los resultados obtenidos del experimento comparativo, la selección de modelos para las series de tiempo utilizará la validación cruzada Monte Carlo, mientras que la validación del algoritmo propuesto al compararlo con otros modelos utilizará la validación cruzada de diez pliegues.

6.3. Método extractor de meta-datos

Como puede observarse en la figura 6.1, la primera etapa de este método consiste en extraer las características seleccionadas de las series de tiempo. La segunda etapa implica una búsqueda de los n -mejores modelos para cada serie. Aquí se requiere la habilidad de poder aumentar nuevos modelos al proceso de evaluación y selección. Una opción es crear una sola función principal que se adapte a todos los cambios del sistema, pero con cada nuevo modelo sería necesario cambiar esta función principal y podría afectar a los demás componentes del método. La estrategia de envoltura (Kohavi & John, 1997) permite cambiar componentes sin tener que afectar a todo el sistema. Por esta razón, seleccionamos la estrategia de envolturas para poder aumentar modelos e incluso probar con diferentes estrategias de búsqueda.

El diseño de la solución para el extractor de meta-datos se muestra en la Figura 6.2. El extractor recibe un conjunto de modelos, cada modelo tiene parámetros y los posibles valores que pueden tomar. Las diferentes combinaciones de modelos que se pueden obtener variando el tipo de modelo y los valores de los parámetros conforman el espacio de búsqueda. El extractor de meta-datos también recibe un conjunto de series de tiempo y retorna los n modelos que mejor desempeño presentan por cada serie de tiempo. La selección de modelos pasa por proceso de validación cruzada Monte Carlo con la finalidad de mejorar la estimación del desempeño de cada modelo con cada serie.

Como se comentó anteriormente, la estrategia de “envoltura de modelo” le permite al método interactuar con diferentes modelos, sin cambiar el comportamiento de los demás componentes. El proceso inicia con el componente validador, donde la función validar inicia dividiendo cada una de las series en dos sub-series: una para entrenamiento, llamada *serieE*, y una serie para prueba, identificado como *serieP* (ver Figura 6.2). El tamaño de *serieE* está formado por la suma de dos valores un valor r definido para cada serie como requerido para entrenamiento y un número aleatorio de observaciones. Las siguientes h observaciones de la serie representan la salida esperada para el horizonte de predicción y conforman la serie de prueba *serieP*. El componente evaluador entrena cada modelo por medio de la respectiva envoltura. El modelo entrenado es utilizado para predecir los siguientes h valores de la serie. La predicción obtenida es comparada con la predicción esperada *serieP*. Los resultados de cada modelo también son evaluados en diferentes horizontes de predicción. El validador retorna los n mejores modelos obtenidos para cada serie de tiempo. La estrategia de búsqueda utilizada en los primeros experimentos es de una búsqueda en profundidad; esto garantiza que la solución devuelta es completa en el espacio de búsqueda.

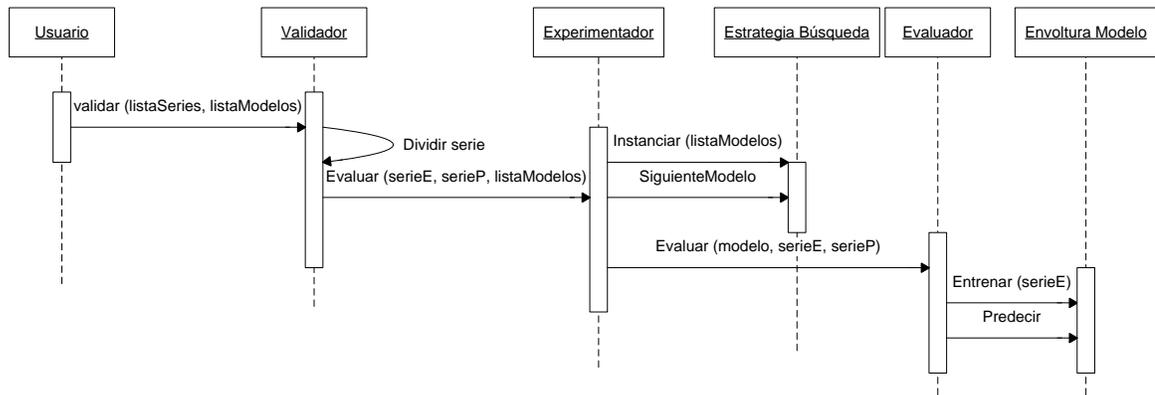


Figura 6.2. Diseño de la etapa de búsqueda del extractor de meta-datos. La entrada es un conjunto de modelos con sus posibles parámetros y un conjunto de series de tiempo. La salida es una lista de los n -mejores modelos para cada serie de tiempo.

6.4. Combinación temporal de modelos

Un método que combine diferentes modelos en diferentes horizontes de predicción puede obtener mejores resultados que solo combinando diferentes modelos sin considerar la naturaleza del horizonte. En este trabajo se propone dividir el horizonte de predicción en tres secciones, cada una con el mismo número de elementos, y que se identifican en este documento como: corto, mediano y largo plazo. Nuestro objetivo es combinar los modelos con mejor desempeño a corto, mediano y largo plazo en la misma serie de tiempo, como se ilustra en la Figura 6.3; **Error! No se encuentra el origen de la referencia.** A esta estrategia la denominamos “combinación temporal”. El algoritmo de combinación temporal funciona de la siguiente manera: Primero, los modelos seleccionados son entrenados, posteriormente cada modelo estima el horizonte completo de predicción. La predicción correspondiente al corto plazo es calculada a través del promedio de los n -mejores modelos a corto plazo. De igual manera se obtienen las predicciones a mediano y largo plazo. Al final los componentes son concatenados para obtener la predicción total.

