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

Por

Rigoberto Salomón Fonseca Delgado

Tesis sometida como requisito parcial para obtener el grado de

DOCTOR EN CIENCIAS EN LA ESPECIALIDAD DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Sta. Ma. Tonantzintla, Puebla
Febrero 2017

Supervisada por:
Dra. María del Pilar Gómez Gil
Investigadora del INAOE

©INAOE 2017
Derechos Reservados
El autor otorga al INAOE el permiso de reproducir y distribuir copias de esta tesis en su totalidad o en partes

A mi familia

Agradecimientos

A CONACYT por el apoyo otorgado a través de la Beca para estudios de Doctorado número 344837. Este trabajo fue parcialmente apoyado por el Proyecto CONACYT CB-2010-155250.

Un agradecimiento muy especial a mi Asesora la Dra. María del Pilar Gómez Gil por su guía, esfuerzo y dedicación en la satisfactoria conclusión de este proyecto de investigación.

Agradezco a mis sinodales por el tiempo dedicado y las valiosas recomendaciones que ayudaron a mejorar el trabajo a lo largo de todo el proceso.

Un agradecimiento a todos quienes conforman el CRECTEALC, en especial al Dr. Sergio Camacho por brindarme la oportunidad de empezar un nuevo camino lleno de oportunidades.

Un afectuoso agradecimiento a mi esposa Liliana, mi familia y amigos por el apoyo y comprensión.

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, gobierno, etc. En los últimos años, varias investigaciones han puesto su atención en reutilizar conocimiento previo a fin de combinar modelos y mejorar la predicción. El proceso de aprender a partir de resultados previos es denominado meta-aprendizaje. En el contexto de combinación de modelos, 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: analizar el comportamiento de los modelos con los datos, buscar modelos que puedan combinarse adecuadamente 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 se centra en el problema de meta-aprendizaje de combinación de modelos, utilizando auto-organización y un análisis automático del comportamiento de los modelos. La meta es mejorar la exactitud en predicción a largo plazo alcanzada por los algoritmos del estado del arte en series de tiempo caóticas. La principal contribución de este trabajo es un nuevo método, basado en meta-características y en las redes neuronales de mapas auto-organizados, capaz de generar combinaciones de modelos de predicción considerando la exactitud de los modelos y su diversidad de comportamiento en el tiempo. El método desarrollado fue comparado con otros métodos del estado del arte, y validado utilizando series sintéticas y series obtenidas de aplicaciones reales, que presentan comportamientos tanto caóticos como no caóticos.

Palabras Clave: *Combinación de modelos, Meta-aprendizaje, Mapas Auto-Organizados, Predicción a largo plazo de series de tiempo, Series de tiempo caóticas*.

Abstract

Long-term prediction of time series from chaotic systems is a difficult task but required in various fields such as economics, medicine, government, etc. In recent years, several studies have turned their attention to reuse prior knowledge with the aim of combining models and improving prediction. The process of learning from previous results is called meta-learning. On the context of model combination, the meta-learning strategy lets to automatically generate an expert, able to recommend an appropriate combination of models to predict a time series in particular. However, the process of meta-learning in time series imposes nontrivial challenges such as: analyzing model behavior with time series data, look for models that can be combined with each series and even generate new methods of meta-learning those consider variations of model performance over time. This research focuses on the problem of meta-learning of model combination, using self-organization and an automatic analysis of the behavior of the models. The goal is to improve the accuracy on long-term prediction achieved by state of the art algorithms in chaotic time series. The main contribution of this work is a new method, based on meta-features and self-organized maps neural networks, capable of generating combinations of prediction models considering the accuracy of the models and their diversity of behavior over time. The developed method was compared with other methods of the state of the art, and validated using synthetic series and series obtained from real applications, that have a chaotic and non-chaotic behavior.

Keywords: *Combination of models, Meta-learning, Self-Organizing Maps, Long-term prediction of time series, Chaotic time series.*

Tabla de contenido

Capítulo 1. Introducción 1

1.1 Problemática 3

1.2 Pregunta de Investigación 3

1.3 Hipótesis 4

1.4 Objetivo General 4

1.5 Objetivos Específicos 4

1.6 Resultados Principales 5

1.6.1 Combinación Basada en Diferentes Ventanas de Predicción 5

1.6.2 Combinación Basada en Auto-Organización 6

1.7 Aportaciones 7

1.8 Organización del Documento 9

Capítulo 2. Conceptos Preliminares 11

2.1 Predicción de Series de Tiempo a Largo Plazo 11

2.2 Revisión de Medidas de Error 12

2.2.1 Medidas Dependientes de la Escala 13

2.2.2 Medidas Basadas en el Porcentaje de Error 14

2.2.3 Medidas Basadas en Errores Relativos 16

2.2.4 Medidas Relativas 17

2.2.5 Errores Escalados 18

2.3 Caos y Auto-Organización 19
2.4 Mapas Auto-Organizados SOM 20
2.4.1 Competir 21
2.4.2 Cooperar 22
2.4.3 Arquitectura de la Red SOM 23
2.4.4 Establecer el Número de Neuronas 23
2.5 Meta-Aprendizaje 24
2.6 Ensambles 25
2.6.1 Fuentes de Diversidad 27
2.7 Resumen del Marco Teórico 28
Capítulo 3. Trabajo Relacionado 31
3.1 Prediciendo Sistemas Caóticos 31
3.2 Ensambles Basados en Redes SOM 33
3.3 Ensambles Basados en Meta-Aprendizaje 34
3.3.1 Método de Lemke y Gabrys 35
3.4 Otros Trabajos 36
3.4.1 Redes Modulares 37
3.4.2 Redes Difusas 39
3.5 Resumen de los Trabajos Relacionados 40
Capítulo 4. Métodos Propuestos 41
4.1 Diagrama General del Esquema de Predicción 42
4.2 Método Extractor de Meta-Datos 43
4.3 Combinación Temporal Validada 46
4.4 Error Representativo 48
4.5 Selección de Modelos Utilizando Redes SOM 50

4.5.1 Construcción de Mapa Auto-Organizado SOM 52
4.5.2 Selección de Modelos 53
4.5.3 Ajustar Selección de Modelos 54
4.5.4 Combinación y Predicción Final 55
4.5.5 Algoritmo Completo y Análisis de Complejidad 55
4.6 Resumen de Métodos Propuestos 59
Capítulo 5. Análisis Experimental 61
5.1 Descripción de los Conjuntos de Datos 61
5.1.1 ¿Cuáles Series son Caóticas? 64
5.2 Descripción del Conjunto de Modelos 65
5.3 Estrategias de comparación 66
5.3.1 Mejor Modelo 67
5.3.2 Mejor Modelo por Base 67
5.3.3 Método de Lemke 67
5.4 Análisis de Eficacia de los Métodos Propuestos 68
5.4.1 Evaluación en Series Consideradas Caóticas 68
5.4.2 Evaluación con Series de la Competencia NN5 71
5.4.3 Análisis de la Eficacia de TVC 73
5.4.4 Recomendaciones para el uso de TVC 76
5.4.5 Análisis de la Eficacia de SOMSM 80
5.4.6 Desempeño de SOMSM con la base de datos NN5 82
5.4.7 Casos extremos de Predicciones con SOMSM 88
5.5 Evaluación con Diferentes Métricas 89
Capítulo 6. Conclusiones 95
6.1 Conclusiones 96

6.2 Aportaciones 97
6.3 Trabajo Futuro 98
Referencias 101
Anexo A. Acrónimos 113
Anexo B. Lista de Símbolos 114
Anexo C. Predicción a Largo Plazo 117
Anexo D. Ejemplos de Series de Tiempo Caóticas 119
Anexo E. El Error de Predicción Crece Exponencialmente 121
Anexo F. Tipos de Modelos de Predicción 123
F.1 Modelos Auto-Regresivos Integrados de Medias Móviles 123
F.2 Redes Neuronales NARX 124
F.3 Trayectorias Más Cercanas 125
Anexo G. Características Del Método de Lemke 127
Anexo H. Técnicas de Re-Muestreo para la Estimación del Error 129
H.1 *Hold-out* CV 130
H.2 *k*-pliegues CV 130
H.3 Dejar Uno Fuera CV 130
H.4 Monte Carlo CV 131
Anexo I. Impacto de la Métrica en la Selección de Modelos en TVC 132

Índice de figuras

Figura 2.1. Estructura hexagonal de una red SOM de 5 filas y 5 columnas. 20

Figura 3.1. Ejemplo reconstrucción del “atractor” de Lorenz. 32

Figura 4.1. Proceso propuesto de construcción de un ensamble para la predicción de una serie de tiempo particular. 43

Figura 4.2. Diseño del extractor de meta-características. 46

Figura 4.3. Combinación Temporal Validada de los modelos de predicción. 47

Figura 4.4. Cálculo del error representativo para horizontes mayores a uno 49

Figura 4.5. Ejemplo de agrupamiento de modelos, utilizando el error representativo con un horizonte igual a 2. 50

Figura 4.6. Método propuesto SOMSM 51

Figura 4.7. Ejemplo de agrupamiento de modelos por su error representativo utilizando una red SOM. 53

Figura 4.8. Algoritmo de selección de modelos utilizando un mapa SOM. 56

Figura 5.1. Resultados sobre las series cuyo máximo exponente de Lyapunov es mayor a cero. 70

Figura 5.2. Caja con bigotes y promedio de SMAPEs del método TVC y otros métodos... 75

Figura 5.3. Análisis de la serie con id =29. 78

Figura 5.4. Predicción del método TVC sobre la serie Id = 29, comparado con la serie de prueba y los otros métodos. 79

Figura 5.5. Resultados obtenidos con las series en las que se recomienda utilizar TVC. 79

Figura 5.6. Caja con bigotes y promedio de SMAPEs del método SOMSM y los otros métodos. 81

Figura 5.7. Resultados del método SOMSM y los otros métodos con las series del conjunto NN5. 83

Figura 5.8. Serie Id 20, correspondiente a la 9 serie del conjunto NN5. 85

Figura 5.9. Histograma y mapa de retorno de la serie Id=20, 9na del conjunto NN5. 86

Figura 5.10. Mapa auto-organizado de los modelos de predicción en función de su error representativo con la serie Id 20, SMAPE validado con MCCV. 87

Figura 5.11. Predicción de SOMSM y valor esperado con la serie Id. 20, 9na del conjunto NN5. 87

Figura 5.12. Histograma y mapa de retorno de la serie caótica Mackey-Glass. 88

Figura 5.13. Predicción de SOMSM y valor esperado con la serie caótica Mackey-Glass. 88

Figura 5.14. Histograma y mapa de retorno de la serie sintética generada con un modelo ARMA definido. 89

Figura 5.15. Predicción de SOMSM y valor esperado con la serie sintética generada con un modelo ARMA definido. 89

Índice de tablas

Tabla 2.1. Medidas de predicción dependientes de la escala. 13

Tabla 2.2. Medidas basadas en el porcentaje de error. 14

Tabla 2.3. Medidas “simétricas” basadas en el porcentaje de error. 15

Tabla 2.4. Medidas basadas en errores relativos 16

Tabla 5.1. Series de tiempo de competencias de predicción internacionales. 62

Tabla 5.2. Series de tiempo sintéticas..... 62

Tabla 5.3. Series de tiempo sintéticas (continuación). 63

Tabla 5.4. Series de tiempo de diversas aplicaciones. 63

Tabla 5.5. Estimación del máximo exponente de Lyapunov (ME) de cada serie. 65

Tabla 5.6. Modelos candidatos para la construcción del ensamble. 66

Tabla 5.7. Posibles valores para los parámetros de los métodos TVC y SOMSM. 68

Tabla 5.8. SMAPE de los métodos propuestos sobre las series consideradas caóticas. 69

Tabla 5.9. Resultados de los métodos propuestos con las series cuyo máximo exponente de Lyapunov es mayor a cero. 70

Tabla 5.10. Prueba significancia estadística Wilcoxon comparando SOMSM-mediana con los otros modelos, con las series de comportamiento caótico. 71

Tabla 5.11. Resultados promedio de TVC, SOMSM y otros modelos sobre las series del torneo NN5, evaluación con métricas: RMSE, SMAPE, GMRAE, RelMAE, MASE. 72

Tabla 5.12. Resultados del torneo NN5 con el conjunto reducido. 73

Tabla 5.13. Resultados de TVC con todas las series de tiempo. 75

Tabla 5.14. Prueba significancia estadística Wilcoxon comparando TVC-mediana con los otros modelos. 76

Tabla 5.15. Resultados de TVC cuando la regla propuso su uso. 80

Tabla 5.16. Prueba significancia estadística Wilcoxon comparando TVC-mediana con los otros modelos, siguiendo la recomendación de uso de TVC. 80

Tabla 5.17. Resultados de SOMSM con todas las series de tiempo. 82

Tabla 5.18. Prueba significancia estadística Wilcoxon comparando SOMSM-mediana con los otros modelos, utilizando todas las series. 82

Tabla 5.19. Resultados de SOMSM sobre las series del conjunto NN5. 83

Tabla 5.20. Prueba significancia estadística Wilcoxon comparando SOMSM-mediana con los otros modelos, con las series del conjunto NN5. 84

Tabla 5.21. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con RMSE, comparado con otros métodos. 90

Tabla 5.22. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con SMAPE, comparado con otros métodos. 90

Tabla 5.23. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con GMRAE, comparado con otros métodos. 91

Tabla 5.24. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con RelMAE, comparado con otros métodos. 92

Tabla 5.25. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con MASE, comparado con otros métodos. 92

Capítulo 1. Introducción

A pesar de los esfuerzos por mejorar la eficacia en la predicción, el problema de predecir múltiples pasos en series de tiempo, partiendo únicamente de 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 como el caso de (Wang, et al., 2009), y suavizado exponencial como (Hyndman & Billah, 2003). Otros algoritmos más 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 (Yu, et al., 2015), lógica difusa (Miranian & Abdollahzade, 2013), redes neuronales (Crone, et al., 2011) y máquinas de soporte de vectores (Sapankevych & Sankar, 2009), entre otros. Asimismo, los modelos de predicción han sido utilizados solos o en combinación con varias estrategias(Makridakis & Hibon, 2000).

Las series de tiempo provenientes de sistemas sensibles a cambios en las condiciones iniciales, donde la relación entre los valores futuros y pasados es una función no-lineal son conocidas como caóticas; las que están catalogadas como no predecibles a largo plazo, debido al impacto de la sensibilidad a las condiciones iniciales en el cálculo de valores futuros (Kantz & Schreiber, 2003). A pesar de ello, muchas aplicaciones que manejan sistemas caóticos requieren una estimación razonable de los valores futuros; ejemplos de este tipo de series pueden encontrarse en la demanda de carga eléctrica (Matijas, et al., 2013), en retiros bancarios (Andrawis, et al., 2011) y en la velocidad del viento en la superficie terrestre (Cao, et al., 2012), entre otros. Por esta razón, varias investigaciones buscan mejorar la exactitud de los modelos de predicción aplicados en diferentes ámbitos.

Por otra parte, se sabe que la combinación de diferentes modelos de predicción es una estrategia que puede obtener resultados mejores que los alcanzados por modelos de predicción aplicados de forma individual (Bates & Granger, 1969). Sin embargo, seleccionar modelos que puedan cooperar y combinarse para obtener un buen resultado, no es un problema trivial (Kuncheva & Whitaker, 2003); por ejemplo, si se combinan dos modelos que predicen el mismo valor equivale a solo utilizar uno de esos modelos y por otro lado, si un modelo devuelve predicciones correctas para cada instante de tiempo y se lo combina con un modelo diferente que siempre devuelve respuestas incorrectas el desempeño disminuye en lugar de mejorar. Por lo tanto, al seleccionar modelos a combinar se busca incrementar la diferencia en las predicciones de los modelos, diversidad de comportamiento, pero teniendo presente el desempeño de cada modelo (Ren, et al., 2016).

Cuando la selección de un conjunto de modelos a combinar se realiza por expertos humanos, el proceso inicia revisando los datos. Posteriormente, 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), (Adhikari & Verma, 2016), (Adhikari, 2015), (Ouyang & Yin, 2014). De éstos, el reportado por Lemke y Gabrys (2010) es de particular interés, debido a que los autores reportan que 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 analiza cada serie de tiempo a través de un conjunto de características de la serie y se crea una asociación entre las características de cada serie con el desempeño estimado de los modelos de predicción; esta relación representa un conocimiento y es denominado meta-característica. 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*, o "*experto*". Dada una tarea de predicción, este experto es capaz de determinar una combinación eficaz de modelos capaces de predecir esta nueva serie. En esta investigación, se explora esta estrategia de predicción y se proponen nuevos algoritmos para mejorar la eficiencia de la predicción.

1.1 Problemática

Las series de tiempo provenientes de sistemas complejos son difíciles de predecir dado que exhiben cambios bruscos de comportamiento; por lo que un solo modelo tiene problemas en estimar todo el horizonte de predicción. Los sistemas complejos también pueden verse como sistemas deterministas afectados por sistemas desconocidos, donde un solo modelo no puede identificar las particularidades de cada uno de los sistemas desconocidos. Conjuntos de modelos combinados, es decir ensambles, pueden retornar mejores resultados que un solo modelo; sin embargo, identificar que modelos deben utilizarse es uno de los problemas principales en la construcción de ensambles. Utilizar diferentes ventanas de predicción puede mejorar la predicción, atacando el problema de los cambios bruscos de comportamiento de la serie de tiempo. Permitir que diferentes modelos capturen diferentes características de la serie de tiempo ataca el problema de los sistemas desconocidos, la selección de los modelos puede guiarse permitiendo que los componentes del ensamble se organicen automáticamente, motivar la auto-organización, para adaptar la selección según el comportamiento de los modelos con la serie de tiempo. 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 métodos existentes pueden mejorarse, diseñando expertos que consideren auto-organización y la inclusión de un manejo de división del horizonte de predicción en diferentes ventanas.

1.2 Pregunta de Investigación

En este proyecto se desea responder la siguiente pregunta:

¿El promover la auto-organización de modelos, considerando diferentes ventanas de predicción, puede usarse para construir ensambles que superen los resultados del modelo con mejor desempeño de los modelos disponibles?

En este documento nos referimos con “modelos disponibles”, a aquellos que pueden ser aplicables a la predicción de la serie de tiempo involucrada, y que dicha solución es

posible de implementarse y ejecutarse en un tiempo razonable, considerando el contexto en que se realiza la predicción.

1.3 Hipótesis

Una combinación eficaz de modelos de predicción puede auto-organizarse a partir de meta-características, para obtener resultados estadísticamente mejores que el modelo con mejor desempeño de los modelos disponibles para la construcción del ensamble.

En particular en esta investigación se utilizan tres tipos de modelos: modelos de medias móviles iteradas auto-regresivas (ARIMA por sus siglas en inglés) (Box & Pierce, 1970), redes neuronales no-lineales auto-regresivas con entradas externas (NARX por sus siglas en inglés) (Leontaritis & Billings, 1985)(Beale, et al., 2015) y kTrayectorias más cercanas (McNames, 1998).

1.4 Objetivo General

Desarrollar un método de meta-aprendizaje capaz de seleccionar un conjunto de modelos y combinarlos, de manera auto-organizada, considerando diferentes ventanas de predicción en series de tiempo caóticas para construir un ensamble cuyo desempeño supere al modelo con mejor desempeño de los modelos disponibles.

1.5 Objetivos Específicos

El objetivo general puede lograrse al alcanzar los siguientes objetivos específicos:

- Establecer un método de selección de los mejores modelos en diferentes ventanas de predicción, para mejorar la eficacia en la predicción a largo plazo con respecto a otros modelos existentes.

- Construir un algoritmo de meta-aprendizaje capaz de extraer características de las series de tiempo y del comportamiento de los modelos de predicción, para mejorar la combinación de los modelos.

• 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.

1.6 Resultados Principales

En este trabajo se presentan dos métodos para la selección y combinación de modelos de predicción a largo plazo que se basan en el diferente comportamiento de los modelos en el tiempo y buscan un balance entre la diversidad y eficacia de los modelos seleccionados. El resto de esta sección describe brevemente los métodos desarrollados, los mismos que son explicados con detalle en capítulos subsecuentes. Es importante resaltar que, a diferencia de otros métodos reportados en la literatura, los métodos propuestos son totalmente independientes de los modelos base y pueden aplicarse con cualquier problema de predicción a largo plazo de series de tiempo.

1.6.1 Combinación Basada en Diferentes Ventanas de Predicción

Varios autores han buscado determinar los mejores métodos 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),(Bates & Granger, 1969).
- 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).

A partir de estas observaciones, se diseñó un método de combinación que selecciona modelos considerando diferentes ventanas de predicción, denominado "*Combinación Temporal Validada*" (TVC por sus siglas en inglés), el cual cumple con el primer objetivo específico de este proyecto de investigación. El método inicia estableciendo la exactitud de los modelos en cada ventana, utilizando la métrica "Promedio Simétrico del Porcentaje de

Error Absoluto" (SMAPE por sus siglas en inglés) y una validación cruzada Monte Carlo; a continuación, se seleccionan los modelos con la menor SMAPE en cada ventana de predicción. Los modelos de cada ventana son combinados usando la función mediana; la predicción resultante es la concatenación de las secuencias predichas por cada ventana de predicción. Este método fue comparado con la selección de los modelos de menor SMAPE sin considerar las ventanas de predicción, la evaluación utilizó series sintéticas y tomadas de aplicaciones reales.

1.6.2 Combinación Basada en Auto-Organización

Dos aspectos han aportado evidencia de ser muy útiles en el diseño de ensambles de modelos de predicción: el uso de meta-características para seleccionar los posibles modelos (Matijas, et al., 2013) (Lemke & Gabrys, 2010) y la identificación de un adecuado balance entre la diversidad de comportamiento y la exactitud de los modelos involucrados (Kuncheva & Whitaker, 2003) (Chandra & Yao, 2006). Por otro lado, el comportamiento de los sistemas caóticos puede analizarse a través de las interacciones entre sus componentes (Helbing, 2012). Entonces, un posible enfoque para emular sistemas complejos, y por lo tanto construir modelos capaces de predecir su comportamiento, puede basarse en un diseño adecuado de interacciones entre componentes, que permita la auto-organización y el auto-control de éstos. Además, dicho modelo debería de proveer suficiente libertad de adaptación a los cambios propios del sistema, además de ser capaz de delimitar su comportamiento. Una manera de construir dicho sistema de componentes que se auto-organizan, es a través del uso de las redes neuronales auto-organizadas (conocidas como SOM por sus siglas en inglés), las cuales siguen reglas simples de competencia y cooperación y fueron propuestas por T. Kohonen (Kohonen, 1982).