La selección de los mejores modelos es realizada con el buscador de modelos del extractor de meta-datos descrito en la sección **Error! No se encuentra el origen de la referencia.** La serie de entrenamiento original se divide en un segmento de entrenamiento y un segmento de validación. Por cada modelo, el SMAPE es calculado entre la predicción obtenida y la esperada en cada horizonte de predicción. Al final, cada modelo tendrá tres valores de SMAPE, correspondientes a las secciones del horizonte de corto, mediano y largo plazo respectivamente. Los modelos

seleccionados para el horizonte a corto plazo son aquellos que tengan menor valor de SMAPE a corto plazo en promedio después de realizar una validación cruzada Monte Carlo. El mismo procedimiento es seguido para seleccionar los modelos de los horizontes de mediano y largo plazo.

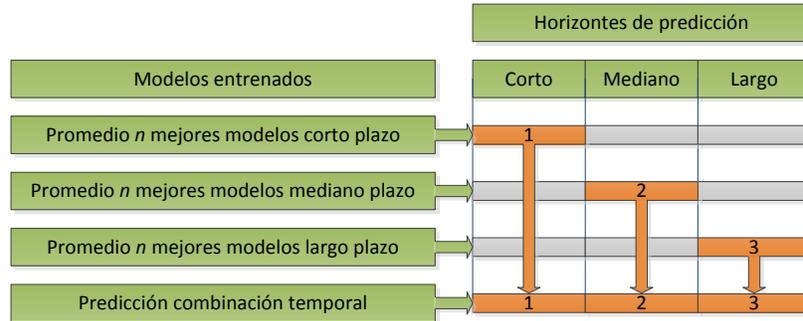


Figura 6.3. Combinación temporal de modelos de predicción previamente entrenados. El resultado está compuesto por la predicción de los modelos en sus diferentes horizontes de predicción.

Una vez definida la combinación temporal, los modelos que la conforman son entrenados con la serie de entrenamiento original. En los resultados reportados en este documento, las predicciones de los n -mejores modelos a corto plazo son combinadas utilizando promedio simple y el proceso es repetido con los modelos de mediano y largo plazo. Las predicciones son representadas como $\{y^s\}$, $\{y^m\}$, y $\{y^l\}$ para los modelos seleccionados de corto, mediano y largo plazo, respectivamente. Los segmentos corto, mediano y largo plazo tienen tamaño b . La predicción temporal completa C_T es obtenida al unir los tres horizontes de predicción. La unión está expresada en la ecuación:

$$C_T = \left\{ \left\{ y_i^s, n + 1 \leq i < n + b \right\} \cup \left\{ y_i^m, n + b \leq i < n + 2b \right\} \right. \\ \left. \cup \left\{ y_i^l, n + 2b \leq i < n + 3b \right\} \right\} \quad (6.1)$$

La predicción de C_T puede evaluarse calculando el SMAPE comparado la predicción obtenida con la predicción esperada. Adicionalmente, se evalúa el resultado según las secciones de corto, mediano y largo plazo.

6.5. Descripción de datos para la evaluación empírica

Un conjunto de diez series de tiempo se seleccionó para realizar una evaluación de los algoritmos desarrollados a la fecha. Este conjunto incluye series de tiempo de los torneos internacionales NN3 y NN5, también incluye series generadas con modelos conocidos como la serie Mackey Glass y series obtenidas de la demanda de vuelos en una aerolínea. La serie Mackey Glass es comúnmente utilizada como prototipo de series caóticas y se obtiene de integrar la ecuación (Mackey & Glass, 1977):

$$\frac{ds}{dt} = \frac{0.2s(t - 17)}{1 + s^{10}(t - 17)} \quad (6.2)$$

Las series escogidas tienen diferente número de observaciones que van desde 103 hasta 800 y los horizontes de predicción deseados varían desde 18 hasta 250 valores futuros. La figura 6.4 muestra dos de las series de tiempo y sus mapas de retorno.

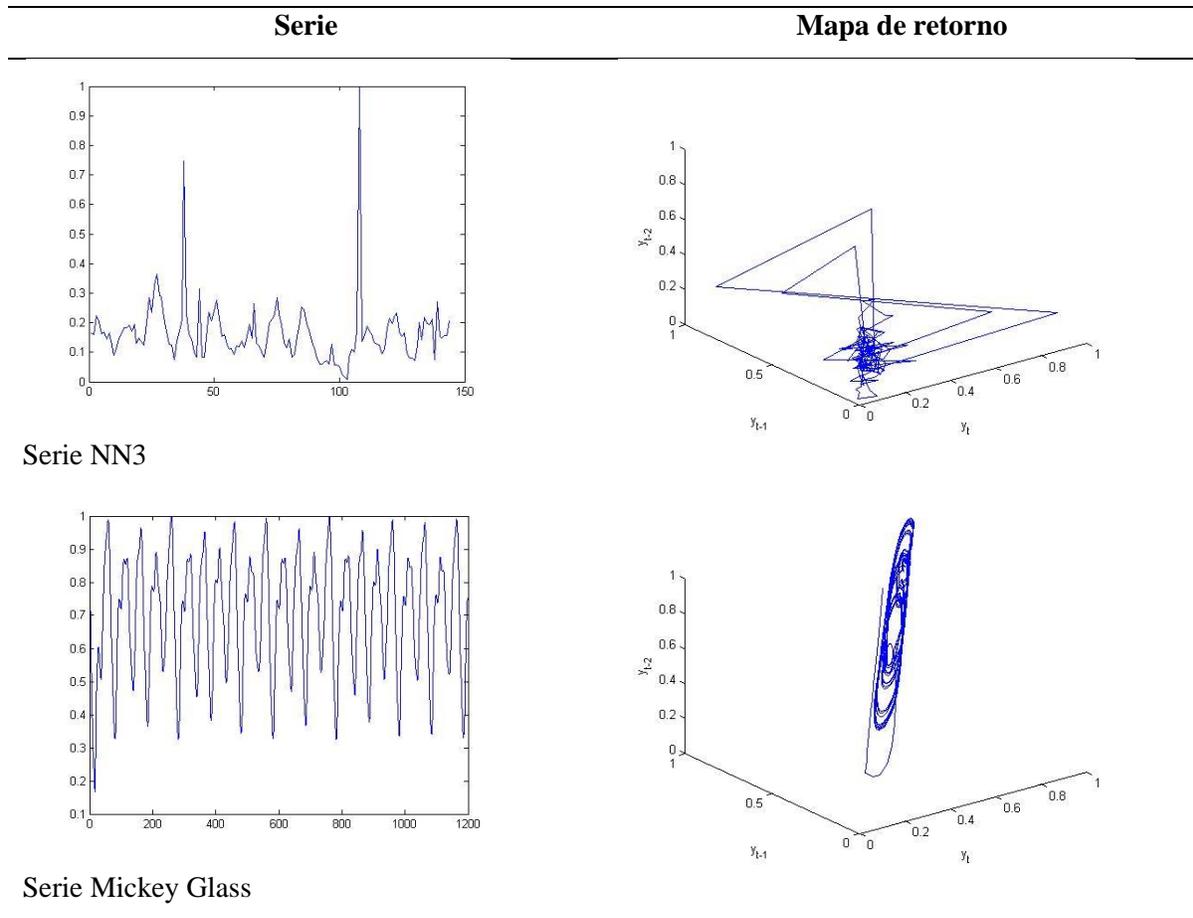


Figura 6.4. Dos ejemplos de las series utilizadas en la primera evaluación. Cada serie está acompañada de su mapa de retorno.

6.6. Caso de estudio y análisis de resultados

Se diseñó un caso de estudio cuyo objetivo es probar si la combinación temporal puede tener un mejor desempeño que la combinación de los n -mejores modelos. El conjunto de series descrito anteriormente fue dividido en un conjunto de entrenamiento y validación y un conjunto de prueba. La selección de los modelos para la combinación temporal y para la combinación de los n -mejores

utilizó el proceso de búsqueda del extractor de meta-datos. La selección fue realizada con el conjunto de entrenamiento y validación. Las combinaciones de modelos fueron evaluadas con el conjunto reservado para pruebas. La evaluación consistió en comparar la predicción obtenida con los valores esperados utilizando las funciones RMSE y SMAPE. Cabe recordar que el mejor valor esperado de RMSE es 0 y el peor es 1, dado que las series están normalizadas; para el caso de SMAPE el mejor valor esperado es 0 y el peor 200 (ver sección 2.4. para un explicación detallada de estas funciones) La Tabla 6.4 muestra los resultados de eficacia obtenidos con una combinación temporal con 3 modelos por cada horizonte de tiempo y los resultados de la combinación de los 3 mejores modelos por cada serie.

Tabla 6.4. Resultados de la combinación temporal y la combinación de los 3 mejores modelos por cada serie de tiempo.

No.	H	Combinación de 3 mejores modelos		Combinación temporal	
		RMSE	SMAPE	RMSE	SMAPE
1	18	0.31	44.02	0.30	41.72
2	18	0.67	101.17	0.57	95.59
3	18	0.14	28.25	0.15	30.40
4	18	0.08	41.36	0.05	30.13
5	56	0.08	26.89	0.07	25.98
6	56	0.07	12.97	0.07	15.14
7	50	0.54	148.19	0.55	140.64
8	250	3.14E-04	0.03	3.11E-04	0.03
9	20	0.01	1.32	0.02	1.44
10	256	4.04E-07	3.13	7.42E-05	3.13
Promedio		0.19 +/-0.23	40.73 +/-45.77	0.18 +/-0.21	38.42 +/-43.18

Estos resultados muestran que la combinación temporal es más eficaz con ciertas series de tiempo y la combinación de los n -mejores modelos funciona mejor con otras series. Un proceso de selección del método de combinación en función de las características de las series podría mejorar la eficacia en la predicción de series de tiempo. Este proceso de selección puede verse como un problema de meta-aprendizaje y será tratado en esta investigación. Por otra parte, en promedio, la combinación temporal fue más eficaz que la combinación de los n -mejores modelos. Sin embargo, una prueba de significancia estadística utilizando el método una prueba t (Kanji, 2006) indicó que esta diferencia no es significativa.

El experimento fue repetido variando el número de elementos a combinar, representado por n , primero n incrementó a razón 1 hasta llegar al valor de 5, después los incrementos fueron de tamaño 5. Los resultados de eficacia promedio obtenidos usando todas las series y medidos con SMAPE se muestran en la figura 6.5, donde se observa que la combinación temporal obtuvo mejores resultados que la combinación de los n -mejores modelos. Adicionalmente, puede notarse en la figura que los resultados de eficacia de ambos métodos tienden a ser los mismos a partir de la combinación de 15 modelos.