Basándose en estos conceptos, se diseñó un método capaz de seleccionar y combinar modelos utilizando una red SOM conjuntamente con meta-características obtenidas del desempeño de los modelos; una de ellas es el vector de error representativo que representa el error del modelo en cada paso de predicción. Este método, denominado "*Self Organizing Maps for Selecting Models*" (SOMSM) cumple con los objetivos específicos segundo y tercero de este proyecto. Una vez calculadas las meta-características, el método genera de

manera no supervisada un mapa de dos dimensiones de éstas, para facilitar la comparación entre modelos, donde cada neurona representa un grupo de modelos con comportamiento similar. Las neuronas lejanas en el mapa contienen modelos cuyo comportamiento es más diferente que el de los modelos en neuronas cercanas. Este mapa permite seleccionar los modelos con el mejor balance entre exactitud y diversidad, lo cual es logrado a través del uso de un umbral de tolerancia de error de predicción y controlando la distancia entre las neuronas utilizadas para la selección. Aquí encontramos que una máxima diversidad, es decir distancia entre neuronas en el mapa, está relacionada con una disminución considerable de la exactitud de los modelos y por lo tanto combinar modelos con máxima diversidad disminuye el desempeño del ensamble. También encontramos que, si se controla la disminución de exactitud debido a la inclusión de diversidad entre los modelos seleccionados, se mejora el desempeño del ensamble comparado con el resultado que se obtiene si solo se considera exactitud.

1.7 Aportaciones

Los sistemas complejos pueden ser afectados por diferentes sistemas desconocidos; dado que no se conoce cuantos sistemas desconocidos existen, proponemos combinar diferentes modelos de predicción con la esperanza de que cada modelo aprenda características diferentes a partir de los datos de la series. En ese sentido, la principal aportación al conocimiento de esta investigación es la creación y análisis de un nuevo método de predicción, llamado SOMSM, que construye ensambles y es capaz de controlar automáticamente la diversidad y la exactitud de los modelos a combinar. SOMSM utiliza una red neuronal auto-organizada y meta-características, las cuales son extraídas del desempeño y comportamiento de los modelos candidatos a formar el ensamble; esta extracción se calcula en cada paso del horizonte de predicción. La principal diferencia de este método con respecto a los existentes, está en que SOMSM utiliza *a priori* la diversidad entre modelos representándola en meta-características, es decir, antes de decidir los modelos componentes del ensamble. Además, la definición del error-representativo en conjunto con la red SOM es original de esta investigación.

Dado que los cambios de comportamiento de las series caóticas son impredecibles, propusimos seguir la estrategia de los expertos humanos al analizar el desempeño de los modelos con tres ventanas definidas. La segunda aportación del proyecto es la creación y análisis de un nuevo método de selección y combinación de modelos, llamado TVC, que considera el desempeño en diferentes ventanas de predicción, utilizando una validación cruzada Monte Carlo.

Estos modelos, hasta donde conocemos, no han sido reportados previamente por otros autores. Los resultados de esta investigación han sido difundidos en las siguientes publicaciones:

1. Fonseca-Delgado, R., & Gómez-Gil, P. (2016). Modeling Diversity in Ensembles for Time-Series Prediction Based on Self-Organizing Maps. En E. Merényi, J. M. Mendenhall, & P. O'Driscoll (Edits.), *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 11th International Workshop on Self Organizing Maps, WSOM 2016* (págs. 119-128). Houston, Texas, USA: Springer International Publishing.
2. Fonseca, R., & Gomez, P. (2016). Automatic Model Selection in Ensembles for Time Series Forecasting. *IEEE Latin America Transactions* (Vol 14, págs. 3811-3819).
3. Fonseca-Delgado, R., & Gomez-Gil, P. (2014). Selecting and combining models with self-organizing maps for long-term forecasting of chaotic time series. *Neural Networks (IJCNN), 2014 International Joint Conference on*, (págs. 2616-2623).
4. Fonseca, R., & Gómez-Gil, P. (2014). Temporal Validated Meta-Learning for Long-Term Forecasting of Chaotic Time Series Using Monte Carlo Cross-Validation. En O. Castillo, P. Melin, W. Pedrycz, & J. Kacprzyk (Edits.), *Recent Advances on Hybrid Approaches for Designing Intelligent Systems* (Vol. 547, págs. 353-367). Springer International Publishing.
5. Fonseca-Delgado, R., & Gómez-Gil, P. (2013). An assessment of ten-fold and Monte Carlo cross validations for time series forecasting. *Electrical Engineering, Computing Science and Automatic Control (CCE), 2013 10th International Conference on*, (págs. 215-220).

6. Fonseca-Delgado, R., & Gómez-Gil, P. (2013). Temporal Self-Organized Meta-Learning for Predicting Chaotic Time Series. *Research in Computing Science , 61* (ISSN: 1870-4069), (págs. 13-22).

1.8 Organización del Documento

Este documento está organizado de la siguiente manera: el capítulo dos presenta los conceptos necesarios para entender claramente el problema y el resto de la tesis. El capítulo tres resume el estado del arte. El capítulo cuatro detalla los algoritmos desarrollados. Los experimentos realizados y sus resultados se presentan en el capítulo cinco. El capítulo seis presenta las conclusiones y el trabajo futuro.

Capítulo 2. Conceptos Preliminares

Este capítulo describe los conceptos básicos y la notación utilizada en las siguientes secciones de esta investigación. Se incluyen definiciones de predicción de series de tiempo, medidas de error, caos, auto-organización, una descripción de la red neuronal SOM, meta-aprendizaje y ensambles.

2.1 Predicción de Series de Tiempo a Largo Plazo

Una serie de tiempo de una variable Y es una secuencia de observaciones y_t registradas a un tiempo t; esta investigación asume intervalos de registro fijos, lo que, implica una frecuencia de muestreo constante. La predicción de múltiples pasos se refiere a la estimación de una secuencia de h valores a futuro basada en observaciones previas y actuales, donde h es conocido como horizonte de predicción (Taieb, et al., 2012). Según el tamaño del horizonte de predicción y la frecuencia de muestreo dada, se puede catalogar a la predicción como de corto, mediano o largo plazo (Crone, et al., 2011); esto depende de cada dominio (ver Anexo A). Aún cuando el objetivo sea realizar una predicción a largo plazo, es importante conocer la exactitud en cada paso para poder confiar en las predicciones finales.

Existen diferentes estrategias para estimar una secuencia de h valores a futuro. Entre ellas, se encuentra la predicción iterativa o recursiva, la cual estima el horizonte de predicción utilizando el mismo modelo y se muestra en la ecuación (2.1). Esta estrategia comienza estimando un valor $\hat{y}_{t+1}$, luego utiliza este valor predicho $\hat{y}_{t+1}$ para estimar el siguiente $\hat{y}_{t+2}$ y así en adelante(Bao, et al., 2014). La predicción iterativa puede ser expresada como:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-m+1}), \tag{2.1}$$
$$\hat{y}_{t+2} = f(\hat{y}_{t+1}, y_t, y_{t-1}, \dots, y_{t-m+2}), \dots,$$
$$\hat{y}_{t+h} = f(\hat{y}_{t+h-1}, \hat{y}_{t+h-2}, \dots, \hat{y}_{t-m+h})$$

La función f estima cada paso para alcanzar h valores futuros utilizando m valores del pasado de Y. La expresión general de la predicción de múltiples pasos puede ser escrita como:

$$\{\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+h}\} = F(y_t, y_{t-1}, \dots, y_{t-m+1}) \tag{2.2}$$

donde la función F es el modelo de predicción(Brockwell & Davis, 2006).

Existen diversos tipos de modelos de predicción. De acuerdo a la manera en que manejan su relación con los valores del pasado, pueden categorizarse en lineales y no-lineales; esta investigación utiliza los modelos ARIMA (lineales), NARX y K-trayectorias (no-lineales); una breve descripción de estos modelos está disponible en el Anexo F.

2.2 Revisión de Medidas de Error

La evaluación de predicciones de múltiples pasos ha sido ampliamente estudiada; a continuación, se incluye una descripción detallada de las principales métricas asociadas a series de tiempo, marcando sus ventajas y desventajas.

Sea y_t una observación en el tiempo t y $\hat{y}_t$ la predicción correspondiente. El error de predicción se define $e_t = y_t - \hat{y}_t$. El objetivo es calcular las predicciones *out-of-sample*, que es cuando el tiempo de predicción está después de n, qué es el número de observaciones de la serie, $\hat{y}_{n+1}, \dots, \hat{y}_{n+h}$. Este cálculo se realiza a partir de los datos de la serie que participaron en el entrenamiento $t = 1, \dots, n$. Cabe hacer notar que en general existen dos escenarios más posibles para predicción, los cuales no son considerados en esta investigación. El primero es el conocido como "*in-sample"* que se refiere a cuando los valores a predecir están en tiempos anteriores a n. El otro escenario es cuando se predice un solo valor al futuro en un tiempo definido.

2.2.1 Medidas Dependientes de la Escala

Existen algunas medidas comúnmente utilizadas para medir la precisión cuando la escala del resultado depende de la escala de los datos. Estas son útiles cuando se comparan diferentes métodos aplicados a los mismos datos, pero no deben ser utilizadas cuando se compara con series de tiempo con diferentes escalas (Hyndman & Koehler, 2006)(Armstrong & Collopy, 1992) (Armstrong, 1985). La Tabla 2.1 muestra las principales medidas de este tipo, acompañadas de sus siglas en inglés, la definición de la medida y el número de ecuación.

Tabla 2.1. Medidas de predicción dependientes de la escala.

Medida	**Siglas en Inglés**	**Definición**	**No. Eq.**
Promedio del Error al Cuadrado (*Mean Square Error*)	MSE	$\text{mean}(e_t^2)$	(2.3)
Raíz del Promedio del Error al Cuadrado (*Root Mean Square Error*)	RMSE	$\sqrt{\text{MSE}}$	(2.4)
Promedio del Error Absoluto (*Mean Absolute Error*)	MAE	$\text{mean}(\lvert e_t \rvert)$	(2.5)
Mediana del Error Absoluto (*Median Absolute Error*)	MdAE	$\text{median}(\lvert e_t \rvert)$	(2.6)

Una de las funciones más utilizadas por expertos de predicción para seleccionar modelos en dominios definidos es RMSE(Armstrong & Collopy, 1992)(Armstrong, 1985); esto se debe a su facilidad de interpretación. En este proyecto de investigación, RMSE será una de las funciones disponibles para seleccionar modelos, recomendada cuando se quiera trabajar con una serie de tiempo particular. RMSE puede utilizarse con series de diferente escala, si se realiza una normalización previa de los valores de las series.

2.2.2 Medidas Basadas en el Porcentaje de Error

El porcentaje de error está definido por:

$$p_t = 100e_t/y_t \tag{2.7}$$

El porcentaje de error tiene la ventaja de ser independiente de la escala, y es utilizado frecuentemente para comparar el desempeño de predicciones en diferentes conjuntos de datos. Las medidas basadas en el porcentaje de error más utilizadas se muestran en la Tabla 2.2, acompañadas de sus siglas en inglés, la definición de la medida y el número de ecuación (Hyndman & Koehler, 2006)(Armstrong & Collopy, 1992)(Armstrong, 1985).

Tabla 2.2. Medidas basadas en el porcentaje de error.

Medida	Siglas en Inglés	Definición	No. Eq.
Promedio del Porcentaje de Error Absoluto (*Mean Absolute Percentage Error*)	MAPE	$\text{mean}(\lvert p_t \rvert)$	(2.8)
Mediana del Porcentaje de Error Absoluto (*Median Absolute Percentage Error*)	MdAPE	$\text{median}(\lvert p_t \rvert)$	(2.9)
Raíz del Promedio del Porcentaje de Error al Cuadrado (*Root Mean Square Percentage Error*)	RMSPE	$\sqrt{\text{mean}(p_t^2)}$	(2.10)
Raíz de la Mediana del Porcentaje de Error al Cuadrado (*Root Median Square Percentage Error*)	RMdSPE	$\sqrt{\text{median}(p_t^2)}$	(2.11)

Estas medidas tienen la desventaja de ser infinitas o indefinidas si $y_t = 0$ para cualquier t en el período de interés, y tener una distribución sesgada cuando y_t es cercano a zero. Las medidas MAPE y MdAPE tienen la desventaja de castigar más errores positivos que negativos, por esta razón se motivó el uso de las medidas llamadas "simétricas". La Tabla 2.3 muestra estas medidas, acompañadas de sus siglas en inglés, la definición de la

medida y el número de ecuación (Hyndman & Koehler, 2006)(Armstrong & Collopy, 1992).

Tabla 2.3. Medidas "simétricas" basadas en el porcentaje de error.

Medida	Siglas en Inglés	Definición	No. Eq.
Promedio Simétrico del Porcentaje de Error Absoluto (*Symmetric Mean Absolute Percentage Error*)	sMAPE	$\text{mean}(200\lvert y_t - \hat{y}_t\rvert/(y_t + \hat{y}_t))$	(2.12)
Mediana Simétrica del Porcentaje de Error Absoluto (*Symmetric Median Absolute Percentage Error*)	sMdAPE	$\text{median}(200\lvert y_t - \hat{y}_t\rvert/(y_t + \hat{y}_t))$	(2.13)

Con la finalidad de no verse afectados cuando los valores predichos son negativos, en las competencias de predicción NN3 y NN5 (Crone, 2010) modificaron la función sMAPE, incluyendo valores absolutos de los denominadores, y continuaron llamándola Promedio Simétrico del Porcentaje de Error Absoluto, aunque las siglas utilizadas son SMAPE. La ecuación que define SMAPE es:

$$\text{SMAPE}(Y, \hat{Y}) = \frac{\sum_{i=1}^{h} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}}{h} \cdot 100 \tag{2.14}$$

Donde Y y $\hat{Y}$ son la secuencias de valores esperados y_t y predichos $\hat{y}_t$ respectivamente; el número de elementos de ambas secuencias es igual al horizonte de predicción representado por h. SMAPE devuelve valores entre 0 y 200, donde un valor de 0 significa una predicción que coincide exactamente con los valores esperados, mientras que, un valor de 200 implica la peor predicción posible.

La función SMAPE es una de las más utilizadas para comparar el desempeño de modelos entre series de diferente tipo por su ventaja de ser independiente de la escala de los datos(Crone, et al., 2008)(Lemke & Gabrys, 2010). En este proyecto de investigación, SMAPE será la medida por defecto para seleccionar modelos, recomendada cuando se

quiera comparar con serie de tiempo de diferente tipo. Es importante mencionar que si el valor esperado y_i es 0 y el valor predicho $\hat{y}_i$ es 0 SMAPE devolverá una forma indeterminada, por lo que se recomienda trasladar los valores de las series de tiempo a valores mayores a cero.

2.2.3 Medidas Basadas en Errores Relativos

Una alternativa a escalar los datos es dividir cada error por el error obtenido utilizando la predicción $\hat{y}'_t$ de otro método de base definido. Este método base definido generalmente es caminata aleatoria (*random walk*), donde la predicción es la última observación de la serie. Una alternativa a caminata aleatoria, frecuentemente utilizada, es el promedio de todas las observaciones. La Tabla 2.4 muestra estas medidas, acompañadas de sus siglas en inglés, la definición de la medida y el número de ecuación (Armstrong & Collopy, 1992).

Tabla 2.4. Medidas basadas en errores relativos

Medida	Siglas en Inglés	Definición	No. Eq.
Error Absoluto Relativo (*Relative Absolute Error*)	RAE	$r_t = \lvert y_n - \hat{y}_n \rvert / \lvert y_n - \hat{y}'_n \rvert$	(2.15)
Mediana del Error Absoluto Relativo (*Median Relative Absolute Error*)	MdRAE	$\text{median}(r_t)$	(2.16)
Promedio Geométrico del Error Absoluto Relativo (*Geometric Mean Relative Absolute Error*)	GMRAE	$\text{geometric mean}(r_t)$	(2.17)

RAE devuelve mediciones tolerantes a valores extremos *"outlayers*", al dividir su comparación de las predicciones realizadas por la predicción de un modelo definido, pero puede dar errores infinitos cuando la predicción de *random walk* coincide con la observación. Al evaluar múltiples pasos de predicción, RAE devolverá un vector de un tamaño igual al horizonte de predicción h. El valor en cada posición es una evaluación de r_t en el instante de tiempo respectivo. Una forma de evitar el problema de las mediciones infinitas devueltas por RAE es utilizando la mediana. Armstrong y Collopy a partir de sus

experimentos sugieren utilizar MdRAE para seleccionar el mejor modelo cuando se cuenta con pocas series de tiempo.

Una forma de resumir mediciones de errores relativos es utilizando la media geométrica. GMRAE es capaz de capturar errores pequeños, cuando un cambio es realizado en un modelo, así es fácil observar los efectos del cambio en el desempeño. A partir de las predicciones en cada instante de tiempo GMRAE puede escribirse de la siguiente manera(Kourentzes, et al., 2014):

$$\mathrm{GMRAE}(Y, \hat{Y}) = \sqrt[h]{\prod_{i=1}^{h} \frac{|y_i - \hat{y}_i|}{|y_i - \hat{y}'_i|}} \tag{2.18}$$

GMRAE es recomendado cuando la tarea involucra es calibrar un modelo para un conjunto de series de tiempo. En este proyecto de investigación, los resultados serán evaluados con GMRAE para tener un punto de vista diferente. Incluso GMRAE estará disponible entre la medidas de evaluación para seleccionar modelos, sin embargo no es recomendada por la posibilidad de devolver valores infinitos cuando la predicción el modelo base definido sea igual al valor esperado.

2.2.4 Medidas Relativas

En lugar de utilizar errores relativos, es posible utilizar medidas relativas. Por ejemplo, sea MAE' el MAE de un método definido. Entonces, el MAE relativo está definido por:

$$\mathrm{RelMAE} = \frac{\mathrm{MAE}}{\mathrm{MAE}'} \tag{2.19}$$

El mismo tipo de medidas pueden definirse con RMSE, MAPE, etc. La ventaja de utilizar medidas relativas es su fácil interpretación. Por ejemplo RelMAE mide la mejora posible del método de predicción propuesto con respecto al método definido. Cuando $\mathrm{RelMAE} < 1$, el método de predicción propuesto es mejor que el método definido y viceversa. En este proyecto de investigación, los resultados serán evaluados con RelMAE para comparar la diferencia con respecto a la medida basada en el error relativo.

2.2.5 Errores Escalados

Según Hyndman y Koehler: "*A pesar de dos décadas en medidas de error de predicciones, pensamos que algunos problemas fundamentales han sido pasados por alto*" (Hyndman & Koehler, 2006). Las medidas de error tienen problemas, pueden devolver valores infinitos o indefinidos en situaciones comunes. Extendiendo las medidas relativas, el error escalado se define por:(Hyndman & Koehler, 2006).

$$Q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \tag{2.20}$$

El Promedio del Error Escalado Absoluto (MASE por *Mean Absolute Scaled Error*) se define por:

$$\text{MASE} = \text{mean}(|Q_t|) \tag{2.21}$$

Un valor mayor a uno indica que la predicción es pero, en promedio, que la predicción *in-sample* de un paso a la vez con un método *naïve* (Hyndman & Koehler, 2006). En este proyecto de investigación, MASE se recomienda cuando hay series de escalas muy diferentes, con valores cercanos a cero o negativos.

Desafortunadamente, no hay una sola medida de evaluación recomendada para todos los escenarios posibles. Revisando los trabajos existentes a la fecha, notamos que las medidas dependientes de la escala, en particular RMSE, son ampliamente utilizadas por expertos de predicción, principalmente por su facilidad de interpretación (Armstrong & Collopy, 1992) (Armstrong, 1985). Las medidas basadas en el porcentaje de error, en particular SMAPE, sirven para comparar el desempeño de diferentes modelos con series de tiempo de diferente tipo (Makridakis & Hibon, 2000) (Crone, et al., 2008). Las medidas basadas en errores relativos son independientes de la escala, de las cuales GMRAE brinda mejores resultados al ajustar un modelo a un tipo de serie de tiempo(Armstrong & Collopy, 1992). Las medidas relativas comparan la evaluación utilizando una medida con respecto a la evaluación de un tercer modelo con la misma medida, un ejemplo es RelMAE. Los errores escalados consideran el error con respecto al error de una predicción de un solo paso por un método simple; una de estas medidas es MASE, la cual es útil cuando se

compara con series de tiempo de escalas muy diferentes y valores cercanos a cero (Hyndman & Koehler, 2006). Este proyecto evalúa las predicciones utilizando RMSE, SMAPE, GMRAE, RelMAE y MASE; las mismas funciones están disponibles para seleccionar los modelos en el marco de trabajo. La medida establecida por defecto es SMAPE y se seleccionó por ser independiente de la escala, simple, con un rango de salida definido y utilizada comúnmente en la literatura para comparar trabajos académicos.

2.3 Caos y Auto-Organización

La teoría del caos lidia con el análisis de sistemas dinámicos no lineales sensibles a cambios en condiciones iniciales. En general, todos los sistemas caóticos son no-lineales, sin embargo, no todos los sistemas no-lineales son caóticos (Kantz, 1994). Dhanya y Kumar (2010) resumen las características de los sistemas caóticos en: (i) estos sistemas son deterministas, es decir, existe algún sistema de ecuaciones deterministas gobernando su comportamiento, (ii) estos sistemas son sensibles a las condiciones iniciales, esto es, un pequeño cambio en el punto inicial puede conllevar resultados significativamente diferentes; (iii) estos sistemas no son aleatorios ni desordenados. Los sistemas caóticos tienen sentido de orden y patrones, incluso cuando ellos no se repitan. En este trabajo, se considera a una serie de tiempo caótica como aquella que es producto de un sistema caótico (ver ejemplos en el Anexo D). La principal dificultad al tratar de predecir este tipo de series es que el error crece exponencialmente con el tamaño del horizonte de predicción (véase Anexo E).

Un sistema es auto-organizado si adquiere una estructura espacial, temporal o funcional sin interferencia específica desde el exterior. Por "específica", queremos decir que la estructura o el funcionamiento no está impuesto en el sistema (Haken, 2006). Un sistema complejo puede mostrar coordinación y cooperación espontánea entre sus elementos, siempre que las interacciones entre los elementos del sistema estén bien escogidas (Helbing, 2012). Esto implica que, las regulaciones para modelar un sistema complejo solo deben definir los límites de acción, en lugar de especificar con precisión qué deben hacer los elementos del sistema. En otras palabras, las regulaciones definen “las reglas del juego”. Esta estrategia debe proveer al sistema de suficientes grados de libertad

para auto-organizar buenas soluciones. Si las reglas de interacción están bien establecidas, este enfoque usualmente conduce a un comportamiento del sistema flexible y adaptable (Helbing, 2012).

La principal ventaja de los sistemas auto-organizados es su capacidad de adaptación, en esta investigación aprovechamos esta ventaja para adaptar el sistema de predicción al comportamiento caótico de la serie de tiempo.

2.4 Mapas Auto-Organizados SOM

El objetivo principal de los mapas auto-organizados (SOM por sus siglas en inglés), propuestos por T. Kohonen (1982), es transformar un conjunto de patrones de entrada de una dimensión arbitraria en un mapa discreto de una o dos dimensiones, y realizar esta transformación adaptivamente de manera topológicamente ordenada(Haykin, 2009). La Figura 2.1 muestra un diagrama de un mapa de neuronas de dos dimensiones, donde cada neurona, representada por un hexágono, está conectada con el vector de entrada. La distribución de las neuronas en el mapa sigue un patrón hexagonal.