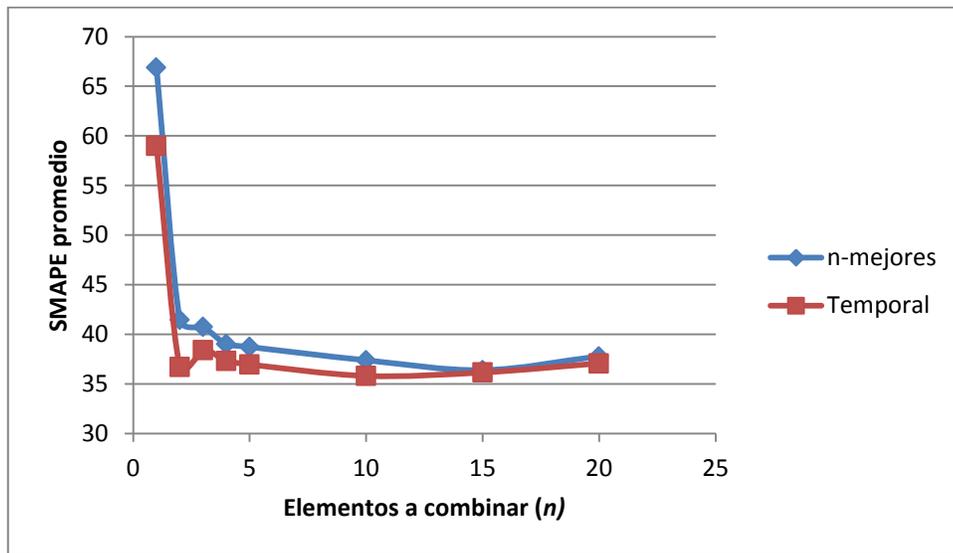


Figura 6.5. Resultados de la combinación temporal y la combinación de los n -mejores modelos variando el número de modelos a combinar, éste es el tamaño de n .

Analizando estos resultados y considerando las hipótesis descritas en esta investigación, se teoriza que los resultados de la combinación temporal pueden ser mejorados separando la serie en componentes⁸. Un ejemplo de la aplicación de esta estrategia es el reportado en (Adeodato, et al., 2008) donde la serie es separada en componentes. Después cada componente es modelado y una predicción es obtenida por cada uno. Finalmente, las predicciones pueden combinarse utilizando un tercer modelo.

6.7. Avances sobre meta-aprendizaje auto-organizado

En una primera etapa de meta-aprendizaje, las series pueden auto-organizarse en función de sus características, considerando el desempeño al predecir cada serie y la correlación entre los errores

⁸ La separación de series en componentes puede verse en la sección 2.5.

de predicción de los modelos. Se propone de manera preliminar que la auto-organización se lleve a cabo utilizando redes neuronales SOM, las cuales siguen dos guías simples: competir y cooperar.

La figura 6.6 ilustra la idea de los mapas auto-organizados resultantes. En la predicción de una nueva serie, el mapa auto-organizado obtenido en la primera etapa permite encontrar las series con las que la nueva serie comparte características semejantes. Las series a su vez están asociadas con un conjunto de modelos también auto-organizados, considerando la eficacia de predicción, y la correlación de los errores de predicción. Los modelos a combinar pueden tomarse a partir de los modelos que mejor desempeño tuvieron con las series de tiempo más semejantes a la nueva serie de tiempo. La combinación puede mejorar cambiando el promedio simple por un promedio pesado. Los pesos pueden obtenerse en función de la distancia euclidiana entre los vectores de características que representan a las series de tiempo. Los modelos asociados con la serie más cercana tendrán un mayor peso en la combinación

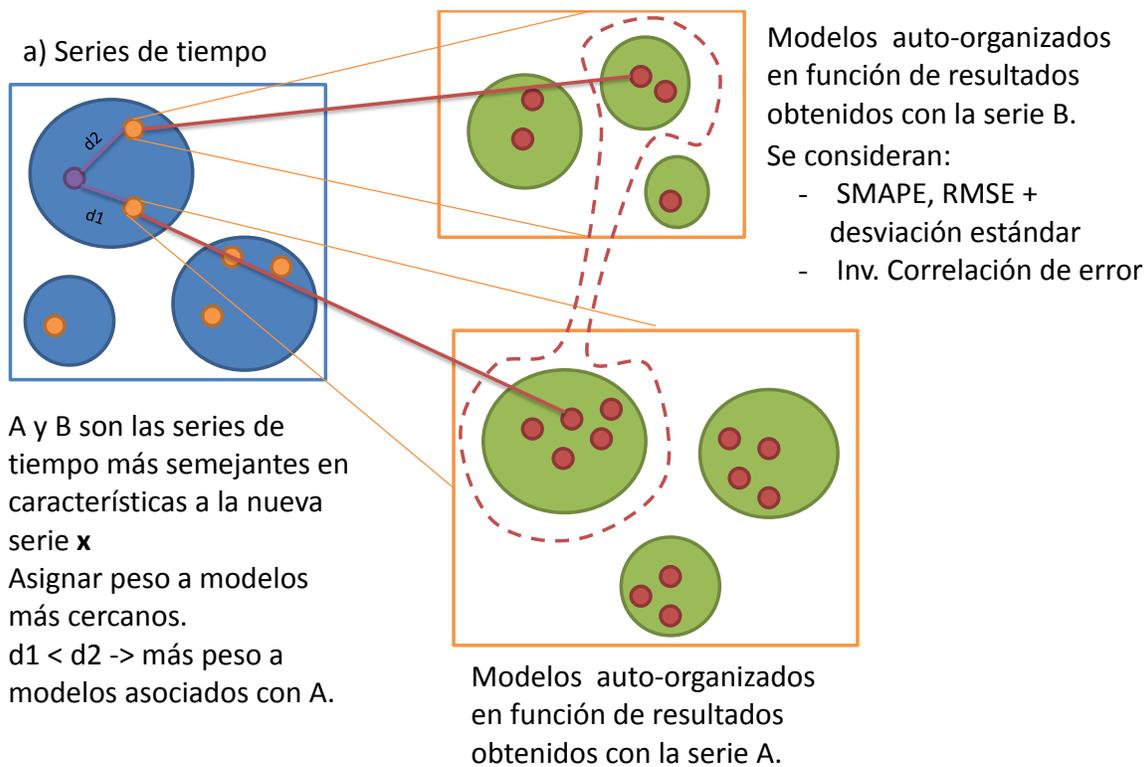


Figura 6.6. Propuesta de meta-aprendizaje auto-organizado. En la figura el bloque a) corresponde a un mapa auto-organizado de series de tiempo representadas por vectores de características. Cada serie de tiempo tiene asociado un mapa auto-organizado de los modelos en función de su desempeño.