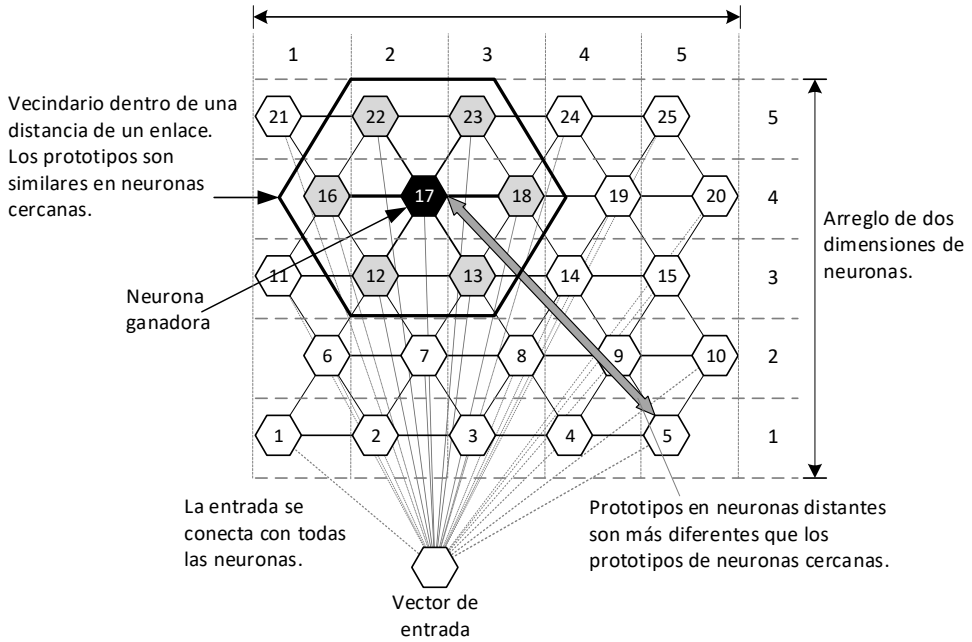


Figura 2.1. Estructura hexagonal de una red SOM de 5 filas y 5 columnas. El vector de entrada se conecta con todas las neuronas en la siguiente capa; el vecindario dentro de una distancia de un enlace a partir de la neurona marcada como ganadora está resaltado en negrita (imagen basada en (Fonseca-Delgado & Gomez-Gil, 2014)).

A diferencia de otros modelos neuronales, cada neurona de salida en una red SOM tiene un vector de pesos, representando un prototipo de un grupo; éste prototipo se adapta a los datos durante el proceso de entrenamiento.

El algoritmo responsable de la formación de los mapas auto-organizados, que es no supervisado, comienza inicializando los pesos sinápticos en la red. Esto puede lograrse asignándoles valores pequeños tomados de un generador de número aleatorios; al hacerlo, no se impone ningún orden previo en el mapa. Una vez que la red ha sido inicializada, hay dos procesos esenciales involucrados en la formación del mapa auto-organizado, los que se exponen a continuación:

1. *Competir*.- Para cada ejemplo de entrada, las neuronas en la red computan sus valores respectivos de la función discriminante. Esta función discriminante provee las bases de la competencia entre las neuronas. La neurona con el mayor valor de la función discriminante es declarada ganadora de la competición.
2. *Cooperar*.- La neurona ganadora determina la ubicación espacial de un vecindario topológico de neuronas excitadas, la que incrementan sus valores individuales de la función discriminante en relación al ejemplo de entrada, a través de un ajuste adecuado aplicado a los pesos sinápticos. El ajuste es realizado tal que la respuesta de la neurona ganadora a la subsecuente aplicación de un ejemplo de entrada similar es realzada.

2.4.1 Competir

Sea $\mathbb{m}$ la dimensión del espacio de entrada (datos). Sea x un ejemplo de entrada (vector) seleccionado aleatoriamente del espacio de entrada definido por:

$$x = [x_1, x_2, \dots, x_{\mathbb{m}}] \tag{2.22}$$

El vector de pesos sinápticos de cada neurona en la red tiene la misma dimensión que el espacio de entrada. Sea el vector de peso sináptico de la neurona j denotado por:

$$w_j = \left[w_{j1}, w_{j2}, \dots, w_{j\mathbb{m}}\right], j = 1,2, \dots, n_N \tag{2.23}$$

donde n_N es el número total de neuronas en la red.

Para encontrar la neurona más semejante al vector de entrada x, todas las neuronas *compiten* y son evaluadas con la función discriminante de distancia Euclidiana; la neurona ganadora $i(x)$ es aquella que tiene un vector de pesos (w_j) con menor distancia Euclidiana al ejemplo de entrada *x* (Haykin, 1999):

$$i(x) = \underset{j}{\operatorname{argmin}} \|x - w_j\|, j = 1,2, \dots, n_N \tag{2.24}$$

la neurona $i(x)$ también es conocida como "mejor emparejamiento".

2.4.2 Cooperar

En el aprendizaje competitivo la neurona ganadora adapta su vector de pesos para aproximar mejor el ejemplo de entrada "se lleva todo", éste proceso permite mapear de un espacio de alta dimensión a un mapa de dos dimensiones; la aportación de las redes SOM propuestas por Kohonen (1982) radica en el proceso de cooperación que permite a las neuronas adaptarse a los ejemplos de entrada manteniendo su relación estructural. La cooperación de la neurona ganadora con sus neuronas vecinas se representa en la actualización de pesos definida por (Haykin, 2009):

$$w_j(n+1) = w_j(n) + \eta(n) v_{j,i(x)}(n) \left(x - w_j(n)\right) \tag{2.25}$$

donde: n representa la iteración actual, $w_j(n+1)$ es el vector de pesos actualizado de la neurona j, $\eta(n)$ es el parámetro coeficiente de aprendizaje, inicia con un valor η_0 y decrece gradualmente cuando n incrementa (Haykin, 1999). La función $v_{j,i(x)}$ define la influencia de la neurona ganadora $i(x)$ en su vecindario, en nuestro caso el vecindario de la neurona ganadora sigue un patrón hexagonal y la influencia que recibe una neurona j depende de su distancia de enlaces con la neurona $i(x)$. La distancia de enlaces es el número de saltos que se deben realizar para conectar la neurona i con la neurona j en el mapa de dos dimensiones (Beale, et al., 2015); la Figura 2.1 muestra el vecindario de un enlace alrededor de la neurona 17.

2.4.3 Arquitectura de la Red SOM

La arquitectura de la red organizada en dos dimensiones es usualmente efectiva para la aproximación de las relaciones similares de datos de alta dimensión(Kohonen, 2013); dentro de los arreglos regulares de dos dimensiones, a veces se prefieren arreglos rectangulares por simplicidad; sin embargo, los arreglos hexagonales son más ilustrativos, precisos y preferibles (Kohonen, 2013), (Kohonen, 2001). La topología hexagonal es efectiva para visualizar el mapa porque los prototipos de cada neurona se distribuyen más suavemente que al utilizar una topología rectangular, donde la distribución genera bordes que siguen el patrón rectangular. Aprovechamos la suavidad de la topología hexagonal con la distancia de enlace para tener vecindarios circulares y poder apreciar visualmente posibles patrones en los mapas.

2.4.4 Establecer el Número de Neuronas

El número de neuronas (n_N) de una red SOM corresponde al número de grupos de vectores creados durante el entrenamiento. Sin embargo, establecer el número óptimo de neuronas es un problema abierto, aunque, algunas estrategias han sido propuestas para realizar esto. De especial interés es el trabajo de Milligan y Cooper (1985) donde fueron analizados 30 procedimientos para evaluar la calidad de un agrupamiento, los autores encontraron que el índice "Criterio de Razón de Varianza" (VRC por sus siglas en inglés) (Calinski & Harabasz, 1974) alcanza "excelentes resultados". En este trabajo se utiliza el índice VRC para determinar n_N, dado un conjunto de posibles valores para n_N, diferentes redes SOM son entrenadas, una por cada valor de n_N; el agrupamiento devuelto por cada red se evalúa con el índice VRC. El número de neuronas n_N a seleccionar es el de la red SOM cuyo agrupamiento obtenga el máximo VRC.

El índice VRC busca maximizar la similitud entre los miembros del mismo grupo y la disimilitud entre miembros de grupos diferentes; el índice VRC se define como (Calinski & Harabasz, 1974):

$$VRC = \frac{Trace_B}{Trace_W} \times \frac{n_E - n_N}{n_N - 1} \tag{2.26}$$

donde $Trace_B$ representa la varianza entre grupos, $Trace_W$ es la varianza dentro de los grupos, n_E es el número de elementos agrupados y n_N es el número de grupos, que es igual al número de neuronas de la red SOM. La varianza entre grupos se define como (Calinski & Harabasz, 1974):

$$Trace_B = \sum_{j=1}^{n_N} |C_j| \|w_j - \bar{\omega}\|^2 \tag{2.27}$$

donde $\bar{\omega}$ es el vector media de los vectores de entrada, $|C_j|$ es el número de elementos en el grupo C_j, de la neurona j, cuyo centroide w_j es el vector de pesos de la neurona j. La función $\|v_1 - v_2\|$ es la distancia Euclidiana entre los vectores v_1 y v_2. La varianza dentro del grupo se define como (Calinski & Harabasz, 1974):

$$Trace_W = \sum_{j=1}^{n_N} \sum_{x \in C_j} \|x - w_j\|^2 \tag{2.28}$$

donde x es uno de los vectores de entrada pertenecientes al grupo C_j.

Un buen agrupamiento debe tener un valor grande de $Trace_B$ y un valor pequeño de $Trace_W$; por lo tanto, un agrupamiento es considerado de mejor calidad que otro agrupamiento cuando tiene un mayor valor en su índice VRC.

2.5 Meta-Aprendizaje

Lemke y colaboradores (2013) revisaron varias definiciones de meta-aprendizaje propuestas en la literatura (Vanschoren, 2010), (Brazdil, et al., 2009), (Giraud-Carrier, 2008), (Vilalta & Drissi, 2002), (Biggs, 1985) y (Rice, 1976) a partir de estas definiciones los autores propusieron y analizaron una definición que generaliza a las anteriores (Lemke, et al., 2013). Esta definición de meta-aprendizaje establece que:

1. Un sistema de meta-aprendizaje debe incluir un subsistema de aprendizaje, el cual se adapta con la experiencia.

2. La experiencia se adquiere al explotar el meta-conocimiento extraído en un episodio previo sobre un único conjunto de datos, y/o de diferentes dominios o problemas.

En esencia, el concepto de meta-aprendizaje implica aprovechar el conocimiento previo del funcionamiento de los modelos de predicción para tomar decisiones (Vilalta & Drissi, 2002). Algunos trabajos extraen conocimiento a partir de episodios previos con diferentes conjuntos de datos, por ejemplo: (Wang, et al., 2009) , (Prudêncio & Ludermir, 2004), (Lemke & Gabrys, 2010) procuran discriminar las series de tiempo a través de un vector de características propias de cada serie y relacionan este vector con el desempeño de un modelo en particular sobre esa serie; esta relación entre características de los datos de la serie y el desempeño de un modelo en particular representa el conocimiento extraído. En este proyecto de investigación el meta-conocimiento es extraído de episodios previos sobre la misma serie de tiempo; este conocimiento se representada en meta-características relacionadas al desempeño y comportamiento de los modelos disponibles para la construcción del ensamble con la serie de tiempo. Estas meta-características permiten decidir que modelos seleccionar y cómo combinarlos.

2.6 Ensambles

Un ensamble combina predicciones de la misma serie de tiempo provenientes de diferentes modelos que pueden o no ser del mismo tipo; la finalidad de un ensamble es alcanzar un desempeño mejor que el alcanzado por sus modelos componentes (Bates & Granger, 1969)(Dzeroski & Zenko, 2004). La diversidad de comportamiento en un ensamble es el grado de desacuerdo entre los modelos de predicción evaluados en la misma secuencia de prueba(Mendes-Moreira, et al., 2012). En el área de regresión, Ueda y Nagano (1996) demostraron la importancia de la diversidad entre los modelos seleccionados en la definición de un ensamble; la demostración se fundamenta en la descomposición sesgo-varianza-covarianza (Ueda & Nakano, 1996), la cual establece que el error de generalización de un ensamble de M modelos de regresión puede descomponerse siguiendo la siguiente ecuación (Ren, et al., 2016):

$$GErr\left(f_{ens}^{(M)}\right) = E\left\{\overline{\text{bias}}^2 + \frac{1}{M}\overline{\text{var}} + \left(1 - \frac{1}{M}\right)\overline{\text{covar}}\right\} \tag{2.29}$$

donde G_{Err} es el error de generalización, $E\{\cdot\}$ es la esperanza matemática, $\overline{\text{bias}}$, $\overline{\text{var}}$ y $\overline{\text{covar}}$ denotan el sesgo, la varianza y la covarianza promedio de los M modelos, respectivamente. El primer concepto denominado sesgo promedio se define por(Brown, et al., 2005):

$$\overline{\text{bias}} = \frac{1}{M}\sum_i (E\{f_i\} - x) \tag{2.30}$$

donde f_i es el valor devuelto por el modelo i-ésimo y x es el valor esperado.

La varianza promedio de los componentes del ensamble se define por(Brown, et al., 2005):

$$\overline{\text{var}} = \frac{1}{M}\sum_i E\{(f_i - E\{f_i\})^2\} \tag{2.31}$$

La covarianza promedio de los modelos componentes del ensamble se define por(Brown, et al., 2005):

$$\overline{\text{covar}} = \frac{1}{M(M-1)}\sum_i \sum_{j \neq i} E\{(f_i - E\{f_i\})(f_j - E\{f_j\})\} \tag{2.32}$$

La ecuación (2.29) ilustra que adicionalmente al sesgo y la varianza de los modelos individuales, el error de generalización de un ensamble también depende de la covarianza entre los modelos individuales. Los primeros dos términos de la ecuación (2.29) solo pueden ser positivos, siendo la covarianza promedio el único término que puede ser negativo. Modelos con un comportamiento diferente pueden retornar una covarianza promedio negativa; en consecuencia, la diversidad de comportamiento entre modelos permite mejorar el error de generalización de un ensamble, siempre que se tenga en cuenta el desempeño y la varianza de los modelos(Brown, et al., 2005).

2.6.1 Fuentes de Diversidad

La clave de un ensamble es la diversidad del comportamiento de sus modelos, la principales fuentes de diversidad son: diversidad de datos, diversidad de parámetros y diversidad de los tipos de modelos (o modelos base)(Ren, et al., 2016).

2.6.1.1 Diversidad de Datos

La diversidad de datos consiste en generar múltiples conjuntos de datos a partir del conjunto original para entrenar diferentes modelos de predicción. Los conjuntos de datos deben ser diferentes uno del otro para que los modelos después de ser entrenados devuelvan salidas diferentes. Los métodos de ensambles para diversidad de datos comúnmente utilizados son: *bootstrap aggregation* (*Bagging*) (Breiman, 1996), *adaptive boosting* (*AdaBoost*)(Freund, et al., 1996), *random subspace* (Ho, 1998) (Tan, et al., 2008), y *random forest*(Breiman, 1996).

La variación de las salidas esperadas es una estrategia diferente para crear diversidad. En lugar de generar diferentes conjuntos de entrenamiento en el espacio de entrada, múltiples salidas son creadas para el entrenamiento de los modelos, un ejemplo de este enfoque es *output smearing* (Breiman, 2000), el cual introduce diversidad al espacio de salida utilizando ruido aleatorio.

En esta investigación no se utilizó diversidad de datos considerando la sensibilidad a los cambios en las condiciones iniciales de las series de tiempo caóticas.

2.6.1.2 Diversidad de Modelos Base

También conocida como diversidad estructural, es obtenida introduciendo modelos de diferente tipo esto incluye con diferente estructura, arquitectura o algoritmo de entrenamiento. Los modelos de un mismo tipo son conocidos como modelos base; estos pueden variar en parámetros, arquitectura, etc. Los ensambles que utilizan modelos de diferente base son conocidos como ensambles heterogéneos; un ejemplo de esta estrategia es el método conocido como *stacking*, donde los diferentes modelos son obtenidos con diferentes algoritmos de aprendizaje en un único conjunto de datos para genera los modelos en el primer nivel o nivel 0 y luego otro algoritmo es utilizado en el nivel 1 para decidir cómo combinar los modelos del nivel 0(Wolpert, 1992). Como ejemplos de *stacking* se

pueden citar los siguientes: (Breiman, 1996), (Ting & Witten, 1999), (Yang, et al., 2001), entre otros.

En esta investigación recurre a modelos de diferentes bases para aumentar la diversidad de comportamiento, esto incluye entrenar la misma estructura con algoritmos de entrenamiento diferente.

2.6.1.3 Diversidad de Parámetros

Los modelos de predicción pueden verse como ecuaciones cuyos parámetros determinan sus propiedades principales, por ejemplo: el grado de la ecuación, el número de variables, o los valores a que alcanzan las constantes después del entrenamiento. Considerando esto, aún utilizando el mismo conjunto de entrenamiento, es posible obtener modelos con un comportamiento diferente en sus salidas al utilizar diferentes valores en los parámetros de los modelos base (Ren, et al., 2016).

En este proyecto se considera la relación en el comportamiento de los modelos para seleccionar que modelos combinar; con respecto a las fuentes de diversidad, se genera diversidad de parámetros y de los tipos de modelos.

2.7 Resumen del Marco Teórico

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 de la frecuencia de muestreo y 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, SMAPE, GMRAE, RelMAE y MASE.

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. Una herramienta poderosa en la predicción de series de tiempo es el uso de ensambles, las características más importantes a considerar en la construcción de un ensamble son la exactitud y la diversidad de los modelos a combinar. En esta investigación proponemos utilizar las reglas simples de las redes SOM para seleccionar y combinar modelos de predicción considerando la exactitud y la diversidad de los modelos involucrados. El meta-aprendizaje nos permite abstraer la auto-organización de los modelos del funcionamiento particular de cada tipo de modelo. En el siguiente capítulo se presentarán los enfoques de investigaciones actuales con relación directa a esta investigación

Capítulo 3. Trabajo Relacionado

Este capítulo revisa los trabajos relacionados con la predicción de series de tiempo que utilizan estrategias similares a la utilizada en esta investigación. Como se comentó en el capítulo uno, varios expertos coinciden en que los mejores resultados para aplicaciones con sistemas caóticos se obtienen utilizando ensambles. La combinación de modelos de predicción inició alrededor del trabajo de (Bates & Granger, 1969); desde entonces un extenso conjunto de métodos basados en ensambles han sido creados. Esta ventaja, con respecto a modelos independientes, ha sido reportado en términos de incremento de robustez y exactitud (Garcia-Pedrajas, et al., 2005). Entonces, se incluye la revisión de algunos modelos de aprendizaje de ensambles de modelos de predicción. En este capítulo, los trabajos están agrupados bajo los siguientes criterios: basados en redes neuronales SOM, basados en meta-aprendizaje, y otros trabajos basados en ensambles.

3.1 Prediciendo Sistemas Caóticos

La predicción de series de tiempo caóticas se basa en la reconstrucción del comportamiento global de un sistema multi-dimensional a partir de lo que registramos en una de sus dimensiones; esto se denomina reconstrucción del espacio de fases, donde una fase es un estado del sistema multi-dimensional. Siguiendo el teorema de "embebido" de Takens (1981), un espacio de fases reconstruido es un espacio métrico de *m*-dimensiones que representa al sistema, posiblemente desconocido, que generó la serie de tiempo. Dada una serie de tiempo caótica $Y = \{y_t \in \mathbb{R} | 1 \leq t \leq N\}$, la dimensión embebida m, y un tiempo de retardo τ entre los valores tomados de la serie, el espacio de fases puede expresarse de la siguiente manera (Takens, 1981):

$$x_t = \left(y_t, y_{t-\tau}, \dots, y_{t-(m-1)\tau}\right), x_t \in \mathbb{R}^m, t = 1, \dots, L \tag{3.1}$$

donde x_t es un vector o un punto en la construcción del espacio de fases, $L = N - (m - 1)\tau$ es el número de puntos en el espacio de fases reconstruido. Asumiendo que no existe ruido en los datos, si la dimensión embebida es lo suficientemente grande, el espacio de fases es homeomorfo[1] al espacio de estados que generó la serie de tiempo; esto es, el espacio de fases contiene la misma información que el espacio de estados original. Existe una función de mapeo determinista $f(\cdot)$ que cumple la siguiente ecuación:

$$y_{t+\Delta t} = f(x_t) \tag{3.2}$$

Usualmente se utiliza un modelo local para predecir el siguiente paso, donde $\Delta t > 0$ es el paso de avance de la predicción. La función f es el modelo de predicción para la serie de tiempo caótica(Yuxia & Hongtao, 2012). Si asumimos un $\tau = 1$ y $\Delta t = 1$ la ecuación (3.2) equivale a la primera predicción de la ecuación (2.1). La Figura 3.1(a) muestra el "atractor"[2] de Lorenz (1963), donde cada punto en la imagen está definido con tres coordenadas (x, y, z). La Figura 3.1(b) muestra la serie de tiempo formada por la coordenada x del "atractor" de Lorenz. La Figura 3.1(c) muestra el "atractor" reconstruido a partir de la serie de tiempo de la coordenada $x(\text{Lorenz}, 1963)$.

a) b) c)

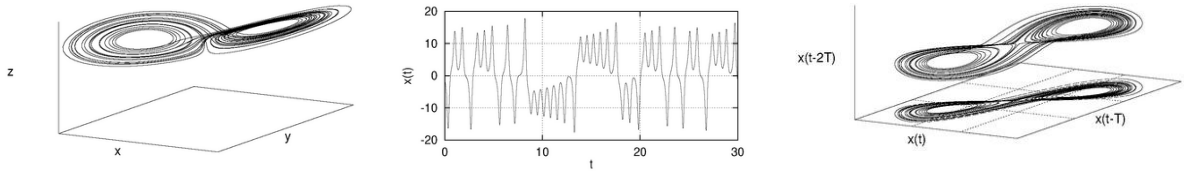


Figura 3.1. Ejemplo reconstrucción del "atractor" de Lorenz. a) Sistema de tres coordenadas, b) Serie de tiempo formada por la coordenada x, c) Reconstrucción a partir de la coordenada x (Sauer, 2006).

Algunos trabajos que utilizan ténicas para encontrar la reconstrucción del espacio embebido incluyen a (Kennel, et al., 1992), (Cao, 1997) y (Lei, et al., 2002). Estos métodos asumen la ausencia de ruído y son muy sensibles a cambios en sus parámetros, por lo que al

[1] Homeomorfo significa que existe una función uno a uno continua y cuya inversa es continua http://mathworld.wolfram.com/Homeomorphism.html.

[2] Un atractor es un conjunto de puntos en el espacio de fases hacia el cual un sistema tiende a evolucionar http://mathworld.wolfram.com/Attractor.html.

aplicarlos en dominios reales el desempeño disminuye. Una alternativa es aprovechar el conocimiento de expertos en el dominio para poder realizar las estimaciones de los valores m y τ (Han, et al., 2015), (Cao, et al., 2012), (Dhanya & Kumar, 2010), (Diaconescu, 2008), (Casdagli, 1992). Cómo se mencionó anteriormente, el costo de contar con expertos en predicción puede ser excesivo o inviable en varias aplicaciones, por lo tanto, este proyecto se enfoca en métodos que puedan funcionar automáticamente en diferentes dominios.

3.2 Ensambles Basados en Redes SOM

En el análisis de series de tiempo, las redes neuronales SOM han sido aplicadas en las tareas agrupamiento y visualización; ejemplos en este sentido son los siguientes trabajos: (Chappell & Taylor, 1993), (Koskela, et al., 1998), (Voegtlin, 2002), (Strickert & Hammer, 2005) y (Strickert, et al., 2005). Pocos trabajos utilizan redes neuronales SOM en la tarea de predicción, entre ellos se pueden diferenciar dos categorías: los trabajos que utilizan redes neuronales SOM para modelar y predecir la serie de tiempo, véase por ejemplo (Sorjamaa, et al., 2007), (Sorjamaa, et al., 2009) y (García-Rois & Burguillo, 2015), y en la otra categoría se utiliza redes neuronales SOM para seleccionar modelos de predicción en la construcción de ensambles; en esta categoría se ubica la presente investigación. En este sentido se puede citar el método SOMAR (Ni & Yin, 2009), que modifica el proceso de entrenamiento de la red SOM para seleccionar y ajustar modelos auto-regresivos (AR); los vectores de pesos asociados a cada neurona definen los parámetros de un modelo AR. La predicción se calcula combinando el modelo AR con mejor desempeño y algunos modelos locales que tienen una exactitud similar. Otro ejemplo de este tipo de modelos es NGMAR (Ouyang & Yin, 2014), el cual extiende a SOMAR y organiza una combinación de modelos AR, utilizando una variante del algoritmo SOM que se denomina gas neural (NG por sus siglas en inglés) (Martinetz, et al., 1993). NG introduce un ordenamiento del vecindario, donde la estructura de las neuronas de NG varía dinámicamente en lugar de ser un mapa predefinido como en SOM. NGMAR aprovecha el posicionamiento dinámico del vecindario del método NG, utilizando como medida de similitud la suma de los coeficientes de auto-correlación. Tanto SOMAR como NGMAR están limitados a combinar únicamente un tipo de modelo y no consideran que la diversidad del ensamble podría mejorarse si se utilizaran modelos de predicción de diferente tipo. Por el contrario, nuestra investigación se

apoya en una nueva meta-característica que permite crear un método de construcción de ensambles independiente del tipo de modelos de predicción utilizados.

3.3 Ensambles Basados en Meta-Aprendizaje

La idea de utilizar características de las series de tiempo para seleccionar una combinación de modelos de predicción apropiada ha sido perseguida desde los años 90. El primer sistema de este tipo estaba basado en reglas y construía una mezcla de métodos de juicio de expertos y cuantitativos (Collopy & Armstrong, 1992); los autores utilizaron características de las series de tiempo en la generación de 99 reglas de asignación de peso a cuatro modelos diferentes, de acuerdo a las características de las series; las reglas fueron obtenidas de la experiencia de expertos, tanto por inspección visual como utilizando conocimiento del dominio. Otros trabajos basados en la construcción de reglas con características de las series son (Adya, et al., 2001), (Vokurka, et al., 1996), (Arinze, et al., 1997), y (Meade, 2000).

La frase meta-aprendizaje en el contexto de series de tiempo fue adoptada por primera vez por la comunidad de aprendizaje computacional en (Prudêncio & Ludermir, 2004), donde se presentaron dos casos de estudio. En el primer caso, un árbol de decisión C4.5 es utilizado para enlazar seis características al desempeño de dos modelos de predicción. En el segundo, el enfoque NOEMON, introducido por (Kalousis & Theoharis, 1999), es utilizado para posicionar tres métodos. NOEMON construye clasificadores para cada par de modelos de predicción, utilizando las seis características para predecir el modelo más promisorio para cada problema de dos clases. Un posicionamiento es generado utilizando las salidas de los clasificadores. Wang y colaboradores (2009) proponen un método donde las series de tiempo se agrupan de acuerdo a nueve características incluyendo medidas de caos, auto-similitud y estadísticas tradicionales como tendencia, estacionalidad y *kurtosis*. En el primer paso, se generan reglas analizando el desempeño de grupos de modelos identificados en un mapa auto-organizado. Además, se genera automáticamente un árbol de decisión utilizando el algoritmo C4.5 con el mismo propósito. Este enfoque se extiende para determinar pesos en una combinación de modelos individuales, basado en características de los datos. Los trabajos nombrados anteriormente consideran diferentes

características de las series de tiempo, pero no consideran la diversidad entre los modelos. Un trabajo de este tipo que incluye medidas de diversidad es el propuesto por (Lemke & Gabrys, 2010), aunque no considera el comportamiento de los modelos en diferentes instantes de tiempo. En la siguiente sección se describe dicho trabajo.

3.3.1 Método de Lemke y Gabrys

El trabajo de Lemke y Gabrys (2010) revisa el estado del arte en la creación de ensambles basados en meta-aprendizaje para la predicción de series de tiempo; desde la generación de reglas de expertos humanos hasta ensambles híbridos basados en árboles de decisión. Después de ese análisis, Lemke y Gabrys proponen un método (denominado en este documento como "método de Lemke" por simplicidad) que comparan contra varios otros trabajos, las predicciones del método de Lemke evaluadas con SMAPE superan los resultados de los otros métodos. Por esa razón, este trabajo fue elegido como la línea base con la que se comparan los métodos desarrollados en esta investigación. El método de Lemke selecciona los modelos a combinar a partir de un conjunto de características, las cuales se detallan en el Anexo G. El trabajo de Lemke contiene varias estrategias para seleccionar los modelos a partir del conjunto de características definido, de ellas, la que mejor resultado obtuvo selecciona los modelos utilizando una función de posicionamiento entre modelos (*ranking*) que depende del desempeño evaluado con SMAPE. Este modelo requiere entrenarse con un conjunto de diferentes series de tiempo que compartan características similares, a diferencia del modelo propuesto en esta investigación, que se entrena con la misma serie de tiempo.

En el primer paso, se calcula una distancia entre los elementos del conjunto de series de tiempo involucradas en el entrenamiento. Esta distancia está basada en meta-características que asocian conocimiento del desempeño de los modelos de predicción con características propias de cada serie de tiempo. Para esto, sean f_ρ meta-características normalizadas, donde ρ es el número de identificación de una meta-característica; la distancia entre dos series s_i y s_j está definida por la norma sin pesar (Lemke & Gabrys, 2010):

$$distancia(s_i, s_j) = \sum_{\rho} \frac{\left| f_{\rho,s_i} - f_{\rho,s_j} \right|}{\max_{k \neq i}(f_{\rho,s_k}) - \min_{k \neq i}(f_{\rho,s_k})} \tag{3.3}$$

Posteriormente, las distancias son agrupadas utilizando el algoritmo *k*-medias, y las series en el grupo más cercano al de la serie de prueba son identificadas para una inspección posterior. El *ranking* se genera usando la variación de la "tasa de tasas ajustada" (ARR por sus siglas en inglés), la cual inicia calculándose sobre parejas de modelos c_p y c_q en la serie s_i, con la siguiente ecuación (Lemke & Gabrys, 2010):

$$ARR_{c_p,c_q}^{s_i} = \frac{SMAPE_{c_q}}{SMAPE_{c_p}} \tag{3.4}$$

Un valor alto de $ARR_{c_p,c_q}^{s_i}$ indica que el modelo c_p es más exacto que el modelo c_q prediciendo la serie s_i. El ARR del modelo c_p, con respecto a todas las series seleccionadas y los modelos disponibles, se obtiene con la siguiente ecuación (Lemke & Gabrys, 2010):

$$ARR_{c_p} = \frac{\sum_{c_q} \sqrt[n_S]{\prod_{s_i} ARR_{c_p,c_q}^{s_i}}}{M} \tag{3.5}$$

donde n_S representa el número de series y M es el número de modelos. El modelo con el mejor *ARR* es seleccionado, o, en un enfoque alternativo, los *ARRs* son utilizados para calcular pesos convexos, es decir pesos mayores o iguales a cero que sumados den 1, para los modelos considerados. El método de Lemke fija el número de grupos de series en 4 y muestra que la variación de éste número no afecta a su método.

3.4 Otros Trabajos

Respecto a la combinación de modelos de predicción utilizando herramientas de inteligencia computacional, podemos nombrar varios trabajos recientes. Rahman y colaboradores (Rahman, et al., 2015) proponen un método para construir ensambles de redes neuronales; con la intención de mantener la diversidad entre las redes seleccionadas, este método se basa en dos capas de procesamiento: la primera busca el número de valores de retardo requeridos para predecir el siguiente valor de la serie de tiempo. La segunda capa

crea redes neuronales con este parámetro y aumenta la diversidad generando diferentes conjuntos de entrenamiento. La predicción final es una combinación ponderada de los modelos seleccionados. Este método considera el concepto de diversidad, aunque solo trabaja con un tipo de modelo. Adhikari y Agrawal (2014) exploran la idea de utilizar modelos lineales para predecir la parte lineal de una serie y modelos no-lineales para predecir la parte remanente. Este método utiliza como modelo lineal el conocido como "caminata aleatoria" y como modelo no-lineal a las redes neuronales (Adhikari & Agrawal, 2014). Asimismo y en este sentido, algunos autores exploran la idea de utilizar modelos de diferente tipo combinando los modelos conocidos como "Auto-Regresivos Integrados de Medias Móviles" (ARIMA por sus siglas en inglés (Box, et al., 1994)), las máquinas de soporte de vectores (SVM por sus siglas en inglés) y las redes neuronales, ajustando el uso de promedio y mediana como estrategia de combinación (Adhikari & Agrawal, 2014). La línea de investigación de combinar modelos de diferente tipo a través de pesos diferentes define los pesos únicamente en función del desempeño previo de los modelos de predicción, en este sentido podemos citar (Adhikari & Verma, 2016) y (Adhikari, 2015). Los trabajos de ensambles recientes concluyen que incluir modelos de diferente tipo aumenta la diversidad; pero no consideran cómo regular entre diversidad y exactitud en la selección de los modelos.

3.4.1 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 siguen la estrategia de descomponer una tarea en sub-tareas; donde diferentes redes son entrenadas para realizar cada sub-tarea, esto es un módulo. La solución completa requiere la contribución de todos los módulos(Sharkey, 1999), basándose en el principio de divide y vencerás(Brassard & Pa, 1996). En este enfoque se pueden encontrar redes que utilizan arquitecturas fuertemente separadas, donde cada red neuronal es construida y entrenada para una sub-tarea específica. La decisión final se basa en los resultados de las redes individuales, también llamadas agentes(Melin, et al., 2007). Las ventajas de las redes modulares, tomadas de (Melin, et al., 2007), se listan 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 desde el enfoque de las redes modulares, tenemos dos tareas una es predecir la serie de tiempo y otra seleccionar los modelos de predicción; las tareas son resultas con diferentes tipos de redes neuronales, además para la tarea de predicción incluimos otros tipos de modelos de predicción.

3.4.2 Redes Difusas

El modelado basado en herramientas matemáticas tradicionales (por ejemplo ecuaciones diferenciales) no es muy apropiado para lidiar con sistemas inciertos o desconocidos por el contrario los sistemas de inferencia difusa utilizando reglas difusas del tipo "si-entonces" pueden modelar aspectos cualitativos del conocimiento humano y procesos de razonamiento (Melin, et al., 2012). El modelado difuso o identificación difusa fue por primera vez explorado en el trabajo de Takagi y Sugeno (1985); actualmente, se han desarrollado nuevos enfoques prometedores en la construcción de ensambles de sistemas de inferencia difusa basados en redes adaptivas (ANFIS, por sus siglas en inglés), por ejemplo podemos citar: (Melin, et al., 2007), (Melin, et al., 2012), (Pulido, et al., 2013) y (Gaxiola, et al., 2014).

Los trabajos basados en sistemas difusos reportan una disminución en el error de predicción, cuya mejora se debe a la arquitectura ANFIS (Melin, et al., 2012); es importante resaltar que al aplicar la variante de los modelos difusos, conocida como tipo 2 (Zadeh, 1975), se obtiene una mejor tolerancia al ruido en los datos en comparación con los modelos difusos tradicionales(Pulido, et al., 2013), (Gaxiola, et al., 2014). Con la finalidad de limitar el problema, en esta investigación no se utilizaron sistemas difusos, en su lugar se aplicaron búsquedas en profundidad en espacios restringidos; creemos que se podría mejorar los resultados si se consideran espacios de búsqueda de mayor tamaño y para manejarlos eficientemente incluir optimización basada en sistemas difusos. Adicionalmente, creemos que como trabajo futuro los métodos desarrollados en esta investigación podrían extenderse para ser tolerantes a ruido en los datos incluyendo sistemas difusos tipo 2. Para las personas que toman decisiones es muy importante poder

explicar los argumentos que motivan su decisión; como trabajo futuro también se podría explorar la capacidad de explicar los resultados aprovechando las reglas difusas encontradas.

3.5 Resumen de los Trabajos Relacionados

En el trabajo relacionado, varios autores coinciden en que los ensambles de modelos de predicción pueden disminuir el error de las predicciones comparado con utilizar solo un modelo. Sin embargo, construir una combinación de modelos de predicción capaz de cooperar eficazmente no es una tarea trivial. Existen diversos enfoques en la construcción de ensambles, en este capítulo se revisaron trabajos enfocados en meta-aprendizaje, aplicación de redes SOM y otros trabajos recientes.

El concepto de meta-aprendizaje puede aprovecharse para trabajar sobre diferentes series de tiempo y con diferentes tipos de modelos de predicción. El trabajo de Lemke y Gabrys (2010) aprovecha el conocimiento de experiencias con otras series para poder seleccionar un conjunto de modelos a combinar y como combinarlos; este trabajo es la línea base con la que se compara la presente investigación.

En los trabajos más recientes, la diversidad incrementa la exactitud de la combinación de modelos. Varios trabajos aumentan la diversidad añadiendo modelos de diferente tipo; sin embargo, no controlan la relación entre diversidad y exactitud al momento de seleccionar los modelos.

La capacidad de adaptación a los datos de las redes neuronales SOM ha sido ampliamente utilizada en la literatura, tanto en modelos de predicción basados en redes SOM para generar las predicciones, como en métodos de construcción de ensambles de modelos de predicción. Nuestra propuesta aprovecha la red neuronal SOM y el concepto de meta-aprendizaje para la construcción de ensambles aplicable en diferentes series, guiando la selección de los modelos a partir de la diversidad y la exactitud de los modelos de predicción.

Capítulo 4. Métodos Propuestos

Varios dominios trabajan con sistemas complejos y requieren predicciones a largo plazo; pero, el costo en dinero, tiempo y recursos de recurrir con un experto en el funcionamiento de los diferentes modelos de predicción puede ser alto. Los expertos analizan el comportamiento de los modelos con la serie de tiempo y recomiendan un modelo o un ensamble de modelos para realizar la predicción. En esta investigación revisamos un extenso conjunto de modelos de predicción, y los resultados sugieren que la exactitud de un ensamble de modelos puede superar a la de cualquiera de los modelos de predicción disponibles para construir el ensamble. Sin embargo, definir un conjunto de modelos que puedan cooperar no es una tarea trivial, en especial cuando el comportamiento de la serie de tiempo cambia bruscamente. En varios trabajos, como por ejemplo: (Makridakis & Hibon, 2000)(Crone, et al., 2008), los expertos analizan el desempeño de los modelos guiándose por tres ventanas conocidas como corto, mediano y largo plazo. El primer método que proponemos analiza el desempeño de los modelos en tres ventanas de predicción, luego selecciona y combina los modelos en función del análisis previo.

En la construcción de un ensamble, la búsqueda de modelos a combinar puede considerar el teorema de descomposición en sesgo, varianza y covarianza de los modelos disponibles(Ueda & Nakano, 1996). El segundo método propuesto en esta investigación analiza el sesgo de los modelos de predicción utilizando una métrica de desempeño y un método de validación, con la finalidad de disminuir la varianza. Proponemos una meta-característica que permite analizar el comportamiento de los modelos en diferentes instantes de tiempo, cada uno bajo las mismas condiciones; al igual que con la covarianza se busca modelos con una relación inversamente proporcional. Es muy importante no llegar al extremo de máxima diversidad, es decir, un comportamiento completamente diferente, porque este comportamiento está asociado con una diminución considerable del desempeño.

En este capítulo se describen los métodos propuestos en este proyecto de investigación. La primera sección contiene el diagrama general del esquema de predicción utilizado. La sección 4.2 describe el método extractor de meta-datos y la sección 4.3 describe el método basado en ventanas de predicción. La sección 4.3 describe el error representativo, el cual es utilizado por el método SOMSM descrito en la sección 4.3. La sección 4.5 describe el análisis de complejidad de los métodos propuestos y finalmente se incluye un resumen de los métodos propuestos.

4.1 Diagrama General del Esquema de Predicción

Algunos sistemas complejos pueden verse como sistemas deterministas afectados por sistemas ocultos, esto implica que se requiera más de un modelo para estimar las predicciones. Antes de seleccionar que modelos utilizar, el primer paso es analizar el comportamiento de todos los modelos con la serie de tiempo. El análisis de los modelos de predicción requiere ser flexible con la finalidad de poder aplicarlo con cualquier tipo de modelo; dado que el conjunto de modelos de predicción puede ser extenso se necesita que se puedan definir diferentes estrategias de búsqueda. La Figura 4.1 muestra la estrategia general que siguen los métodos propuestos en esta investigación para la construcción de un ensamble de modelos de predicción, la estrategia está compuesta de dos pasos principales: meta-aprendizaje del ensamble y ejecución del ensamble; ambos pasos se realizan de manera completamente automática, esto es, sin la intervención de expertos humanos. En el primer paso, el proceso de *meta-aprendizaje* analiza el comportamiento de los modelos de predicción con la serie de tiempo; a partir de los resultados del análisis se construye un ensamble, el cual combina los modelos seleccionados con la estrategia de combinación establecida para predecir la serie de tiempo. Se desarrollaron dos estrategias de selección y combinación de modelos de predicción; los métodos desarrollados ajustan sus parámetros en una secuencia reservada para validación. En el segundo paso, el ensamble construido genera la predicción de la serie de tiempo.

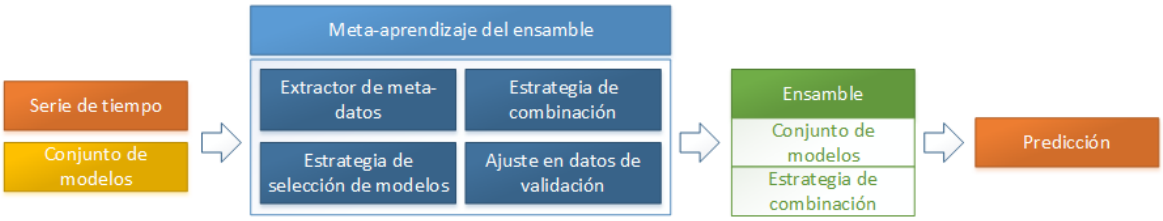


Figura 4.1. Proceso propuesto de construcción de un ensamble para la predicción de una serie de tiempo particular.

4.2 Método Extractor de Meta-Datos

La primera etapa de este método consiste en extraer las meta-características seleccionadas de los modelos de predicción y de la serie de tiempo. El extractor recibe un conjunto de tipos de modelos (llamados modelos base), con sus respectivos parámetros y los posibles valores que pueden tomar. Las diferentes combinaciones de modelos que se pueden obtener variando el modelo base y los valores de los parámetros conforman el espacio de búsqueda. El tamaño del espacio de búsqueda se limita a partir de un conjunto de modelos base con $\mathcal{B}$ elementos, cada modelo base b tiene un conjunto de $\mathcal{P}_b$ parámetros y cada parámetro puede tener v_p valores posibles. El número total de modelos posibles (M) puede estimarse por la siguiente ecuación.

$$M = \sum_{b=1}^{\mathcal{B}} \sum_{p=1}^{\mathcal{P}_b} v_p \tag{4.1}$$

Dado que el espacio de búsqueda crece en múltiplos dependiendo de $\mathcal{B}$, $\mathcal{P}_b$ y v_p, se requiere una estrategia de búsqueda. En esta investigación limitamos el tamaño del espacio de búsqueda para poder realizar una búsqueda completa, siguiendo una estrategia de primero en profundidad; esto garantiza una solución óptima dentro del espacio de búsqueda definido.

El extractor de meta-datos recibe la serie de tiempo y analiza los modelos disponibles para construir las meta-características, las cuales están relacionadas con la exactitud de cada modelo con la serie de tiempo involucrada. Estas meta-características se obtienen

siguiendo un proceso de validación cruzada Monte Carlo (MCCV por sus siglas en inglés), lo que permite estimar la exactitud real de cada modelo.

Existen varias métricas para evaluar la exactitud de predicción de un modelo con una serie de tiempo, aunque, no hay una sola que sea recomendada para todos los escenarios posibles. Para los métodos propuestos, escogimos evaluar la exactitud de los modelos de predicción utilizando, una medida basada en el porcentaje de error, conocida como "Promedio Simétrico del Porcentaje de Error Absoluto" (SMAPE por sus siglas en inglés). La principal ventaja de SMAPE es que devuelve valores en un rango definido de 0 y 200, donde 0 es una predicción sin error y 200 la peor predicción posible.

La predicción de MCCV sigue dos consideraciones: primero, el proceso de evaluación debe estar enfocado en la tarea en la que será utilizado el modelo (Picard & Cook, 1984), que en este contexto es el estimar h valores consecutivos. Por esta razón, se realiza una validación basada en bloques como se recomienda en (Bergmeir & Benitez, 2012). Segundo, considerando la sensibilidad a los cambios en las condiciones iniciales, se toman en cuenta los valores iniciales en todas las evaluaciones. De forma general, MCCV consiste en realizar un número determinado k de entrenamientos y pruebas del modelo, usando segmentos de la serie generados de forma aleatoria dentro de un rango determinado. Estos segmentos se definen de acuerdo a la siguiente ecuación:

$$\begin{aligned} Y^{tr(j)} &= \{y_t \in Y | 1 \leq t \leq r(j)\}, \\ Y^{test(j)} &= \{y_t \in Y | r(j) + 1 \leq t \leq r(j) + h\} \end{aligned} \tag{4.2}$$

donde Y es la serie de entrada, $Y^{tr(j)}$ es una nueva secuencia de entrenamiento, tomada de la serie de tiempo Y para la iteración j; $Y^{test(j)}$ es la secuencia de prueba respectiva de los siguientes h valores; $r(j)$ depende de un valor aleatorio y se define como:

$$r(j) = r_0 + \text{random}[0, N - h] \tag{4.3}$$

r_0 es un número fijo de valores que son requeridos como mínimo para entrenar un modelo y N es el tamaño de la serie Y. Un modelo de predicción es entrenado con $Y^{tr(j)}$, el modelo entrenado estima los siguientes h valores, los cuales son evaluados utilizando SMAPE y la

secuencia $Y^{test(j)}$. Este entrenamiento y evaluación se repite con k parejas de secuencias, MCCV retorna el promedio de las evaluaciones en todas las iteraciones (ver Anexo H.4).