6.8. Comentarios finales

Los resultados obtenidos a la fecha cubren los primeros tres puntos de la metodología descrita en el capítulo 5. Básicamente, el producto obtenido a la fecha es un algoritmo de combinación de modelos que considera diferentes secciones en el horizonte de predicción y un buscador de mejores modelos basado en envolturas al cual se ha denominado extractor de meta-datos. El caso de estudio ejecutado mostró que la combinación temporal en promedio obtuvo mejor resultado que combinar los n -mejores modelos por cada serie de tiempo. Sin embargo la diferencia entre ambos métodos no es estadísticamente significativa.

Estos resultados aún no incluyen auto-organización, combinación de modelos basada en pesos por secciones del horizonte de predicción ni cooperación, que son puntos fundamentales de esta investigación. Los pesos a utilizar en la combinación pueden obtenerse auto-organizando los modelos. Además, se teoriza que la cooperación conseguirá que los métodos más eficaces en el corto plazo ayuden a los métodos en plazos mayores. Asimismo, se planea analizar el efecto de las condiciones iniciales a partir de las cuales los métodos de mediano plazo estiman las predicciones.

Otro punto importante aún no completamente explotado en los resultados alcanzados a la fecha es el uso de modelos basados en diferentes enfoques. Próximamente se añadirán modelos de Regresión de Soporte vectorial (SVR por sus siglas en inglés), y basados en redes neuronales recurrentes. Asimismo, está pendiente incluir la predicción basada en componentes estructurales (ver sección 2.5). Las predicciones de cada componente estructural se combinarán siguiendo una estrategia de redes modulares.

Capítulo 7

Conclusiones generales

En este documento se propone el desarrollo de un nuevo algoritmo de predicción a largo plazo, que funcione razonablemente con series de tiempo caóticas. Analizando el estado del arte se encontraron tres conceptos importantes a considerar en el diseño de este tipo de sistemas: las características de las series influyen de manera importante en la predicción, el tamaño del horizonte de predicción afecta en el desempeño del modelo y los mejores resultados los han presentado combinaciones de modelos.

El modelo propuesto utiliza conceptos de meta-aprendizaje, análisis de series de tiempo, teoría de ensambles, teoría de predicción y redes neuronales artificiales. En el documento se incluye una descripción detallada de los conceptos más importantes tomados de estas áreas y utilizados en la definición de algoritmo propuesto. Se definieron 3 hipótesis a probar a través del modelo a construirse, que son:

- Una combinación eficaz de modelos de predicción puede definirse a partir de las características de las series de tiempo involucradas, utilizando conocimiento previo.
- La eficacia de predicción puede mejorarse si se considera en la combinación de los modelos la división del horizonte a predecir, en diferentes ventanas.
- La manera de combinar modelos de predicción puede seguir una estrategia de auto-organización considerando reglas simples cómo competir y cooperar.

A la fecha se ha implementado un algoritmo que muestra de forma empírica la factibilidad de qué la segunda hipótesis sea correcta. Siguiendo la metodología descrita, se realizarán las extensiones necesarias para probar las otras dos hipótesis planteadas. Debido a que se va a comparar el resultado de esta investigación con el trabajo de Lemke y Gabrys (Lemke & Gabrys, 2010), actualmente se está trabajando en replicar su experimento, sin embargo la construcción del conjunto de meta-entrenamiento descrito por ellos no ha sido sencilla.

Los pasos pendientes a cubrir incluyen la definición de un extractor de meta-datos basado en envolturas con una estrategia de búsqueda completa, la construcción de una base de meta-aprendizaje considerando la estrategia de combinación propuesta y las utilizadas por (Lemke & Gabrys, 2010), el desarrollo de un método de meta-aprendizaje que considere los diferentes horizontes de predicción, la revisión de los modelos auto-organizados y la extensión el algoritmo de meta-aprendizaje con la estrategia apropiada de auto-organización.

Referencias

Abarbanel, H. D. & Gollub, J. P., 1996. Analysis of observed chaotic data. *Physics Today*, Volumen 49, p. 86.

ABS, 2005. *An Introductory Course on Time Series Analysis -- Electronic Delivery*. s.l.:Australian Bureau of Statistics.

Adeodato, P. y otros, 2008. *A systematic solution for the NN3 Forecasting Competition problem based on an ensemble of MLP neural networks*. s.l., s.n., pp. 1-4.

Alarcon-Aquino, V. & Barria, J., 2006. Multiresolution FIR neural-network-based learning algorithm applied to network traffic prediction. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(2), pp. 208-220.

Andrawis, R. R., Atiya, A. F. & El-Shishiny, H., 2011. Combination of long term and short term forecasts, with application to tourism demand forecasting. *International Journal of Forecasting*, 27(3), pp. 870-886.

Andrawis, R. R., Atiya, A. F. & El-Shishiny, H., 2011. Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. *International Journal of Forecasting*, 27(3), pp. 672-688.

Armstrong, J. S., 1985. *Long-range forecasting from crystal ball to computer*. second ed. s.l.:John Wiley & Sons.

Bates, J. M. & Granger, C. W. J., 1969. The Combination of Forecasts. *Operational Research Quarterly*, Dec, 20(4), pp. 451-468.

Beale, M. H., Hagan, M. T. & Demuth, H. B., 2012. *Neural Networks Toolbox User's Guide*. s.l.:MathWorks.