Los tipos de modelos de predicción que se pueden aplicar depende del dominio en el que se trabaje; con la finalidad de alcanzar flexibilidad para diferentes dominios, el diseño del ensamble requiere soportar en el proceso de evaluación y selección, diferentes tipos de modelos de predicción con diferentes parámetros y diferentes conjuntos de posibles valores para cada parámetro. Una opción es crear una sola función principal que se adapte a todos los cambios del sistema, pero con cada nuevo tipo de 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 cambiar el comportamiento de los demás componentes. Por esta razón, seleccionamos la estrategia de envolturas para poder aumentar tipos de modelos y probar con diferentes estrategias de búsqueda. La Figura 4.2 muestra el método de extracción de meta-características que inicia con el componente validador, de tipo MCCV, donde la función validar comienza dividiendo la serie en dos sub-series: una para entrenamiento, llamada $Y^{tr(\cdot)}$, y una serie para prueba, identificado como $Y^{test(\cdot)}$. La estrategia de búsqueda devuelve el siguiente modelo a analizar, el modelo pertenece a un tipo de modelo y tiene los valores definidos para sus respectivos parámetros. El componente evaluador entrena cada modelo por medio de la respectiva envoltura; luego el modelo entrenado predice los siguientes h valores de la serie. La predicción obtenida es evaluada con la predicción esperada $Y^{test(\cdot)}$; el resultado de cada modelo es registrado en un motor de base de datos (BDD). Es importante aclarar que para los experimentos de esta investigación el método de extracción de meta-características utiliza únicamente la secuencia de entrenamiento de cada serie de tiempo; cuando se aplique en un escenario real utilizará toda la serie disponible.

Los registros de las predicciones y evaluaciones, devueltos por el extractor de meta-datos, permiten extraer diferentes características de desempeño de los modelos ya sea considerando todo el horizonte de predicción, secciones del horizonte de predicción o cada instante de tiempo.

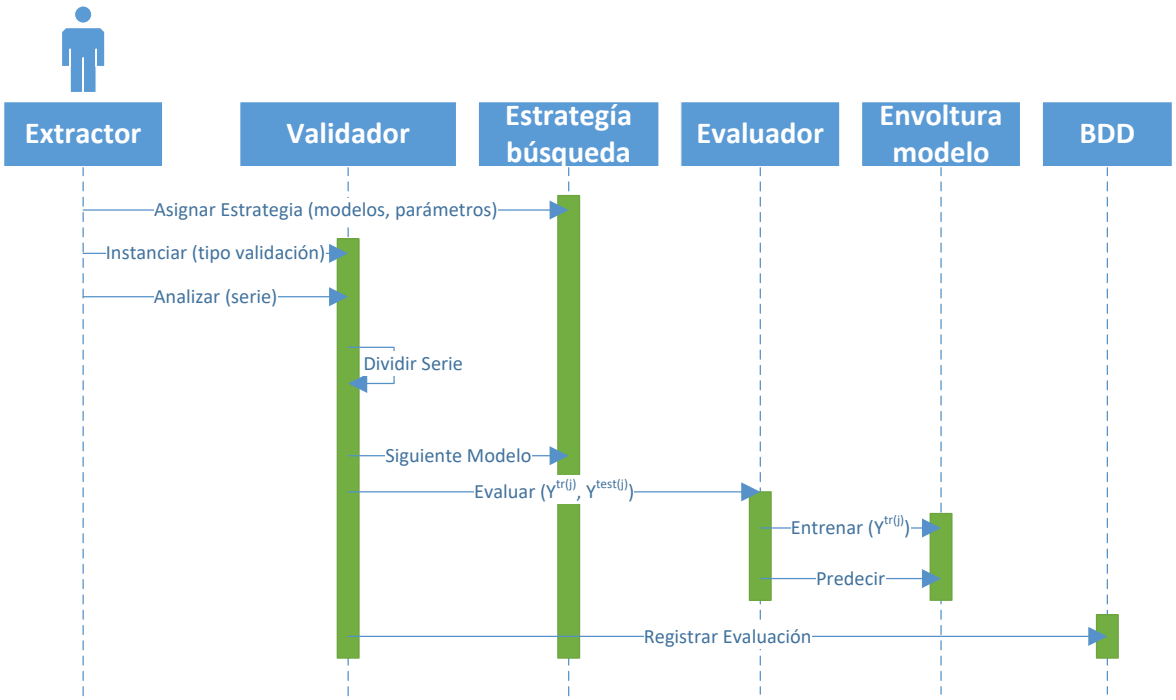


Figura 4.2. Diseño del extractor de meta-características. La entrada es un conjunto de modelos con sus posibles parámetros y una serie de tiempo. La salida es un conjunto de evaluaciones registradas en una base de datos (BDD).

4.3 Combinación Temporal Validada

Como nombramos anteriormente, dado que por la naturaleza de las series de tiempo el comportamiento de las series varía con el horizonte de predicción; y por lo tanto, el desempeño de los modelos de predicción varía con el horizonte de predicción. Un método que combine diferentes modelos en diferentes instantes de predicción puede obtener mejores resultados que solo combinando diferentes modelos sin considerar la naturaleza del horizonte. El método propuesto que se describe a continuación, denominado "Combinación Temporal Validada" (TVC, por sus siglas en inglés) divide el horizonte de predicción en tres secciones o ventanas de predicción, cada una con el mismo número de elementos, e identificadas como: corto, mediano y largo plazo; esta división asemeja al trabajo realizado por expertos humanos al analizar el desempeño de los modelos(Crone, et al., 2008), (Makridakis & Hibon, 2000). El 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 4.3. TVC selecciona los modelos de acuerdo a su exactitud en cada ventana de predicción, que es validada con MCCV. La predicción del ensamble correspondiente al corto plazo es calculada a través de la combinación de las predicciones de los *k*-modelos con mejor exactitud en la ventana de corto plazo. De igual manera, se obtienen las predicciones a mediano y largo plazo, al terminar todas las predicciones por ventana son concatenadas para obtener la predicción final.

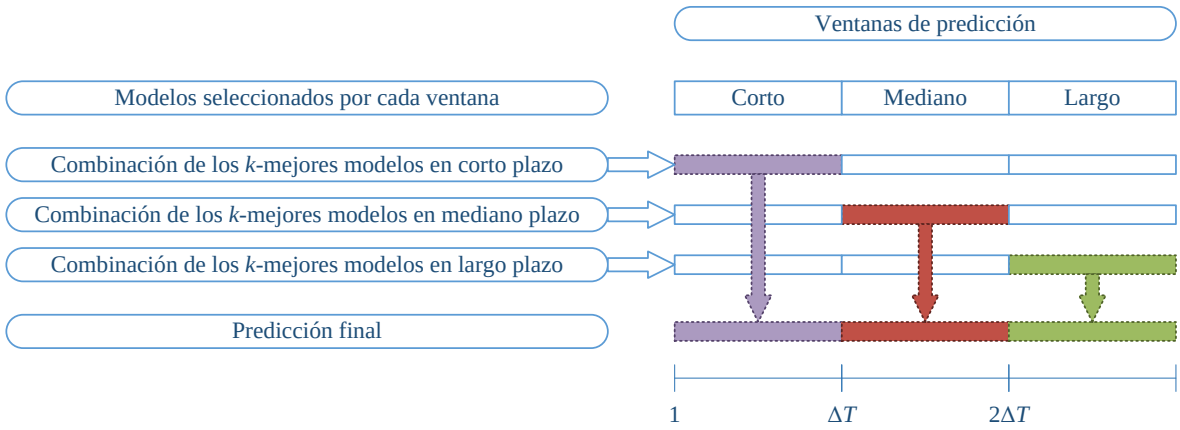


Figura 4.3. Combinación Temporal Validada de los modelos de predicción.

El extractor de meta-datos descrito en la sección 4.2 permite obtener tres evaluaciones de exactitud de cada modelo de predicción, cada una correspondiendo a una ventana de predicción. En cada iteración de MCCV, la serie de prueba es dividida en tres ventanas de predicción: corto, mediano y largo plazo; por su parte cada modelo retorna una secuencia de predicción de tamaño igual a h, que a su vez es dividida en tres ventanas. La función SMAPE compara la predicción obtenida y la esperada por cada ventana de predicción. La evaluación de exactitud por cada ventana es el promedio de los SMAPE's obtenidos en todas las iteraciones del MCCV. Al final, cada modelo tiene tres evaluaciones de exactitud, correspondientes a las ventanas de predicción de corto, mediano y largo plazo respectivamente.

Los modelos seleccionados para corto plazo son aquellos que tengan el menor valor SMAPE promedio en la ventana de corto plazo; el mismo procedimiento es seguido para seleccionar los modelos en las ventanas de mediano y largo plazo. Una vez definido el

conjunto de modelos a combinar, éstos son entrenados con la serie de entrenamiento completa, las predicciones de los modelos seleccionados para corto plazo son combinadas utilizando como funciones de combinación la mediana o el promedio de las predicciones de los modelos. El proceso es repetido con las ventanas de mediano y largo plazo. Las predicciones combinadas de los modelos seleccionados de corto, mediano y largo plazo son representadas como $\{y^s\}$, $\{y^m\}$, y $\{y^l\}$, respectivamente. Las ventanas corto, mediano y largo plazo tienen tamaño ΔT; si h no se puede dividir exactamente en 3, entonces las primeras dos ventanas utilizan un $\Delta T = \lfloor h/3 \rfloor$, y la última ventana es de mayor tamaño hasta completar h. La predicción del ensamble $\hat{Y}$ es obtenida al concatenar las tres ventanas de predicción. La concatenación está expresada en la siguiente ecuación:

$$\hat{Y} = \left\{ \{y_i^s, N+1 \leq i \leq N+\Delta T\} \cup \{y_i^m, N+\Delta T+1 \leq i \leq N+2\Delta T\} \right. \\ \left. \cup \{y_i^l, N+2\Delta T+1 \leq i \leq N+h\} \right\} \tag{4.4}$$

TVC requiere dos parámetros: el número k de modelos a combinar por cada ventana y la función de combinación de las predicciones de los modelos seleccionados, que por defecto es mediana, siendo una alternativa el promedio simple. En el capítulo 5 se muestra experimentalmente que la estrategia de analizar la exactitud de los modelos en el tiempo en función de ventanas de predicción, permite a TVC tener un mejor desempeño que seleccionar el modelo con mejor SMAPE considerando todo el horizonte de predicción.

4.4 Error Representativo

El error representativo es una meta-característica propuesta en esta investigación, que representa el desempeño de los modelos con la serie en cada instante de tiempo; la cual permite analizar la diversidad de comportamiento entre modelos de predicción, antes de la creación del ensamble. El teorema de descomposición sesgo, varianza y covarianza(Ueda & Nakano, 1996) recomienda combinar modelos con una covarianza negativa, es decir una relación lineal inversamente proporcional, esto relaciona la diversidad entre modelos con un comportamiento diferente en las mismas condiciones. Entre los trabajos que utilizan el concepto de diversidad en la creación de ensambles de modelos de predicción se puede citar: (Brown, et al., 2005)(Rahman, et al., 2015)(Mendes-Moreira, et al., 2012), entre

otros; estos trabajos construyen ensambles con diferentes subconjuntos de modelos, combinan las predicciones y evalúan la predicción obtenida, la diferencia entre las evaluaciones de los ensambles construidos es utilizada como medida de diversidad. El error representativo busca analizar la diversidad a través de la diferencia de comportamiento de los modelos de predicción previo la construcción del ensamble.

El error representativo extiende la idea de TVC analizando el desempeño de los modelos de predicción en cada instante de tiempo desde t_1 hasta t_h; la Figura 4.4 muestra el proceso para calcularlo. Primero, el modelo es entrenado utilizando *N* - *h* observaciones de la serie y posteriormente predice los siguientes *h* valores. A continuación, el error representativo es calculado como un vector de error de cada instante de tiempo, es decir, la diferencia entre cada valor predicho y el valor esperado en las últimas *h* observaciones.

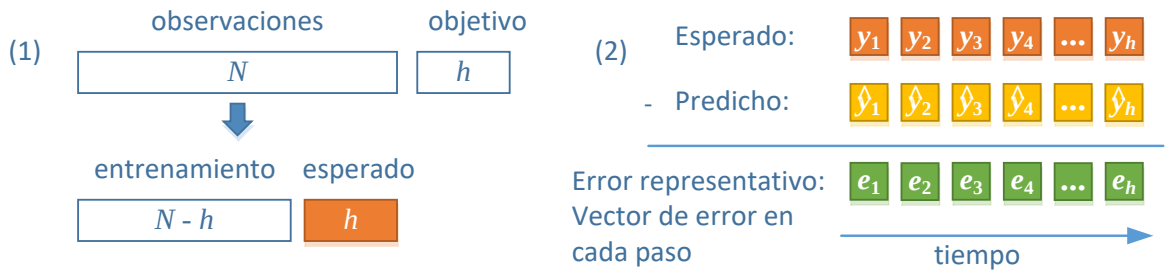


Figura 4.4. Cálculo del error representativo para horizontes mayores a uno

Los modelos con un error representativo similar tienen un comportamiento similar en cada unidad de tiempo. La Figura 4.5 simboliza como podrían agruparse los modelos para un horizonte de predicción de 2 pasos; los ejes corresponden con el error obtenido en los tiempos 1 y 2 respectivamente; los pequeños símbolos de diferentes colores presentan a modelos que tienen un error semejante en cada instante de tiempo y los hexágonos muestran los grupos de modelos que se formarían en ese ejemplo.

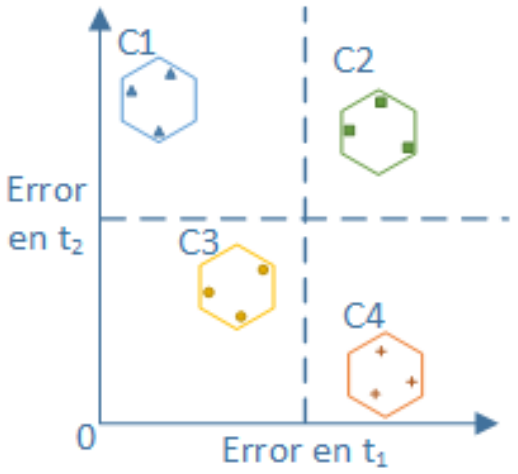


Figura 4.5. Ejemplo de agrupamiento de modelos, utilizando el error representativo con un horizonte igual a 2. Cada símbolo representa un modelo de predicción y los hexágonos representan grupos de modelos.

Un punto importante a considerar es que algunos modelos base pueden obtener diferentes resultados usando el mismo conjunto de entrenamiento; esto sucede por ejemplo con las redes neuronales, debido a la inicialización aleatoria de sus pesos. En estos casos el cálculo del error representativo se repite *k* veces y el vector final será el promedio de los *k* vectores obtenidos en cada repetición; el valor de *k* lo define el usuario, en esta investigación utilizamos $k = 10$. El vector de error representativo permite comparar el desempeño en cada instante de tiempo entre modelos diferentes entrenados con el mismo conjunto de entrenamiento y comparados con respecto a una misma secuencia de prueba.

4.5 Selección de Modelos Utilizando Redes SOM

En la ecuación (2.29) se muestra que la selección de modelos para construir un ensamble requiere considerar el sesgo, la varianza y la covarianza entre los modelos. El método propuesto, denominado SOMSM (*Self Organizing Map for Selection of Models*), permite manejar el sesgo y la varianza entre los modelos, utilizando la estimación de exactitud validada con MCCV; mientras que el error representativo, a semejanza de la covarianza, permite analizar la diversidad de comportamiento entre los modelos. SOMSM recibe un conjunto de posibles modelos de predicción y una serie de tiempo; el objetivo es producir *h* valores de predicción. La Figura 4.6 muestra los pasos principales de SOMSM,

el cual inicia analizando el comportamiento de cada modelo para extraer meta-características. Un mapa auto-organizado de dos dimensiones es creado con las meta-características y usando este mapa, se selecciona de forma automática un subconjunto de modelos considerando su diversidad y exactitud; para la selección se desarrollaron diversas estrategias que se describen más adelante. Posteriormente, se depura esta selección de los modelos, utilizando el último segmento de la serie de tamaño igual al horizonte de predicción objetivo, como recomiendan(Bergmeir & Benitez, 2012), (Picard & Cook, 1984). Los modelos seleccionados son entrenados con la secuencia de entrenamiento y sus predicciones son combinadas para obtener la predicción final como se realiza en diferentes marcos de evaluación de predicción a largo plazo(Crone, et al., 2008),(Makridakis & Hibon, 2000). A continuación, se detalla cada parte del método propuesto.

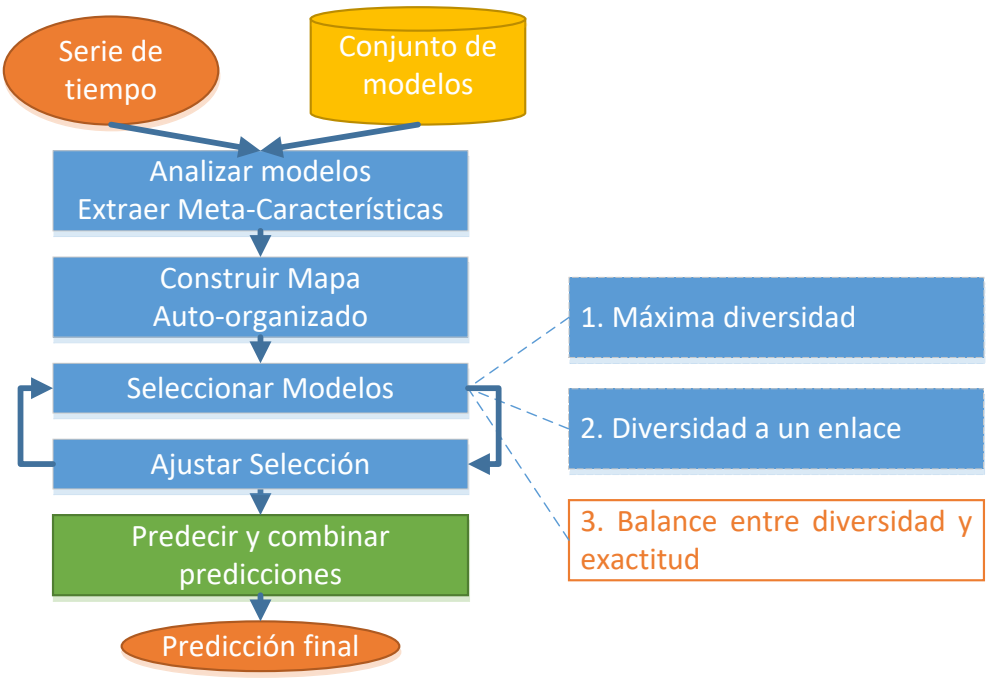


Figura 4.6. Método propuesto SOMSM, recibe una serie de tiempo y un conjunto de modelos, define un ensamble considerando diversas estrategias. El ensamble retorna la predicción final.

4.5.1 Construcción de Mapa Auto-Organizado SOM

Los errores representativos de cada modelo son ingresados a una red SOM para agrupar a los modelos de acuerdo a su comportamiento en el tiempo. A diferencia del método TVC que analiza el desempeño en diferentes ventanas, SOMSM utiliza los vectores de error representativo para analizar el comportamiento de los modelos en cada instante de tiempo dentro de una sola secuencia de tamaño igual al horizonte de predicción. Como se describió en la sección 3.2, las redes SOM permiten identificar grupos, con la peculiaridad de que los elementos en un grupo son similares a los elementos en un grupo vecino en el mapa de salida. En este trabajo se utiliza una red SOM con un mapa de salida de dos dimensiones definido por el número de filas y columnas; la neurona de entrada se conecta con todas las neuronas del mapa de salida, que siguen una estructura hexagonal. Las neuronas tienen asociado un vector de pesos, que representa al prototipo de un grupo. El entrenamiento de la red SOM es no supervisado, itera un número fijo de veces y sigue dos principios definidos: competir y cooperar. Ante cada vector de entrada las neuronas compiten analizando cuál de ellas se parece más a la entrada, utilizando una distancia Euclidiana. Los pesos de la neurona ganadora son modificados con respecto al vector de entrada, cooperando con las neuronas en su vecindario. El método propuesto utiliza el índice VRC (ver sección 2.4) para escoger el número de grupos (neuronas) adecuados para la red SOM. Para esto, utilizando los errores representativos de cada modelo, una red SOM es entrenada por cada uno de los posibles tamaños (número de neuronas) definidos por el usuario, y se selecciona la red SOM que maximiza el índice VRC.

La Figura 4.7 muestra un ejemplo hipotético de agrupar tres modelos utilizando sus respectivos vectores de error representativo y una red SOM. En la parte izquierda de la figura (a) se puede observar el comportamiento de los modelos reflejado por el error representativo a lo largo del tiempo; en la parte derecha (b) se aprecia la distribución de los modelos utilizando una red SOM de 3 filas y 4 columnas. En dicha figura puede apreciarse el cómo los modelos 1 y 2 tienen un comportamiento similar y fueron agrupados en neuronas vecinas, mientras que el modelo 3 con un comportamiento diferente fue asignado a una neurona lejana con respecto a las neuronas que contienen a los modelos 1 y 2. Entre

más lejanas estén las neuronas, unas de otras, mayor será la diversidad entre los modelos de predicción agrupados en ellas.

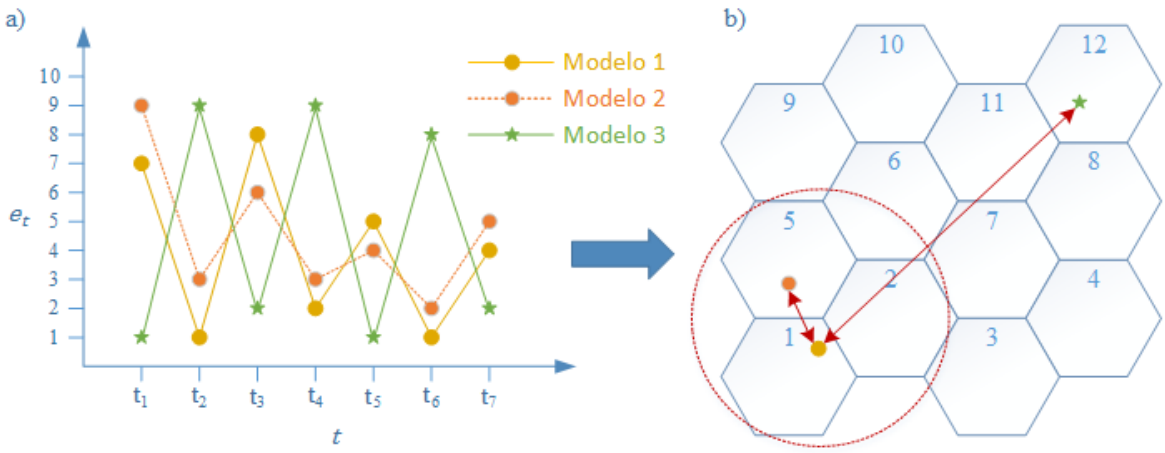


Figura 4.7. Ejemplo de agrupamiento de modelos por su error representativo utilizando una red SOM. Cada símbolo representa un modelo de predicción y los hexágonos representan grupos de modelos.

4.5.2 Selección de Modelos

Una vez agrupados los modelos en el mapa, se seleccionan algunos para construir el ensamble. Este proceso inicia seleccionando el modelo, denotado por *b*, que presenta la mejor estimación de exactitud, denotada como q_b; el grupo (neurona) que lo contiene se denota como *A*. Como se mencionó en la sección 2.6, la diversidad de comportamiento entre los modelos permite mejorar el desempeño del ensamble, teorema sesgo-varianza-covarianza (Ueda & Nakano, 1996). Con la finalidad de buscar diversidad entre los modelos, se define un vecindario circular alrededor de *A*; los modelos en neuronas vecinas tendrán un error representativo diferente al de *b*, lo suficiente para pertenecer a otro grupo pero lo suficientemente similar para tener un buen desempeño, dado que un comportamiento similar está asociado con un error similar. El radio del vecindario se determina con *l*-enlaces a partir de *A*, donde *l* es el número mínimo de saltos en el mapa que se tienen que dar para conectar cualquier neurona con *A*; el resultado es un vecindario de neurona alrededor de la neurona *A*.