Bertelle, C., Duchamp, G. H. & Kadri-Dahmani, H., 2009. *Complex Systems and Self-organization Modelling*. s.l.:Springer-Verlag.

Borovkova, S. A., 1998. *Estimation and prediction for nonlinear time series*, s.l.: s.n.

Box, G. E. P., Jenkins, G. M. & Reinsel, G. C., 1994. *Time Series Analysis Forecasting and Control*. 3rd ed. s.l.:Prentice-Hall International.

Box, G. E. P. & Pierce, D. A., 1970. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 65(332), pp. 1509-1526.

Brassard, G. & Pa, 1996. *Fundamentos de Algoritmia*. s.l.:Prentice-Hall.

Campbell, L., 1882. *The Life of James Clerk Maxwell*. s.l.:Macmillan.

- Cao, L., 1997. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*, 110(1-2), pp. 43-50.
- Cao, Q., Ewing, B. T. & Thompson, M. A., 2012. Forecasting wind speed with recurrent neural networks. *European Journal of Operational Research*, 221(1), pp. 148-154.
- Cartwright, M. L., 1965. Jacques Hadamard. 1865-1963. *Biographical memoirs of fellows of the Royal Society*, Volumen 11, pp. 75-99.
- Casdagli, M., 1992. Chaos and Deterministic versus Stochastic Non-Linear Modelling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(2), pp. pp. 303-328.
- Castiello, C. & Fanelli, A. M., 2011. Computational Intelligence for Meta-Learning: A Promising Avenue of Research. En: N. Jankowski, W. Duch & K. Grabczewski, edits. *Meta-Learning in Computational Intelligence*. s.l.:Springer Berlin Heidelberg, pp. 157-177.
- Cortes, C. & Vapnik, V., 1995. Support-Vector Networks. *Machine Learning*, 20(3), pp. 273-297.
- Crone, S. F., 2008. *Time Series Forecasting Competition for Computational Intelligence*. [En línea] Available at: <http://www.neural-forecasting-competition.com/NN5/index.htm>
- Crone, S. F., Hibon, M. & Nikolopoulos, K., 2011. Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, 27(3), pp. 635-660.
- Crone, S. F., Nikolopoulos, K. & Hibon, M., 2008. *Automatic Modelling and Forecasting with Artificial Neural Networks—A forecasting competition evaluation*, s.l.: s.n.
- De-Gooijer, J. G. & Hyndman, R. J., 2006. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3), pp. 443-473.
- Desai, R. C. & Kapral, R., 2009. *Dynamics of Self-Organized and Self-Assembled Structures*. s.l.:Cambridge University Press.
- Deutsch, M., Granger, C. W. & Terasvirta, T., 1994. The combination of forecasts using changing weights. *International Journal of Forecasting*, 10(1), pp. 47-57.
- Dhanya, C. & Kumar, D. N., 2010. Nonlinear ensemble prediction of chaotic daily rainfall. *Advances in Water Resources*, 33(3), pp. 327-347.
- Dietterich, T., 2000. Ensemble Methods in Machine Learning. En: *Multiple Classifier Systems*. s.l.:Springer Berlin Heidelberg, pp. 1-15.
- Drucker, H. y otros, 1997. Support Vector Regression Machines. *Advances in Neural Information Processing Systems*, Issue 9, pp. 155-161.
- Dzeroski, S. & Zenko, B., 2004. Is Combining Classifiers with Stacking Better than Selecting the Best One?. *Machine Learning*, Volumen 54, pp. 255-273.
- Fonseca, R. & Gómez Gil, M. d. P., 2013. An assessment of ten-fold and Monte Carlo cross validations for time series forecasting. *To be published in the Proceedings of the 2013 10th International Conference on Electrical Engineering, Computing Science and Automatic Control*.

- Fraser, A. M. & Swinney, H. L., 1986. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, Feb, Volumen 33, pp. 1134-1140.
- García Pedrero, Á. M. & Gómez Gil, M. D. P., 2009. *Arquitectura Neuronal Apoyada en Señales Reconstruidas con Wavelets para Predicción de Series de Tiempo Caóticas*, Santa María Tonantzintla: s.n.
- García-Pedrajas, N., Hervas-Martinez, C. & Ortiz-Boyer, D., 2005. Cooperative coevolution of artificial neural network ensembles for pattern classification. *Evolutionary Computation, IEEE Transactions on*, June, 9(3), pp. 271-302.
- Gómez Gil, M. d. P., 2010. *Notas del curso de Redes Neuronales Avanzadas*. [En línea] Available at: <http://ccc.inaoep.mx/~pgomez>
- Gómez-Gil, P., 1998. *The Effect of Non-linear Dynamic Invariants in Recurrent Neural Networks for Prediction of Electrocardiograms*, s.l.: s.n.
- Grassberger, P. & Procaccia, I., 1983. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1), pp. 189-208.
- Grigorenko, I., 2006. *Optimal Control and Forecasting of Complex Dynamical Systems*. s.l.:World Scientific.
- Haken, H., 2006. *Information and Self-Organization: A Macroscopic Approach to Complex Systems*. s.l.:Springer.
- Haykin, S., 1999. *Neural Networks A Comprehensive Foundation*. Second Edition ed. s.l.:Pearson Prentice Hall.
- Haykin, S., 2005. *Neural Networks*. Second ed. Delhi, India: Pearson Prentice Hall.
- Helbing, D., 2012. *Social Self-Organization Agent-Based Simulations and Experiments to Study Emergent Social Behavior*. s.l.:Springer-Verlag.
- Hibon, M. & Evgeniou, T., 2005. To combine or not to combine: selecting among forecasts and their combinations. *International Journal of Forecasting*, 21(1), pp. 15-24.
- Hudson Beale, M., Hagan, M. T. & Demuth, H. B., 2012. *Neural Network Toolbox User's Guide R2012b*. s.l.:MathWorks.
- Islam, M., Yao, X. & Murase, K., 2003. A constructive algorithm for training cooperative neural network ensembles. *Neural Networks, IEEE Transactions on*, July, 14(4), pp. 820-834.
- Kanji, G. K., 2006. *100 Statistical Tests*. Third ed. s.l.:Sage Publications.
- Kantz, H., 1994. A robust method to estimate the maximal Lyapunov exponent of a time series. *Physics Letters A*, 185(1), pp. 77-87.
- Kantz, H. & Schreiber, T., 2003. *Nonlinear Time Series Analysis*. s.l.:Cambridge University Press.
- Kaplan, D. & Glass, L., 1995. *Understanding nonlinear dynamics*. s.l.:Springer-Verlag New York.
- Kennel, M. B., Brown, R. & Abarbanel, H. D. I., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, Mar, Volumen 45, pp. 3403-3411.

- Kohavi, R. & John, G. H., 1997. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), pp. 273-324.
- Kohonen, T., 1990. The self-organizing map. *Proceedings of the IEEE*, 78(9), pp. 1464-1480.
- Kordik, P. & Cerny, J., 2011. Self-organization of Supervised Models. En: N. Jankowski, W. Duch & K. Grabczewski, edits. *Meta-Learning in Computational Intelligence*. s.l.:Springer Berlin Heidelberg, pp. 179-223.
- Koryakin, D., Lohmann, J. & Butz, M. V., 2012. Balanced echo state networks. *Neural Networks*, Volumen 36, pp. 35-45.
- Kuncheva, L. I., 2004. *Combining Pattern Classifiers Methods and Algorithms*. s.l.:John Wiley & Sons.
- Kuncheva, L. & Whitaker, C., 2003. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning*, 51(2), pp. 181-207.
- Lei, M., Wang, Z. & Feng, Z., 2002. A method of embedding dimension estimation based on symplectic geometry. *Physics Letters A*, 303(2-3), pp. 179-189.
- Lemke, C. & Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10-12), pp. 2006-2016.
- Leontaritis, I. J. & Billings, S. A., 1985. Input-output parametric models for non-linear systems Part II: stochastic non-linear systems. *International Journal of Control*, 41(2), pp. 329-344.
- Lorenz, E. N., 1963. Deterministic Nonperiodic Flow. *J. Atmos. Sci.*, mar, 20(2), pp. 130-141.
- Mackey, M. C. & Glass, L., 1977. Oscillation and chaos in physiological control systems. *Science*, 197(4300), pp. 287-289.
- Makridakis, S. & Hibon, M., 2000. The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4), pp. 451-476.
- Matijas, M., Suykens, J. A. & Krajcar, S., 2013. Load forecasting using a multivariate meta-learning system. *Expert Systems with Applications*, pp. -.
- May, R. M., 1976. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560), pp. 459-467.
- Mees, A. I., 2001. *Nonlinear Dynamics and Statistics*. s.l.:Birkhauser Boston.
- Melin, P., Mancilla, A., Lopez, M. & Mendoza, O., 2007. A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. *Applied Soft Computing*, 7(4), pp. 1217-1226.
- Merz, C. J., 1999. Using correspondence analysis to combine classifiers. *Machine Learning*, 36(1-2), pp. 33-58.
- Packard, N. H., Crutchfield, J. P., Farmer, J. D. & Shaw, R. S., 1980. Geometry from a Time Series. *Phys. Rev. Lett.*, Sep, Volumen 45, pp. 712-716.
- Pavlidis, N., Tasoulis, D. & Vrahatis, M., 2003. *Financial forecasting through unsupervised clustering and evolutionary trained neural networks*. s.l., s.n., pp. 2314 - 2321 Vol.4.

- Rocha, L. M., 2013. *Biologically Inspired Computing*. s.l.:s.n.
- Ronco, E. & Gawthrop, P., 1995. Modular neural networks: a state of the art. *Rapport Technique csc-95026*.
- Ruelle, D. & Takens, F., 1971. On the nature of turbulence. *Communications in Mathematical Physics*, 23(4), pp. 343-344.
- Sapankevych, N. & Sankar, R., 2009. Time Series Prediction Using Support Vector Machines: A Survey. *Computational Intelligence Magazine, IEEE*, may , 4(2), pp. 24-38.
- Sharkey, A., 1999. Multi-Net Systems. En: A. Sharkey, ed. *Combining Artificial Neural Nets*. s.l.:Springer London, pp. 1-30.
- Strauss, D. G. y otros, 2011. An ECG index of myocardial scar enhances prediction of defibrillator shocks: An analysis of the Sudden Cardiac Death in Heart Failure Trial. *Heart Rhythm*, 8(1), pp. 38-45.
- Taieb, S. B., Bontempi, G., Atiya, A. F. & Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8), pp. 7067-7083.
- Takens, F., 1981. Detecting strange attractors in turbulence. En: D. Rand & L. Young, edits. *Dynamical Systems and Turbulence, Warwick 1980*. s.l.:Springer Berlin Heidelberg, pp. 366-381.
- Ting, K. M. & Witten, I. H., 1999. Issues in Stacked Generalization. *Journal of Artificial Intelligence Research*, abs/1105.5466(10), pp. 271-289.
- Trewin, D., 2003. *A Guide to Interpreting Time Series - Monitoring Trends*. s.l.:Australian Bureau of Statistics.
- Wang, X., Smith-Miles, K. & Hyndman, R., 2009. Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing*, 72(10-12), pp. 2581-2594.
- Wolpert, D. H., 1992. Stacked generalization. *Neural Networks* , 5(2), pp. 241-259.
- Wong, W.-K., Bai, E. & Chu, A., 2010. Adaptive Time-Variant Models for Fuzzy-Time-Series Forecasting. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, dec. , 40(6), pp. 1531-1542.
- Wu, Z. & Huang, N. E., 2004. A study of the characteristics of white noise using the empirical mode decomposition method. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2046), pp. 1597-1611.
- Yang, Z., Lu, W. & Harrison, R., 2001. Evolving Stacked Time Series Predictors with Multiple Window Scales and Sampling Gaps. *Neural Processing Letters*, 13(3), pp. 203-211.
- Zemouri, R., Gouriveau, R. & Zerhouni, N., 2010. Defining and applying prediction performance metrics on a recurrent NARX time series model. *Neurocomputing*, 73(13-15), pp. 2506-2521.