Con la finalidad de forzar la diversidad en la selección de modelos, una vez seleccionado el vecindario de neuronas incluyendo la neurona *A*, se selecciona el *p*% de los modelos con mejor exactitud estimada de cada neurona (grupo de modelos); los cuales conforman una lista preliminar de modelos. Para no seleccionar modelos con una exactitud estimada demasiado baja, se aplica un criterio de poda a la lista preliminar. Este criterio utiliza un factor de tolerancia α, y consiste en seleccionar los modelos que tengan una estimación de exactitud menor o igual a $\alpha \times q_b$. Un valor α cercano a 1 indica que se seleccionen modelos con una estimación de exactitud cercana a la del mejor modelo, es decir se hace énfasis en la exactitud al seleccionar los modelos; mientras que si α es cercano 0, se permite mayor tolerancia de error y se seleccionan modelos provenientes de las neuronas vecinas favoreciendo la diversidad entre ellos. Por lo tanto, el parámetro α permite balancear entre exactitud y diversidad en la función seleccionar-modelos.

Asimismo, se propusieron dos estrategias más para seleccionar los modelos:

1. Maximizar diversidad: se seleccionan los modelos agrupados en la neurona más lejana con respecto a la neurona *A* en el mapa utilizando la función de distancia Euclidiana.
2. Diversidad en un vecindario cercano: se seleccionan los modelos de las neuronas dentro de un vecindario de un-enlace a partir de *A*. Se incluyen todas las neuronas que se puedan conectar en un salto con *A*.

En los experimentos reportados en el Capítulo 5 se observó que los resultados de la estrategia de balancear entre exactitud y diversidad superaron a los obtenidos por las otras dos estrategias.

4.5.3 Ajustar Selección de Modelos

Dado que el método SOMSM requiere definir de antemano los parámetros correspondientes a: *l* número de enlaces para definir el vecindario, *p%* de modelos a seleccionar y α tolerancia, se incluye un proceso automático para ajustar estos valores; éste proceso utiliza una secuencia de validación tomada de los últimos *h* valores de la serie de tiempo, los modelos son entrenados con *N* - *h* observaciones, un ejemplo de la división de

realizada para obtener la secuencia de validación puede verse en la Figura 4.4 - 1. El método SOMSM (*l*, *p%*, α) devuelve una predicción que es evaluada con SMAPE y la secuencia de validación; dado un conjunto de posibles valores para cada parámetro (*l*, *p%*, α), se evalúan todas las combinaciones y se selecciona el conjunto de parámetros con los que SOMSM devuelve el mínimo SMAPE en la secuencia de validación. Los conjuntos de posibles valores para los parámetros son definidos por el usuario.

4.5.4 Combinación y Predicción Final

Los modelos seleccionados son entrenados con la serie completa y cada modelo predice los siguientes *h* valores. Las predicciones de todos los modelos seleccionados son combinadas con las funciones promedio o mediana y el vector resultante corresponde a la predicción final.

4.5.5 Algoritmo Completo y Análisis de Complejidad

La Figura 4.8 muestra el algoritmo de construcción de ensamble seguido por SOMSM y descrito en las secciones anteriores de este capítulo. El algoritmo tiene tres partes: (1) extracción de meta-características, (2) construcción del mapa y (3) selección de modelos aplicando poda.

El costo computacional del método propuesto incluye la complejidad de las fases de entrenar (*tr*) y predecir (*fr*) un modelo particular; estos valores dependen del número de observaciones *N*, del horizonte de predicción *h* y de la funcionalidad de cada modelo involucrado. La complejidad computacional de cada parte del algoritmo de construcción de ensambles del método SOMSM es la siguiente:

```
Función construir ensamble SOMSM
Entradas
   Y: Serie de tiempo
   C_M: Conjunto de modelos candidato
   l: Número de enlaces para definir el vecindario
   p: Porcentaje de modelos a seleccionar
   α: Factor de tolerancia de error
   S: Conjunto de posibles tamaños de SOM
Retorna: Lista de modelos del ensamble
Extracción de meta-características de cada modelo c
   q_c: Estimación de exactitud de un modelo c
   e_c: Error representativo de un modelo c
   for all c ∈ C_M do
     q_c = estimar exactitud con MCCV (Y, c)
     e_c = calcular error representativo (Y, c)
   end for
Construcción del mapa para definir el vecindario de modelos
   mapa = SOM con máximo VRC ({r_c|∀c ∈ C_M}, S)
   q_b = min ({q_c|∀c ∈ C_M})
   b ∈ C_M, el modelo b tienen la mejor estimación de exactitud (q_b)
   A = mapa->buscar neurona que contenga al modelo (b)
   G = mapa->definir vecindario de neuronas (A, l)
Selección de modelos podando con el factor de tolerancia de error α
   Γ = { } : Lista de modelos del ensamble
   for all N_a ∈ G do : Por cada neurona N_a en G
     N_p = seleccionar porcentaje de modelos de neurona (N_a, p)
     for all c ∈ N_p do
       if q_c ≤ α × q_b then
         Γ = Γ ∪ {c}
       end if
     end for
   end for
```

Figura 4.8. Algoritmo de selección de modelos utilizando un mapa SOM.

1. Extracción de meta-características de los modelos con la serie de tiempo. La estimación de exactitud utilizando MCCV y el cálculo del error representativo requieren un número de operaciones acodado superiormente por $O\big(k \times (tr + fr + h)\big)$, donde k es el número de iteraciones, y h corresponde al número de operaciones requeridas para evaluar los h valores predichos. Los valores de tr y fr son mucho mayores que h, siguiendo la regla del máximo (Brassard & Bratley, 1996) y asumiendo un número total de modelos M, la complejidad de extraer las dos meta-características está acotada superiormente por $O\big(2 \times M \times k \times (tr + fr)\big)$.

2. Construcción del mapa SOM a partir de las meta-características. Ésta inicia encontrando el mejor agrupamiento, variando el número de neuronas de la red SOM en un rango S pre-definido de posibles tamaños. Un entrenamiento de la red SOM está acotado superiormente por $O(M \times n_N \times h \times E)$, donde E es el número épocas y n_N es el número de neuronas de la red SOM. Cada SOM realiza un agrupamiento de modelos, éste es evaluado con el índice VRC. Tomando en cuenta que la complejidad de entrenamiento de la red SOM es mucho mayor que el cómputo del índice VRC, entonces, la complejidad de encontrar el mejor agrupamiento está limitada superiormente por $O(M \times |S| \times \max(S) \times h \times E)$, donde $|S|$ es el número de posibles tamaños de la red SOM, y $\max(S)$ es el mayor tamaño disponible. El siguiente paso dentro de la construcción del mapa es la selección de la neurona que contiene el modelo con la mejor estimación de exactitud (neurona A) y la definición del vecindario de neuronas; este paso demanda un número de operaciones acotado superiormente por $O(M + \max(S))$. Dado que encontrar el mejor agrupamiento requiere muchas más operaciones que la selección de la neurona A y su vecindario, la complejidad de la construcción del mapa SOM está acotada superiormente por $O(M \times |S| \times \max(S) \times h \times E)$.
3. Selección de modelos de la neurona A y su vecindario de neuronas. Ésta requiere un número de operaciones limitado superiormente por $O(M \times \max(S))$.

La complejidad de la extracción de meta-características $O\big(2 \times M \times k \times (tr + fr)\big)$ es mucho mayor que la complejidad de las otras partes, siguiendo la regla del máximo dicha complejidad define la cota superior del método propuesto. Un número mayor de modelos M aumenta la diversidad de entre modelos y permite aumentar las posibilidades de encontrar mejores combinaciones; sin embargo M afecta directamente a la complejidad de método propuesto; éste es un compromiso entre exactitud y eficacia del método propuesto.

El número de operaciones de tr y fr dependen de cada modelo base; a continuación, se muestran los órdenes de complejidad de acuerdo a los modelos base utilizados en este proyecto.

kTrayectorias más cercanas.- Depende de m elementos de retardo o tamaño de la secuencia, k número de vecinos y el tamaño de la secuencia de entrenamiento N; el número de secuencias posibles a partir de la serie de entrenamiento es de $N - m + 1$, dado que $N \gg m$ por simplicidad utilizamos N como el número de secuencias posibles. Para agilizar la búsqueda de las secuencias de tamaño m se construye un índice basado en un árbol de ordenamiento para los N elementos. El orden de complejidad de tr está acotado superiormente por $O(N \times \log(N) \times m)$. Al predecir se buscan las k secuencias de tamaño m más semejantes a la secuencia de entrada, el orden de complejidad de fr está acotado superiormente por $O(k \times N \times m)$.

NARX.- Es una red neuronal que depende del número de neuronas de retraso m, el número de neuronas en la capa oculta n_H y el número de secuencias posibles de tamaño m, al igual que en el análisis de kTrayectorias por simplicidad utilizamos N como el número de secuencias posibles. El número de operaciones depende del algoritmo de entrenamiento utilizado y los datos disponibles; dado que en general los algoritmos de entrenamiento no pueden garantizar la finalización de la ejecución, estos incluyen un número máximo de épocas M_E. En una época se exhiben todas las secuencias posibles a la función de adaptación de pesos; si el número de épocas es igual a M_E finaliza el proceso de entrenamiento. El orden de complejidad de tr está acotado superiormente por $O((m \times n_H)^2 \times N \times M_E)$. El proceso de predicción se limita superiormente por $O(m \times n_H)$.

ARIMA.- En proceso de entrenamiento elabora un conjunto de ecuaciones, cuyo número de variables depende de los términos (p, d, q), los valores de las variables son aproximados a través del método de Programación Cuadrática Secuencial (SQP por sus siglas en inglés). El orden de complejidad de tr depende del número de operaciones que demanda SQP, la cual está acotado superiormente por $O(N^3)$, la demostración respectiva está disponible en: http://www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf. El proceso de predicción está acotado superiormente por $O(p \times d \times q)$.

En los métodos base utilizados en esta investigación, tr requiere más operaciones que fr. Entre los tres métodos base, el tr que requiere mayor número de operaciones es el de NARX, re-emplazando el orden de complejidad del tr de NARX en el orden de

complejidad del método propuesto se tiene que éste está acotado superiormente por $O(2 \times M \times k \times (m \times n_H)^2 \times N \times M_E)$.

4.6 Resumen de Métodos Propuestos

Como resultado de esta investigación se propusieron dos métodos de predicción basados en ensambles y meta-características. La selección de los modelos que constituyen el ensamble parte de la estimación de exactitud, la cual es validada utilizando MCCV; dado que esta estimación relaciona la exactitud de un modelo con la serie analizada la consideramos una meta-característica. En esta investigación se propone una nueva meta-característica, denominada vector de error representativo, la que muestra el comportamiento en el tiempo de cada modelo de predicción. Los métodos propuestos utilizan estas méta-características para guiar la construcción de ensambles, considerando exactitud y diversidad del comportamiento de los modelos involucrados.

En la revisión de la literatura del capítulo 3 encontramos que el desempeño de los modelos de predicción varía según el horizonte de predicción, de aquí que varios investigadores utilizan tres ventanas de predicción para analizar el comportamiento de los modelos. A partir de esta observación se propuso el método denominado TVC; este método, utilizando MCCV, estima la exactitud de los modelos en tres ventanas de predicción corto, mediano y largo plazo. TVC selecciona conjuntos de modelos para cada ventana de predicción a partir de su estimación de exactitud en cada ventana; la predicción devuelta es la concatenación del promedio de las predicciones de los modelos seleccionador por cada ventana.

En la construcción de ensambles dos factores a tomar en cuenta son el desempeño y la diversidad de los modelos de predicción; considerar estos factores para cualquier tipo de modelo y cualquier tipo de serie de tiempo impone retos no triviales, como son la capacidad de adaptarse a los datos de la serie de tiempo abstrayéndose de la funcionalidad de los modelos de predicción involucrados en la construcción del ensamble. El segundo método propuesto en esta investigación, denominado SOMSM, utiliza la capacidad de adaptación

de las redes SOM y el vector de error representativo para buscar un balance entre exactitud y diversidad al momento de seleccionar los modelos que conforman el ensamble.

Al analizar la complejidad computacional de los métodos propuestos, la extracción de meta-características demanda el mayor número de operaciones; y por lo tanto, limita superior mente la complejidad de los métodos propuestos en $O\big(2 \times M \times k \times (tr + fr)\big)$; re-emplazando el valor de tr que utiliza NARX se puede concluir que el orden de complejidad del método propuesto está acotado superiormente por $O(2 \times M \times k \times (m \times n_H)^2 \times N \times M_E)$.

Capítulo 5. Análisis Experimental

Este capítulo describe los resultados obtenidos por los métodos propuestos, al aplicarse a diferentes tipos de series de tiempo y comparándolo con otros modelos. El capítulo inicia con la descripción de los conjuntos de datos utilizados para los experimentos. Posteriormente se describen los modelos de predicción que se consideraron como entrada a los métodos para la construcción de los ensambles, incluyendo una descripción de los métodos contra los que se realizan comparaciones de la efectividad de los métodos propuestos. Para ambos métodos propuestos, se detalla cada experimento y se incluye un análisis de los resultados obtenidos.

Es importante hacer notar que todos los experimentos realizados son ciegos, esto es, se realizaron sin seleccionar ni eliminar ningún resultado. Asimismo, todas las secuencias de prueba son *"out-of-sample",* lo que significa que en ningún momento se utilizó dicha secuencia para realizar ningún tipo de ajuste a los modelos, el tamaño de las secuencias de prueba es igual al horizonte de predicción que depende del dominio.

5.1 Descripción de los Conjuntos de Datos

Treinta y cinco series conforman el conjunto de datos utilizado para evaluar los métodos propuestos; todas las series fueron normalizadas entre 0 y 1 para comparar fácilmente los resultados utilizando diferentes métricas. Las tablas 5.1-5.4 describen cada serie: la primera columna indica el identificador de la serie; la segunda columna incluye una descripción del origen de los datos, donde N es el número de observaciones de la serie de entrenamiento. La tercera columna muestra el número de elementos de la serie a predecirse, esto es, el horizonte de predicción (h); cada serie tiene una secuencia de prueba que corresponde a los siguientes h valores de la serie de tiempo, estos valores no participan en el entrenamiento. Las series se agruparon en tres tipos: la Tabla 5.1 muestra las series provenientes de competencias de predicción internacionales; las Tablas 5.2 y 5.3 describe las series sintéticas y la Tabla 5.4 contiene series obtenidas de diversas aplicaciones.

Tabla 5.1. Series de tiempo de competencias de predicción internacionales.

Id	Descripción	Horizonte
1-11	Conjunto reducido que es una muestra representativa del conjunto completo de la competencia de predicción NN3; estas series provienen de aplicaciones industriales; N varía entre 115 y 126.	18-19
12-22	Conjunto reducido que es una muestra representativa del conjunto completo de la competencia de predicción NN5, corresponden a retiros bancarios. Estas series tienen valores faltantes, que fueron interpolados de los valores vecinos; todas fueron manejadas con $N = 735$.	56

Tabla 5.2. Series de tiempo sintéticas

Id	Descripción	Horizonte
23	Integración de la ecuación diferencial de Mackey y Glass (1977): $\frac{dx}{dt} = \frac{0.2x_{t-17}}{1+x_{t-17}^{10}} - 0.1x_t$ Con x_o=1.2, un paso de integración de 0.1. Con estos parámetros la ecuación tiene un comportamiento caótico (Ouyang & Yin, 2014); $N = 750$.	250
24	Valor *x* de la integración de las ecuaciones de Lorentz (1963): $\frac{dx}{dt} = -10x_t + 10y_t$, $\frac{dy}{dt} = 28x_t - y_t - x_t z_t$, $\frac{dx}{dt} = x_t y_t - \frac{8}{3} z_t$ La integración se realizó con x_o = 12, y_o = 2, z_o = 9, paso de integración de 0.02. Se sabe que con estos parámetros la ecuación presenta un comportamiento caótico (Ouyang & Yin, 2014); $N = 750$. El código está disponible en: http://www.mathworks.com/examples/matlab/community/11765-mackeyglass.	250
25	Valor *x* del mapa canónico de Hénon, definido por (Hénon, 1976): $x_{t+1} = y_t + 1 - 0.4x_t^2$; $y_{t+1} = 0.3x_t$ Estos parámetros de la ecuación producen un comportamiento caótico (Miranian & Abdollahzade, 2013); $N = 750$.	250
26	Ejecución de un modelo ARMA(2,1) definido como: $y_{t+1} = 0.5y_t - 0.3y_{t-1} + \varepsilon_{t+1} + 0.2\varepsilon_{t-1}$ donde ε_t sigue una distribución Gaussiana con media 0. Esta serie presenta un comportamiento muy cercano al aleatorio; $N = 500$.	50

Tabla 5.3. Series de tiempo sintéticas (continuación). Ejemplos extremos con linealidad y no linealidad periódica

Id	Descripción	Horizonte
27	Función seno, generada con un paso de $2\pi/64$; $N = 750$.	250
28	Línea recta, con $N = 100$, generada con la ecuación: $y = 0.5x + 1$	20
29	Integración de dos funciones seno, basado en (Beale, et al., 2015). $y_1 = 0.25\sin(2\pi t/10) + 1$, $y_2 = 0.25\sin(2\pi t/5) - 1$ y_1 y y_2 son alternadas 10 veces, cada 100 valores; $N = 750$.	250

Tabla 5.4. Series de tiempo de diversas aplicaciones.

Id	Descripción	Horizonte
30	Número de pasajeros mensuales (en miles) de una aerolínea desde Enero de 1949 a Diciembre de 1960; con *N* = 132, disponible en (Hyndman, 2016). Según (Adhikari & Agrawal, 2014) esta serie mensual tiene un comportamiento estacional.	12
31	Muertes mensuales accidentales, ocurridas en Estados Unidos de América entre 1973-1978; con *N* = 60, disponible en (Hyndman, 2016). Según (Adhikari & Agrawal, 2012) esta serie mensual tiene un comportamiento estacional aditivo.	12
32	Número anual de linces atrapados en la rivera MacKenzie entre 1821-1934; con *N* = 100, disponible en (Hyndman, 2016). Según (Adhikari & Agrawal, 2014) esta serie es estacionaria y no-estacional.	14
33	Número de manchas solares registrado entre 1700-1988; con *N* = 221, disponible en (Hyndman, 2016). Según (Adhikari & Agrawal, 2014) esta serie es estacionaria y no estacional.	67
34	Producción nacional de Estados Unidos de América entre 1890 y 1974; con *N* = 70, disponible en (Hyndman, 2016). Según (Adhikari & Agrawal, 2014) esta serie es no-estacionaria y no-estacional.	17
35	Nacimientos divididos por 10,000 provenientes de madres de 23 años en Estados Unidos de América entre 1917 y 1975; con *N* = 49, disponible en (Hyndman, 2016). Según (Adhikari & Agrawal, 2014) esta serie es no-estacionaria y no-estacional.	10

Otros trabajos que siguen este esquema de evaluación son: (Crone, et al., 2011)(Makridakis & Hibon, 2000) y(Lemke & Gabrys, 2010), entre otros. Con la finalidad de tener una serie de validación para ajustar parámetros, las series de entrenamiento son divididas en una nueva serie de entrenamiento con $N - h$ observaciones y una secuencia de validación formada por los últimos *h* valores de la serie de entrenamiento.

5.1.1 ¿Cuáles Series son Caóticas?

Las series de tiempo pueden catalogarse como caóticas cuando su máximo exponente de Lyapunov (ME) es mayor a 0 (Kantz & Schreiber, 2003). La definición analítica de dicho número es imposible para series de tiempo obtenidas de sistemas dinámicos de los que se desconocen las ecuaciones que los definen, por lo que se han creado algunos métodos numéricos que permiten aproximar dicho valor. Dichos métodos son altamente dependientes de las condiciones en que se utilizan, por ejemplo el tamaño de la serie o el paso de integración. Aun así, esta aproximación numérica sirve para dar indicios sobre la sensibilidad a cambios en las condiciones iniciales, es decir comportamiento caótico, presente en una serie. A fin de catalogar los datos utilizados en los experimentos reportados aquí, se calculó el máximo exponente de Lyapunov de cada serie de tiempo utilizando la función *lyap_spec* proporcionada por el paquete TISEAN (Kantz & Schreiber, 2003), y descrita en (Sano & Sawada, 1985). La Tabla 5.5 muestra estos valores, marcando con negrita a las series con valores mayores de cero, presumiblemente caóticas. Las series marcadas con NA (no aplica) son conocidas por no ser caóticas. De acuerdo a los ME obtenidos, 14 de las 35 series utilizadas para nuestros experimentos presumiblemente presentan un comportamiento caótico.

Tabla 5.5. Estimación del máximo exponente de Lyapunov (ME) de cada serie. Si ME es mayor a 0, la serie puede considerarse caótica. El cálculo utilizó la función lyap_spec del paquete TISEAN.

NN3		NN5		Sintéticas		Aplicaciones	
Id.	**ME**	**Id.**	**ME**	**Id.**	**ME**	**Id.**	**ME**
1	-0.28	12	-0.002	**23**	**0.05**	30	-0.08
2	**0.15**	**13**	**0.12**	**24**	**0.02**	31	-0.38
3	**0.14**	14	-0.13	**25**	**0.42**	**32**	**0.26**
4	-0.03	**15**	**0.07**	26	-0.12	33	-0.02
5	-0.18	16	-0.03	**Extremos**		**34**	**0.02**
6	-0.46	**17**	**0.04**	**Id.**	**ME**	35	-0.15
7	-0.33	**18**	**0.07**	27	NA		
8	-0.48	**19**	**0.10**	28	NA		
9	-0.30	**20**	**0.01**	29	NA		
10	-0.09	21	-0.02				
11	-0.38	**22**	**0.03**				

5.2 Descripción del Conjunto de Modelos

Los métodos propuestos en esta investigación reciben un conjunto de modelos de predicción, como candidatos para la construcción del ensamble que devolverá la predicción final. La Tabla 5.6 lista los modelos candidatos utilizados en nuestros experimentos, los cuales fueron generados con tres bases distintas: ARIMA, NARX y *k*-trayectorias; la descripción detallada de los modelos está disponible en el Anexo F. Los parámetros de los modelos de predicción fueron escogidos con la intención de aumentar la diversidad de comportamiento entre ellos. El entrenamiento de la red neuronal NARX se realizó utilizando los siguientes algoritmos disponibles en *Matlab*®: "*trainbr*" (Retro-propagación (RP) con regulación Bayesiana), "traincgf" (RP de gradiente conjugado con actualizaciones Fletcher-Reeves), y "*trainlm*" (RP Leveberg-Marquardt).

Tabla 5.6. Modelos candidatos para la construcción del ensamble.

Modelos generados con base ARIMA	
Parámetros	**Valores posibles**
Componentes auto-regresivos p	0, 1, 2, 3
Componente integrada d	0, 1, 2, 3
Componentes de media móvil q	0, 1, 2, 3
Estacionalidad	0, 12
Nota: Las combinaciones $p = 0$, $d = 0$ y $q = 0$ no son consideradas.	
Número total de modelos con base ARIMA	126
Modelos generados con base NARX	
Parámetros	**Valores posibles**
Neuronas de retardo	25, 30, 35
Neuronas de la capa oculta	20, 25, 30
Algoritmo de entrenamiento	trainbr, traincgf, trainlm
Número total de modelos con base NARX	27
Modelos generados con base *k*-trayectorias	
Parámetros	**Valores posibles**
Número de retardos	25, 30, 35
Número *k* de vecinos	1, 10
Función de distancia	Euclidiana
Número total de modelos con base k-Trayectorias	6
Total de modelos	159

El tamaño del espacio de búsqueda para encontrar el modelo con la mejor estimación de desempeño es igual al tamaño del conjunto de modelos, es decir 159 modelos posibles.

5.3 Estrategias de comparación

Los métodos propuestos fueron comparados de varias maneras, siguiendo algunas de las prácticas actuales que aparecen en el estado del arte; éstas se describen a continuación.

5.3.1 Mejor Modelo

Un ensamble se considera exitoso si su desempeño supera al desempeño de cualquiera de los modelos que lo componen; esto puede evaluarse comparando el desempeño del ensamble con el desempeño del modelo utilizado que presentaría el menor error al aplicarse individualmente. Para hacer esto, en todos los experimentos ejecutados se calculó el error de desempeño obtenido por cada uno de los modelos disponibles para construir el ensamble, utilizando la serie de entrenamiento y las métricas SMAPE y MCCV. Posteriormente, se selecciona el modelo con el menor SMAPE; este modelo es denominado "mejor modelo." El procedimiento es repetido para seleccionar el mejor modelo por cada serie de tiempo.

5.3.2 Mejor Modelo por Base

En la práctica, es común que los expertos humanos recomienden un tipo de modelo base, apropiado para la serie de tiempo que se desea predecir, en vez de un modelo específico. A fin de representar esta práctica, se compararon los modelos propuestos contra el modelo con el menor error entre modelos generados a partir de un mismo modelo base. Por ejemplo, el mejor modelo NARX, es el modelo de base NARX con el menor error de todos los modelos generados con la base NARX. Estos modelos los identificamos en los resultados como "mejor modelo ARIMA", "mejor modelo NARX" y "mejor modelo *k*-trayectorias".

5.3.3 Método de Lemke

Dado que el trabajo más cercano a los propuestos fue el publicado por Lemke y Gabrys (2010) (ver sección 3.3.1), nuestros métodos se compararon contra éste, el cual fue codificado siguiendo la descripción publicada por los autores. Para esto, se utilizó el mismo conjunto de modelos disponibles para la construcción del ensamble propuestos en nuestros experimentos. Siguiendo las recomendaciones de los autores, el número de grupos que utiliza en su primer paso es cuatro.

5.4 Análisis de Eficacia de los Métodos Propuestos

La eficacia de los métodos propuestos es analizada sobre series de tiempo caóticas y otros subconjuntos de series; como se describió anteriormente, estos resultados se compararon con el desempeño del método de Lemke y de los mejores modelos de cada base. Cabe recordar que los resultados de eficacia se calcularon con valores de un conjunto de pruebas. Entonces, los parámetros de los métodos SOMSM y TVC (ver secciones 4.3 y 4.5) son ajustados utilizando una secuencia de validación; la Tabla 5.7 muestra los posibles valores de los parámetros de cada método. En los experimentos descritos en esta sección, la selección de los modelos es guiada por el desempeño medido con SMAPE. Adicionalmente, se evalúa el efecto en los resultados, usando dos diferentes funciones de combinación de predicciones: mediana (identificado como método-Mediana) y promedio (identificado como método-promedio).

Tabla 5.7. Posibles valores para los parámetros de los métodos TVC y SOMSM.

Método	Parámetro	Valores posibles
TVC	Número de modelos k	$\{1,...,M\}$
SOMSM	Tamaños de la red SOM ($N_C \times N_F$), utilizamos mapas cuadrados $N_C = N_F$.	{2x2, 3x3, ..., 10x10}
	Tamaño del vecindario l	$\{0,1,...,N_C\}$ (0 significa solo neurona A)
	Porcentaje de modelos a seleccionar p%	{10,20,...,100}
	Factor de tolerancia α	{0,1,1.5,2,2.5,...,5} (0 significa sin poda)

5.4.1 Evaluación en Series Consideradas Caóticas

La Tabla 5.8 detalla los resultados, evaluados con SMAPE, sobre las series cuya estimación del máximo exponente de Lyapunov es mayor a 0, donde se puede presumir que tienen un comportamiento caótico. La tabla compara los métodos propuestos con el mejor modelo, el mejor modelo de cada tipo y el método de Lemke.

Tabla 5.8. SMAPE de los métodos propuestos sobre las series consideradas caóticas. Estas son series cuyo máximo exponente de Lyapunov es mayor a cero, comparado con los otros métodos.

Tipo de Serie	SOMSM Promedio	SOMSM Mediana	TVC Promedio	TVC Mediana	Mejor	Mejor NARX	Mejor ARIMA	Mejor kTrayector	Lemke
NN3	38.47	39.84	32.58	36.00	42.91	38.91	42.91	**27.32**	30.61
NN3	29.65	24.17	34.38	**22.87**	38.80	82.65	24.91	38.80	35.33
NN5	24.54	23.06	**20.43**	**20.43**	20.75	25.65	46.84	20.75	26.28
NN5	**17.48**	18.69	17.95	19.04	20.99	22.97	19.30	20.99	20.74
NN5	14.75	14.75	16.52	15.58	14.75	15.31	29.60	14.75	**13.73**
NN5	**24.49**	25.14	26.13	24.95	30.36	27.92	26.40	30.36	29.33
NN5	23.67	20.09	26.00	22.11	**18.83**	24.86	34.90	18.83	19.12
NN5	15.70	**11.91**	14.96	12.91	13.73	17.05	16.89	13.73	14.42
NN5	17.52	17.18	17.85	**17.05**	21.07	17.49	22.16	21.07	18.82
Mackey Glass	0.0019	0.0007	0.0019	0.0007	**0.0006**	**0.0006**	39.87	7.61	7.07
Lorentz	38.25	38.24	32.74	**30.45**	58.31	58.31	40.55	30.60	55.51
Henon	74.69	48.91	48.32	61.70	46.40	60.65	46.40	53.04	**45.69**
Lynx	65.25	**65.12**	69.95	69.10	68.51	104.02	70.51	68.51	69.18
GNP	**3.87**	13.81	3.89	16.70	5.94	32.13	5.94	34.31	29.71
Promedio	27.74 ±20.25	**25.78 ±16.31**	25.84 ±17.17	26.35 ±17.88	28.67 ±19.12	37.71 ±27.75	33.37 ±15.58	28.62 ±15.72	29.68 ±16.55

La Figura 5.1 muestra el resumen de los resultados de los métodos propuestos y otros métodos evaluados con SMAPE sobre las series cuyo máximo exponente de Lyapunov es mayor a cero. La Tabla 5.9 resumen los resultados evaluados con diferentes métricas.

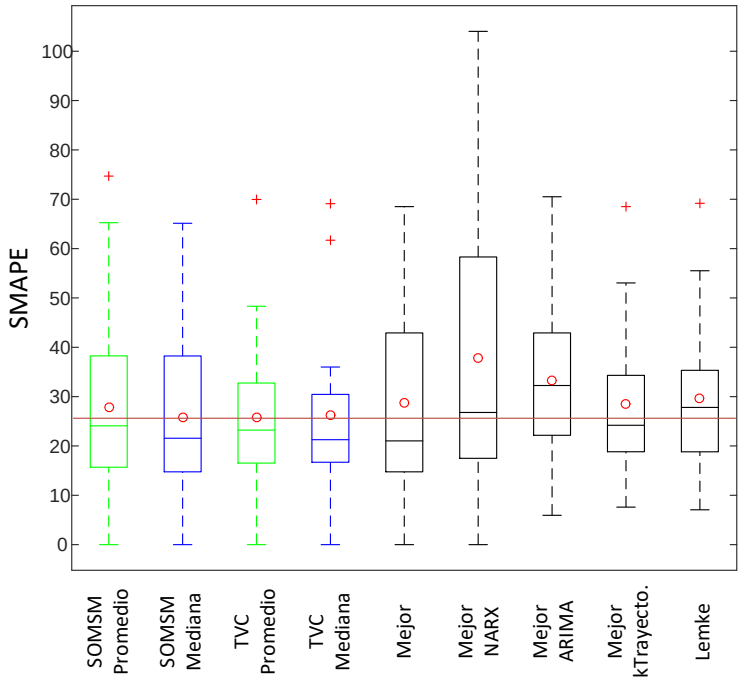


Figura 5.1. Resultados sobre las series cuyo máximo exponente de Lyapunov es mayor a cero. Caja con bigotes y promedio de SMAPEs obtenidos por los métodos propuestos y otros métodos.

Tabla 5.9. Resultados de los métodos propuestos con las series cuyo máximo exponente de Lyapunov es mayor a cero. Evaluación con diferentes métricas y comparación con otros modelos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
SOMSM Promedio	0.11 ±0.10	27.74 ±20.25	0.50 ±0.39	0.54 ±0.36	1.06 ±1.74
SOMSM Mediana	**0.11 ±0.07**	**25.78 ±16.31**	0.46 ±0.32	0.51 ±0.29	1.19 ±1.84
TVC Promedio	**0.11 ±0.07**	25.84 ±17.17	0.45 ±0.31	0.51 ±0.30	**0.96 ±1.45**
TVC Mediana	0.12 ±0.08	26.35 ±17.88	**0.42 ±0.31**	**0.50 ±0.29**	1.12 ±1.55
Mejor	0.13 ±0.11	28.67 ±19.12	0.50 ±0.32	0.60 ±0.41	1.59 ±3.43
Mejor NARX	0.17 ±0.13	37.71 ±27.75	0.56 ±0.33	0.68 ±0.41	2.16 ±3.87
Mejor ARIMA	0.15 ±0.08	33.37 ±15.58	0.71 ±0.41	0.75 ±0.38	2.02 ±3.05
Mejor kTrayect.	0.13 ±0.08	28.62 ±15.72	0.49 ±0.29	0.54 ±0.25	1.43 ±1.93
Lemke	0.13 ±0.07	29.68 ±16.55	0.50 ±0.29	0.55 ±0.25	1.59 ±2.36

El método SOMSM usando la función de combinación mediana obtiene en promedio mejores resultados, evaluados con SMAPE, sobre las series con máximo exponente de Lyapunov mayor a cero. La Tabla 5.10 muestra el resultado de la prueba de hipótesis nula Wilcoxon. Igual que en tablas anteriores un valor igual a cero significa que la diferencia no es significativa; cuando la diferencia si es significativa, el valor asignado corresponde al *p-value* con el que la prueba devuelve una diferencia significativa con una probabilidad de 95%. El poder estadístico está junto al *p-value*, es bajo porque la diferencia del efecto es pequeña y son pocas series de tiempo.

Tabla 5.10. Prueba significancia estadística Wilcoxon comparando SOMSM-mediana con los otros modelos, con las series de comportamiento caótico. Un valor cero indica que la diferencia no es significativa, en caso contrario se muestra el p-value respectivo y el poder estadístico.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
SOMSM Pro.	0	0	0	0	0
TVC Promedio	0	0	0	0	0
TVC Mediana	0	0	0	0	0
Mejor	0	0	0	0	0
Mejor NARX	2.44E-04 (82%)	1.22E-03 (79%)	8.54E-03 (91%)	2.44E-04 (91%)	2.44E-04 (48%)
Mejor ARIMA	2.95E-02 (44%)	1.66E-02 (72%)	6.71E-03 (74%)	1.07E-02 (65%)	2.95E-02 (27%)
Mejor kTray.	0	0	0	0	0
Lemke	0	4.19E-02 (62%)	0	0	4.19E-02 (48%)

5.4.2 Evaluación con Series de la Competencia NN5

Las series de la competencia de predicción "NN5" han sido ampliamente utilizadas para evaluar métodos de predicción de múltiples pasos. La Tabla 5.11 muestra el promedio de los resultados obtenidos por los métodos SOMSM y TVC utilizando las 11 series del conjunto reducido del torneo NN5, comparados con los resultados promedio de los otros métodos; los resultados de TVC y SOMSM están sombreados. La primera columna en la tabla muestra el nombre del método y la función de combinación, si aplica. Las siguientes columnas presentan el desempeño medido con: RMSE, SMAPE, GMRAE, RelMAE, y MASE; el mejor resultado por cada métrica está marcado en negrita.

Tabla 5.11. Resultados promedio de TVC, SOMSM y otros modelos sobre las series del torneo NN5, evaluación con métricas: RMSE, SMAPE, GMRAE, RelMAE, MASE.

NN5	RMSE	SMAPE	GMRAE	RelMAE	MASE
TVC-Mediana	0.11 ± 0.06	22.24 ± 7.77	0.56 ± 0.27	0.64 ± 0.27	0.63 ± 0.32
TVC-Promedio	0.12 ± 0.05	24.50 ± 7.82	0.61 ± 0.28	0.70 ± 0.25	0.68 ± 0.30
SOMSM-Promedio	0.10 ±0.048	23.00 ±6.50	0.60 ±0.31	0.65 ±0.26	0.63 ±0.29
SOMSM-Mediana	**0.10 ±0.047**	**21.00 ±7.00**	**0.54 ±0.28**	**0.61 ±0.25**	**0.59 ±0.28**
Mejor Modelo	0.11 ±0.06	23.61 ±8.42	0.64 ±0.32	0.67 ±0.27	0.65 ±0.32
Mejor NARX	0.11 ±0.05	23.46 ±5.94	0.59 ±0.23	0.67 ±0.22	0.65 ±0.26
Mejor ARIMA	0.13 ±0.05	29.66 ±9.08	0.76 ±0.27	0.83 ±0.26	0.81 ±0.30
Mejor kTrayec.	0.11 ±0.07	25.83 ±14.72	0.69 ±0.46	0.70 ±0.33	0.69 ±0.43

La mayor parte de las series del conjunto reducido del torneo NN5 presentan un comportamiento caótico (ver Tabla 5.5). Con la finalidad de comparar los métodos propuestos con otros trabajos reportados en la literatura, la Tabla 5.12 muestra los resultados del torneo NN5, evaluados con SMAPE, los cuales están disponibles en: http://www.neural-forecasting-competition.com/NN5/results.htm; nótese que estos resultados no reportan la desviación estándar. La tabla se encuentra ordenada de acuerdo al desempeño de cada método. Cada fila en la tabla muestra el alias de registro del participante y el SMAPE promedio que obtuvo, incluyendo los resultados obtenidos por SOMSM y TVC utilizando las funciones de combinación mediana y promedio. Cabe señalar que, aunque no se participó en dicha competencia, los métodos propuestos fueron evaluados exactamente bajo las mismas condiciones del torneo.

Tabla 5.12. Resultados del torneo NN5 con el conjunto reducido. Se incluyen los resultados de SOMSM y TVC utilizando las funciones de combinación promedio y mediana.

Nombre contendiente	SMAPE promedio	Nombre contendiente	SMAPE promedio
Wildi	**17.6**	SOMSM-Promedio	23.7
Rauch	19.0	Merkusheva	23.8
D'yakonov	19.9	Reilly	23.9
Andrawis	20.5	Lewicke	24.5
Lagoo	21.0	TVC-Promedio	24.5
Luna	21.1	Teddy	24.7
Hung	21.3	Beadle	24.9
Noncheva	21.7	Fillon	25.4
SOMSM-Mediana	21.7	Dang	25.4
Gutierrez	21.9	Coyle	25.9
TVC-Mediana	22.2	Naïve Seasonal	27.8
Wichard	22.4	Rabie	28.2
Vogel	22.4	Adeodato	29.9
Puma-Villanueva	23.1	Tung	34.6
Gao	23.2	Carbajal	40.1
Pasero	23.2	Naïve Level	48.6
Brentnall	23.4		

En los resultados se puede apreciar que de los métodos propuestos utilizando la función de combinación mediana obtienen mejor desempeño que utilizando la función de combinación promedio; esto se debe a la tolerancia a valores extremos de la función mediana.

5.4.3 Análisis de la Eficacia de TVC

En esta sección se muestran los resultados obtenidos con el método TVC (ver sección 4.3). Como se describió anteriormente, estos resultados se compararon con el desempeño del método de Lemke y de los mejores modelos de cada base. Cabe recordar que los resultados de eficacia se calcularon con valores de un conjunto de pruebas. Entonces, los parámetros de los métodos son ajustados utilizando una secuencia de validación. Dado que el método TVC selecciona los *k* mejores modelos de cada ventana de predicción, los

posibles valores de *k* se seleccionaron en un rango de uno hasta el número de modelos disponibles. En los experimentos descritos en esta sección, la selección de los modelos es guiada por el desempeño medido con SMAPE. Sin embargo, debido a que los resultados pueden cambiar de acuerdo a la métrica utilizada para seleccionar los modelos, se realizaron experimentos usando otras métricas, los cuales se reportan en el Anexo I. Adicionalmente, se evalúa el efecto en los resultados, usando dos diferentes funciones de combinación de predicciones: mediana (identificado como TVC-Mediana) y promedio (identificado como TVC-promedio).

La Figura 5.2 muestra una gráfica tipo “cajas con bigotes”, así como el valor promedio de los resultados (círculos) evaluados con SMAPE sobre todas las series, obtenidos por TVC comparándolos según las estrategias descritas en la sección 5.3. Cada caja muestra una línea horizontal que representa la mediana de los resultados; la parte inferior de la caja indica el 25-avo percentil, que es el valor bajo el cual está el 25% de los casos, la parte superior muestra el 75-avo percentil, correspondiente al 75% de los casos; las líneas punteadas están conectadas con los valores extremos, correspondientes aproximadamente al 99% de los casos, las cruces representan valores extremos y los círculos representan el valor promedio. La Tabla 5.13 muestra los resultados de TVC, evaluados con diferentes métricas, comparadas con los resultados de otros modelos.

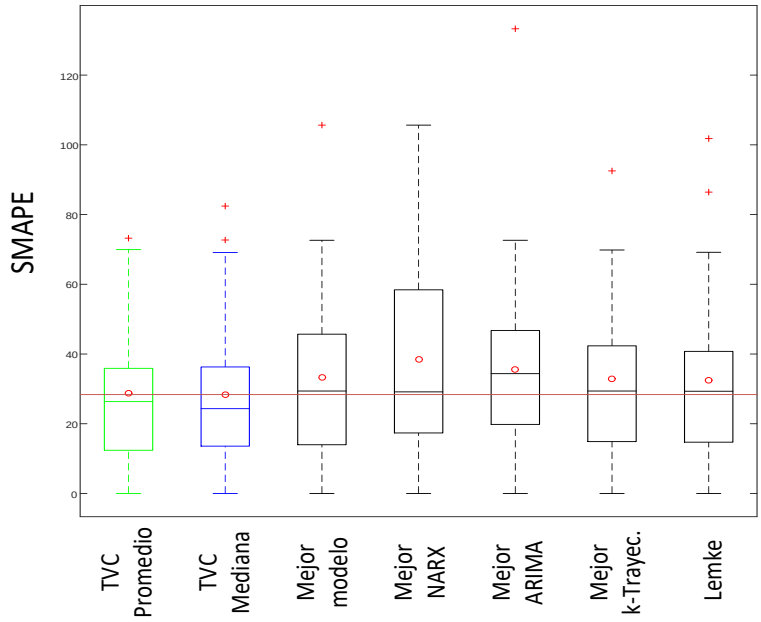


Figura 5.2. Caja con bigotes y promedio de SMAPEs del método TVC y otros métodos. Los resultados corresponden a las 35 series de tiempo, donde el método TVC obtiene mejore desempeño en promedio que los otros métodos.

Tabla 5.13. Resultados de TVC con todas las series de tiempo. Las predicciones se evalúan con diferentes métricas y comparan con las predicciones de otros modelos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
TVC - Promedio	**0.12 ±0.078**	29 ±20	0.55 ±0.42	0.61 ±0.42	**0.97 ±1.1**
TVC - Mediana	0.13 ±0.086	**28 ±21**	**0.51 ±0.37**	**0.59 ±0.36**	1.00 ±1.2
Mejor	0.15 ±0.140	33 ±25	0.66 ±0.61	0.72 ±0.53	1.50 ±2.7
Mejor NARX	0.17 ±0.150	38 ±30	0.68 ±0.55	0.77 ±0.50	1.80 ±3.0
Mejor ARIMA	0.15 ±0.100	36 ±25	0.65 ±0.45	0.72 ±0.41	1.50 ±2.2
Mejor kTrayec.	0.15 ±0.110	33 ±22	0.61 ±0.46	0.66 ±0.37	1.60 ±2.2
Lemke	0.14 ±0.095	33 ±23	0.60 ±0.45	0.63 ±0.36	1.40 ±1.9

En la Figura 5.2 y en la Tabla 5.13 puede notarse que TVC obtiene los mejores resultados en promedio. Estos resultados fueron analizados usando una prueba de hipótesis nula Wilcoxon (Gibbons & Chakraborti, 2003), los cuales se describen en la Tabla 5.14. Si el valor obtenido por la prueba es cero, significa que la diferencia no es significativa. Cuando la diferencia si es significativa, el valor asignado corresponde al *p-value* con el que la

prueba devuelve una diferencia significativa con una probabilidad de 95%. Los resultados de TVC-mediana son estadísticamente significativos comparando TVC con el mejor NARX y con el mejor ARIMA evaluando todas las métricas. La diferencia de TVC comparado con el mejor modelo y el mejor k-trayectorias es estadísticamente significativa sólo con algunas métricas. La diferencia de TVC combinado con mediana comparado con la combinación con promedio y el método de Lemke no es estadísticamente significativa.

Por otra parte, se calculó el poder estadístico[3] de la prueba Wilcoxon, el cual se muestra entre paréntesis junto al valor *p-value*. Puede notarse que en todos los casos su valor es bajo, principalmente porque la diferencia del efecto es pequeña y el número de series de tiempo utilizadas fue solo de 35. Los casos marcados con azul corresponden a los experimentos que obtuvieron un poder estadístico mayor o igual al 80% de confianza en la prueba de significancia estadística.