Anexos

Anexo I. Sistemas dinámicos

Según Dhanya (Dhanya & Kumar, 2010) basado en los trabajos (Lorenz, 1963) y (Ruelle & Takens, 1971), un sistema dinámico es cualquier sistema que evoluciona en el tiempo a partir de algún estado inicial conocido y puede ser descrito por un conjunto de ecuaciones. Este es usualmente descrito en términos de trayectorias en el *espacio de estados* (que es un espacio abstracto construido matemáticamente en el cual cada dimensión representa una variable del sistema). Dado que el conjunto de ecuaciones no es conocido previamente en el análisis de la serie de tiempo, el espacio de estados es representado por *espacio de fase* que puede ser reconstruido a partir de la misma serie de tiempo. Cada *trayectoria* en el espacio de fase representa la evolución del sistema a partir de una condición inicial. Un *atractor* puede ser definido como un subconjunto de trayectorias de entre todas las trayectorias, originados de diferentes condiciones iniciales, eventualmente convergerán. Donde este subconjunto de trayectorias atrae a todas las otras trayectorias en el espacio de fase, éste es denominado atractor del sistema. Para una serie de tiempo de un sistema regular, el atractor tendrá una dimensión entera. Pero para un sistema caótico, que es irregular y es sensible a las condiciones iniciales, el atractor puede ser caracterizado por una dimensión no-entera por lo que es denominado *atractor extraño*.

Dado que la dinámica de una serie de tiempo caótica no es conocida, al igual que atractor teórico original, el espacio de estados de una serie de tiempo escalar es aproximado al espacio de fase donde el atractor es reconstruido utilizando la serie escalar. Así, la reconstrucción del espacio de fase provee una simplificada representación de varias dimensiones de una serie de tiempo no-lineal de una sola dimensión, la reconstrucción del espacio de fase es el mapa de retorno. Packard y colaboradores (Packard, et al., 1980) proponen un método de retardos para reconstruir el espacio de fase, introducido y demostrador por Takens (Takens, 1981). De acuerdo a este enfoque, para una serie de tiempo escalar x_i donde $i = 1, 2, \dots, N$, las dinámicas pueden estar completamente embebidas en un espacio de fase de m -dimensiones representado por el vector,

$$y_j = (x_j, x_{j+\tau}, x_{j+2\tau}, \dots, x_{j+(m-1)\tau}) \quad (\text{AI. 1})$$

donde $j = 1, 2, \dots, N - (m - 1) \tau / \Delta t$; la variable m se denomina dimensión embebida ($m \geq d$, donde d es la dimensión del atractor); el retraso es τ y el tiempo de muestreo es Δt . La dimensión

m puede ser considerada como el mínimo número de variables de estado requeridas para describir el sistema. Los métodos más populares para estimar la dimensión embebida son Grassberger-Procaccia (GPA) (Grassberger & Procaccia, 1983) y el método de los falsos vecinos más cercanos (FNN por sus siglas en inglés) (Kennel, et al., 1992).

El tiempo de retardo τ es la longitud promedio de memoria del sistema. Un tiempo de retraso debe escogerse apropiadamente para la mejor representación del espacio de fase. Si τ es muy pequeño, las coordenadas del espacio de fase no serán independientes, resultando en pérdida de información de las características de la estructura del atractor. Por el otro lado, si τ es muy grande, entonces podría haber una correlación no dinámica entre los vectores de estado dado que las trayectorias de los vecinos divergen, resultando en pérdida de información del sistema original. El mejor valor de τ , usualmente, se determina utilizando tanto la función de auto-correlación como el método de información mutua (Fraser & Swinney, 1986).

Anexo II. Características utilizadas en el análisis de series de tiempo

Las tablas a continuación muestran las características seleccionadas para el análisis de series de tiempo realizado de los trabajos de (Wang, et al., 2009) y (Lemke & Gabrys, 2010). Estas características serán utilizadas para desarrollar los modelos propuestos en esta investigación. La tabla II.1 contiene características que son obtenidas a través de análisis estadísticos realizados sobre la serie de tiempo en cuestión; la tabla II.2 contiene características particulares a modelos de auto-correlación

Tabla II.1. Lista de características estadísticas

No.	Abreviación	Característica
1	Std	Desviación estándar de la serie sin tendencia.
2	skew	Oblicuidad de la serie de tiempo
3	kurt	Kurtosis de la serie de tiempo
4	length	Longitud de la serie
5	trend	Desviación estándar de la serie / Desviación estándar de la serie sin tendencia
6	turn	Número de puntos de cambio
7	step	Promedio de pasos de cambio
8	lyap	Mayor exponente de Lyapunov
9	Iqr	Rango <i>interquartile</i>
10	Mean	Promedio
11	Median	Mediana
12	Mode	Moda

Tabla II. 2. Lista de características de auto-correlación

No.	Abreviación	Característica
13	acf[1,2]	Auto correlación de uno o dos retrasos
14	pacf[1,2]	Auto correlación parcial de uno o dos retrasos
15	Season	Medida de Estacionalidad, pacf[7] diaria, pacf[12] mensual