Tabla 5.14. Prueba significancia estadística Wilcoxon comparando TVC-mediana con los otros modelos. Un valor cero indica que la diferencia no es significativa; en caso contrario se muestra el p-value y el poder estadístico respectivos. Los valores marcados en azul representan a los mejores poderes estadísticos obtenidos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
TVC - Prom.	0	0	0	0	0
Mejor	0	2.09E-02 (64%)	2.95E-03 (62%)	1.83E-02 (63%)	2.28E-02 (34%)
Mejor NARX	4.29E-04 (79%)	1.76E-04 (87%)	6.57E-04 (71%)	3.56E-04 (85%)	5.48E-04 (63%)
Mejor ARIMA	1.40E-02 (25%)	8.36E-03 (43%)	9.33E-03 (65%)	7.23E-03 (51%)	1.22E-02 (22%)
Mejor kTray.	0	4.75E-02 (52%)	2.27E-02 (60%)	4.94E-02 (52%)	0
Lemke	0	0	0	0	0

5.4.4 Recomendaciones para el uso de TVC

A fin de determinar las condiciones en las que conviene utilizar el método TVC, se definió una regla para aplicarse sobre la serie, y determinar si debe o no utilizar para una serie de tiempo dada. De los experimentos ejecutados, pudo observarse que el método TVC

[3] El poder estadístico es calculado utilizando el software G*Power 3.1.9.2. Disponible en: http://www.gpower.hhu.de/

es apropiado para predecir series de tiempo que, de forma general, son difíciles de predecir por otros modelos aplicados individualmente. Para cuantificar el alcance de esta observación, se utilizó una de las características propuestas por Lemke y Gabris (2010), llamada "div1" (ver Anexo G). Esta característica se calcula usando el SMAPE promedio de los modelos sobre una secuencia de validación, al cual se le resta la desviación promedio con respecto a la media. Un valor de *div1* cercano a cero indica que los modelos disponibles predicen apropiadamente la serie de tiempo; un valor lejano de cero indica que los modelos disponibles no predicen adecuadamente la serie de tiempo. Adicionalmente, se observó que el sesgo (*skewness*) de los datos de una serie de tiempo normalizada entre 0 y 1, puede utilizarse para estimar la capacidad de predicción de TVC. Con respecto a este valor, si existe un comportamiento sesgado a la izquierda significa que la mayoría de valores están cercanos al mínimo valor posible, y por ende existen cambios bruscos que son difíciles de predecir por métodos trabajando individualmente.

Usando estas observaciones y el algoritmo de aprendizaje de reglas denominado "Poda Incremental Repetida para Producir Reducción de Error" (RIPPER, por sus siglas en inglés) (Cohen & Singer, 1999), sobre el desempeño de TVC con las 35 series de tiempo, se determinó la siguiente regla para utilizar TVC.

$$div1 \geq 35.12 \text{ y } skew \geq 0.47 => usar\ TVC\ mediana \tag{5.1}$$

A fin de ejemplificar la racionalidad de esta regla, la Figura 5.3 muestra un análisis de la dificultad de predicción de la serie id = 29. En la parte (a) de la figura, se muestran los datos de entrenamiento y prueba de la serie, marcando la mediana y el promedio de los datos. Nótese que esta serie es periódica, donde el número de valores entre cada período es de 100. Este valor grande en el tamaño del periodo dificulta la predicción de esta serie con los modelos utilizados individualmente. La sección (b) de la figura muestra el histograma de los datos de la serie donde se aprecia un claro sesgo en los datos. La sub-figuras siguientes muestran los histogramas de desempeño de los modelos de predicción, evaluado con SMAPE, utilizando el (c) horizonte completo y ventanas de (d) corto, (e) mediano y (f) largo plazo; en estas sub-figuras el recuadro de líneas punteadas enfatiza como el desempeño de los modelos disminuye, es decir tiende a 200, cuando crece el horizonte de predicción.

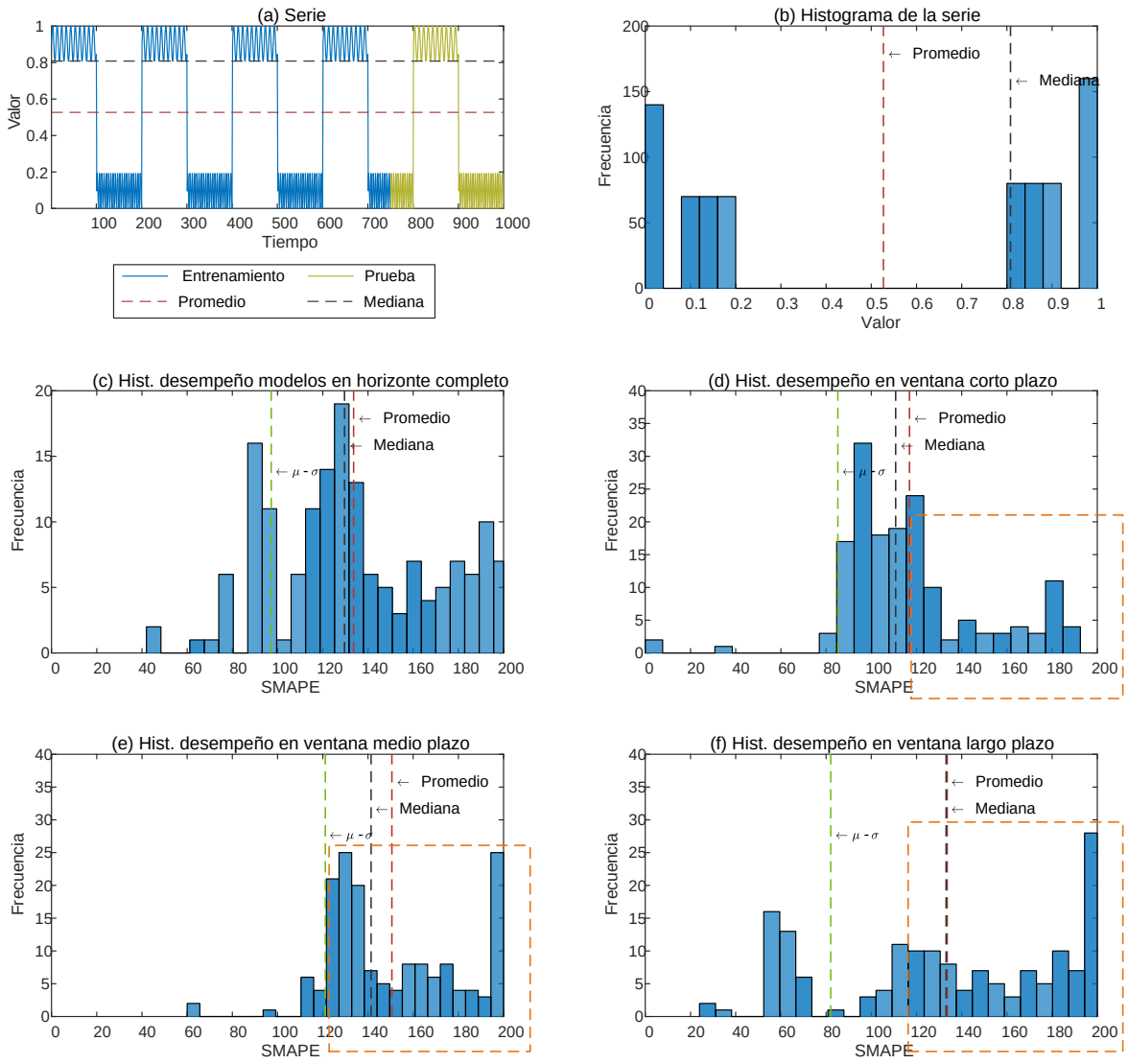


Figura 5.3. Análisis de la serie con id =29. La cual es sugerida para el uso de TVC utilizando la regla definida. (a) Datos de entrenamiento y prueba. (b) Histograma de los valores de la serie. (c) Desempeño de los modelos en el horizonte completo. Desempeño en las ventanas de (d) corto, (e) mediano y (f) largo plazo.

La Figura 5.4 muestra las predicciones del método TVC, combinando con mediana y con promedio, comparadas con los valores reales (serie de prueba) y con las predicciones de los otros métodos. Aquí se puede apreciar que la predicción más cercana a la esperada fue obtenida por TVC, en ambos casos de combinación: mediana y promedio.

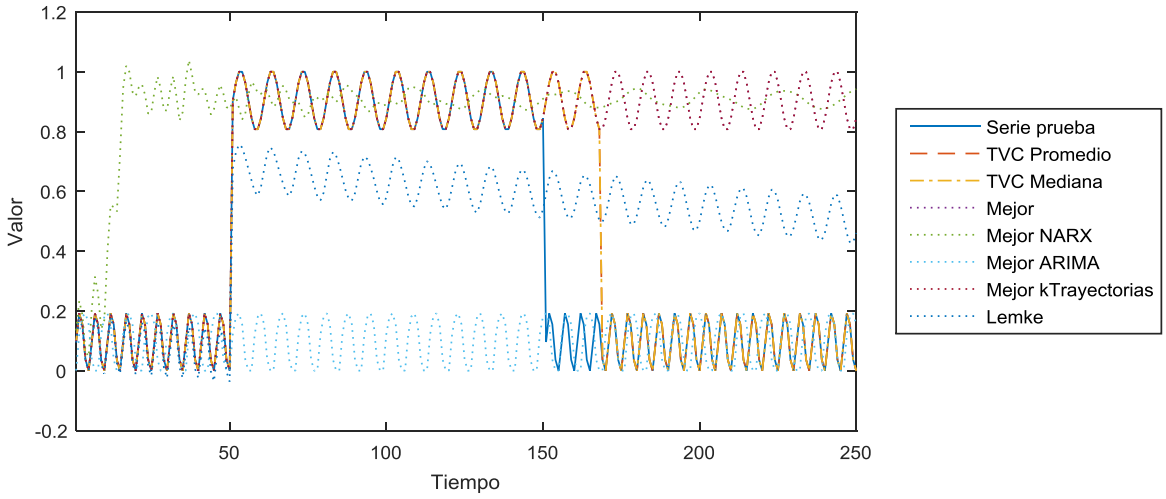


Figura 5.4. Predicción del método TVC sobre la serie Id = 29, comparado con la serie de prueba y los otros métodos.

La Figura 5.5 muestra una gráfica “caja con bigotes” de los resultados obtenidos para los casos donde la regla recomendó el uso de TVC, al aplicarse con todas las series. En estos experimentos se midió el desempeño usando SMAPE; la Tabla 5.15 muestra estos mismos resultados evaluados con otras métricas.

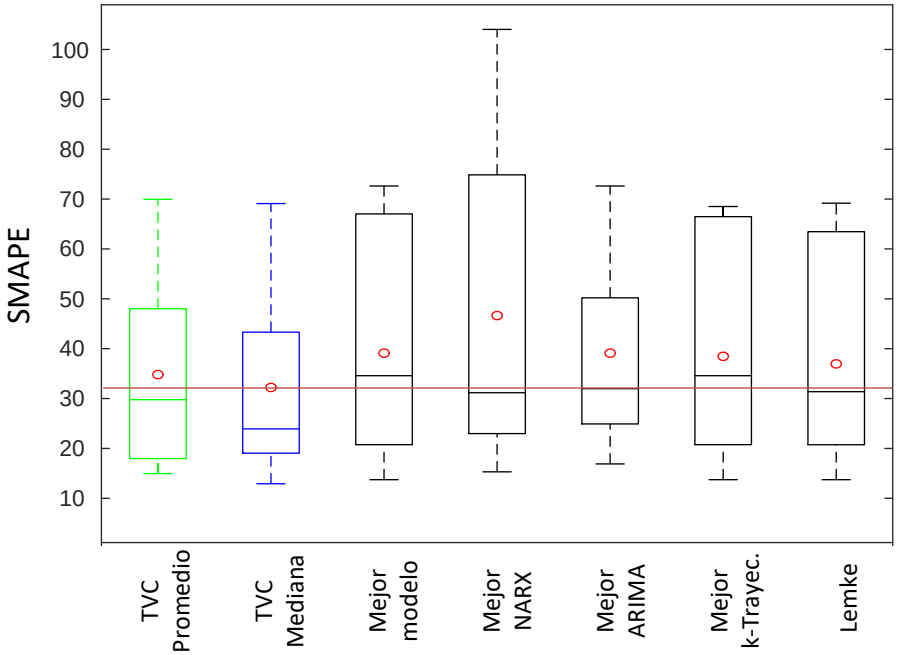


Figura 5.5. Resultados obtenidos con las series en las que se recomienda utilizar TVC. Caja con bigotes y promedio de SMAPEs del método TVC comparado con los otros métodos. TVC tienen un mejor desempeño en promedio.

Tabla 5.15. Resultados de TVC cuando la regla propuso su uso. Se muestran evaluaciones usando diferentes métricas y comparaciones con otros modelos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
TVC - Promedio	0.11 ±0.068	35 ±19	0.53 ±0.28	0.60 ±0.27	0.70 ±0.48
TVC - Mediana	**0.11 ±0.066**	**32 ±19**	**0.46 ±0.25**	**0.54 ±0.24**	**0.63 ±0.43**
Mejor	0.12 ±0.069	39 ±22	0.62 ±0.35	0.66 ±0.30	0.77 ±0.49
Mejor NARX	0.14 ±0.073	47 ±30	0.59 ±0.30	0.67 ±0.24	0.78 ±0.43
Mejor ARIMA	0.13 ±0.064	39 ±19	0.67 ±0.37	0.70 ±0.29	0.80 ±0.47
Mejor kTrayec.	0.12 ±0.070	38 ±21	0.60 ±0.33	0.66 ±0.29	0.76 ±0.46
Lemke	0.12 ±0.064	37 ±20	0.55 ±0.27	0.62 ±0.24	0.72 ±0.42

La Tabla 5.16 muestra el resultado de la prueba de hipótesis nula Wilcoxon (Gibbons & Chakraborti, 2003). En esta tabla, el valor cero significa que la diferencia no es significativa; cuando la diferencia es significativa, el valor asignado corresponde al *p-value* con el que la prueba devuelve una diferencia significativa con una probabilidad de 95%. El poder estadístico se muestra entre paréntesis junto al *p-value*; este valor es bajo principalmente porque el número de series es pequeño y la diferencia del efecto es pequeña.

Tabla 5.16. Prueba significancia estadística Wilcoxon comparando TVC-mediana con los otros modelos, siguiendo la recomendación de uso de TVC. El valor de cero indica que la diferencia no es significativa, caso contrario se muestran el p-value y el poder estadístico respectivos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
TVC - Prom.	3.91E-02 (47%)	1.95E-02 (57%)	7.81E-03 (93%)	7.81E-03 (73%)	7.81E-03 (65%)
Mejor	4.88E-02 (50%)	2.73E-02 (67%)	9.77E-03 (71%)	5.86E-03 (82%)	5.86E-03 (85%)
Mejor NARX	1.95E-03 (45%)	3.91E-03 (59%)	1.95E-03 (96%)	9.77E-03 (88%)	9.77E-03 (75%)
Mejor ARIMA	1.95E-03 (54%)	1.95E-03 (67%)	3.91E-03 (80%)	1.95E-03 (62%)	1.95E-03 (73%)
Mejor kTray.	4.88E-02 (52%)	2.73E-02 (60%)	9.77E-03 (62%)	5.86E-03 (76%)	5.86E-03 (78%)
Lemke	0	2.73E-02 (53%)	1.95E-02 (73%)	1.37E-02 (76%)	1.37E-02 (70%)

5.4.5 Análisis de la Eficacia de SOMSM

En esta sección se muestran los resultados obtenidos por el segundo método propuesto, SOMSM, el cual se basa en mapas auto-organizados y meta-características. La Figura 5.6 muestra las cajas con bigotes y el valor promedio de los resultados evaluados con SMAPE

sobre todas las series comparado con los otros métodos. Al igual que con TVC, se presentan resultados utilizando las funciones de combinación promedio y mediana. La Tabla 5.17 muestra los resultados de SOMSM evaluados con diferentes métricas y usando todas las series de tiempo, también comparados con los resultados de otros métodos. Los resultados de esta tabla muestran que el método SOMSM, combinando con la función mediana, obtiene resultados en promedio mejores que los otros métodos. La Tabla 5.18 muestra el resultado de la prueba de hipótesis nula Wilcoxon (Gibbons & Chakraborti, 2003), donde el valor "cero" significa que la diferencia no es significativa; cuando la diferencia si es significativa, el valor asignado corresponde al *p-value* con el que la prueba devuelve una diferencia significativa con una probabilidad de 95%. Al igual que en casos anteriores, el poder estadístico de esta prueba se muestra junto al *p-value*; nótese que éste es bajo porque la diferencia del efecto es pequeña y el número de series de tiempo es pequeño.

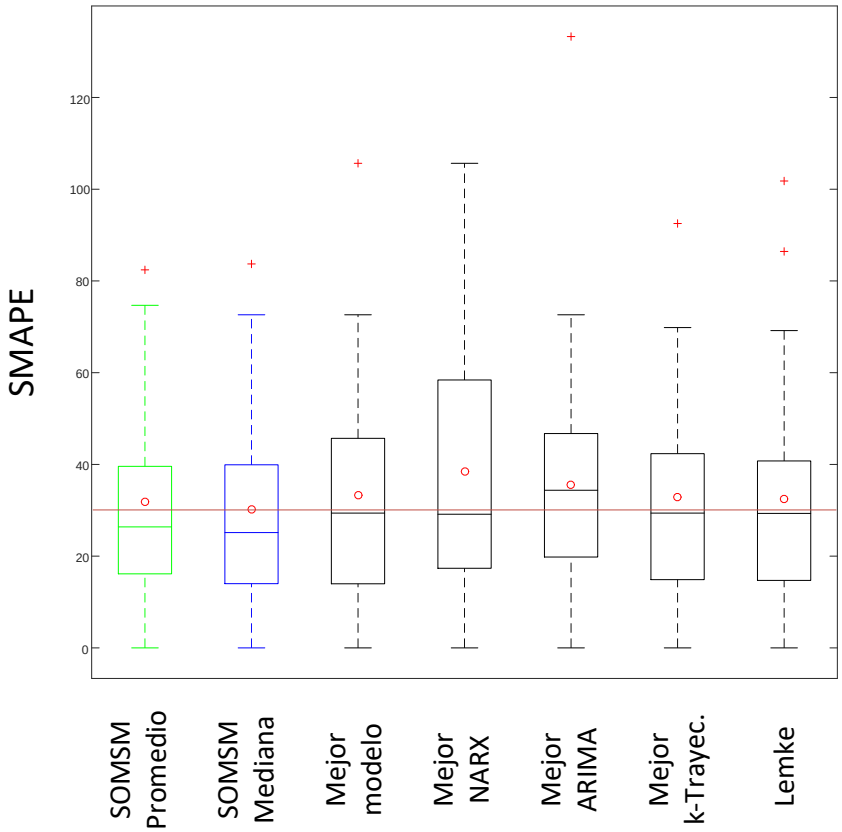


Figura 5.6. Caja con bigotes y promedio de SMAPEs del método SOMSM y los otros métodos. Los resultados fueron obtenidos con las 35 series de tiempo. SOMSM tienen un mejor desempeño en promedio.

Tabla 5.17. Resultados de SOMSM con todas las series de tiempo. Evaluación con diferentes métricas y comparación con otros modelos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
SOMSM - Prom.	0.14 ±0.110	32 ±22	0.59 ±0.43	0.64 ±0.37	**1.1 ±1.4**
SOMSM - Mediana	**0.13 ±0.100**	**30 ±22**	**0.55 ±0.42**	**0.61 ±0.36**	1.2 ±1.5
Mejor	0.15 ±0.140	33 ±25	0.66 ±0.61	0.72 ±0.53	1.5 ±2.7
Mejor NARX	0.17 ±0.150	38 ±30	0.68 ±0.55	0.77 ±0.50	1.8 ±3.0
Mejor ARIMA	0.15 ±0.100	36 ±25	0.65 ±0.45	0.72 ±0.41	1.5 ±2.2
Mejor kTrayec.	0.15 ±0.110	33 ±22	0.61 ±0.46	0.66 ±0.37	1.6 ±2.2
Lemke	0.14 ±0.095	33 ±23	0.60 ±0.45	0.63 ±0.36	1.4 ±1.9

Tabla 5.18. Prueba significancia estadística Wilcoxon comparando SOMSM-mediana con los otros modelos, utilizando todas las series. Un valor cero indica que la diferencia no es significativa, en caso contrario se muestra el p-value respectivo y el poder estadístico.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
SOMSM - Pro.	0	4.49E-02 (42%)	0	0	0
Mejor	1.66E-02 (40%)	3.85E-03 (83%)	3.63E-03 (51%)	1.59E-03 (60%)	3.16E-03 (29%)
Mejor NARX	7.56E-05 (82%)	6.55E-05 (92%)	1.29E-02 (52%)	1.42E-04 (81%)	8.72E-05 (64%)
Mejor ARIMA	1.47E-02 (23%)	7.56E-03 (53%)	2.87E-02 (34%)	7.97E-03 (46%)	1.09E-02 (15%)
Mejor kTraye.	2.27E-02 (58%)	0	0	4.44E-02 (29%)	3.58E-02 (27%)
Lemke	0	0	0	0	0

5.4.6 Desempeño de SOMSM con la base de datos NN5

La Figura 5.7 muestra la caja con bigotes de los resultados de SOMSM utilizando la función de combinación mediana y promedio sobre el conjunto NN5; los resultados son comparados con los obtenidos por otros métodos. La Tabla 5.19 muestra los resultados de SOMSM sobre las series del conjunto NN5, evaluados con diferentes métricas; también se compara con los resultados de otros métodos.

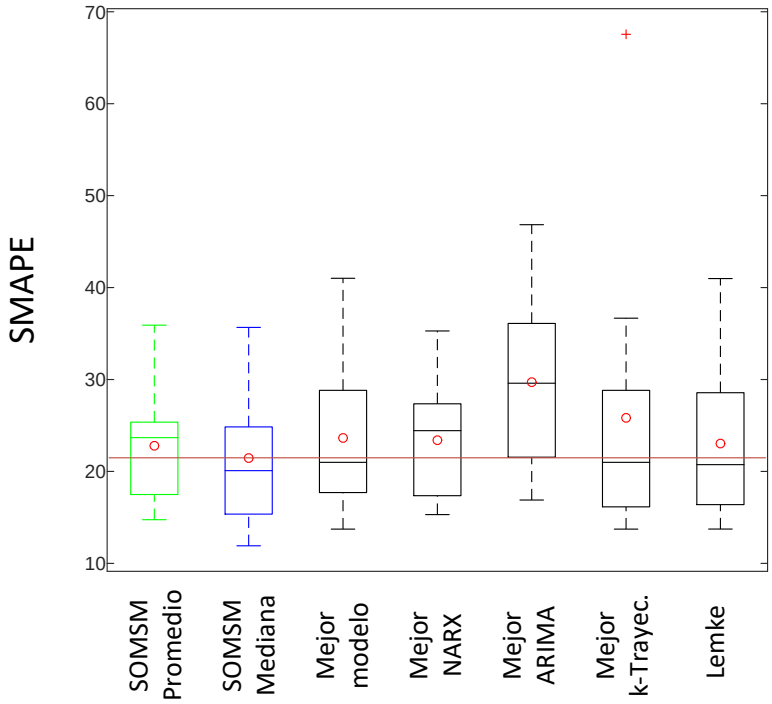


Figura 5.7. Resultados del método SOMSM y los otros métodos con las series del conjunto NN5. Caja con bigotes y promedio de SMAPEs obtenidos. SOMSM tienen un mejor desempeño en promedio.

Tabla 5.19. Resultados de SOMSM sobre las series del conjunto NN5. Evaluación con diferentes métricas y comparación con otros modelos.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
SOMSM - Prom.	0.10 ±0.048	23 ±6.5	0.60 ±0.31	0.65 ±0.26	0.63 ±0.29
SOMSM - Mediana	**0.10 ±0.047**	**21 ±7.0**	**0.54 ±0.28**	**0.61 ±0.25**	**0.59 ±0.28**
Mejor	0.11 ±0.056	24 ±8.4	0.64 ±0.32	0.67 ±0.27	0.65 ±0.32
Mejor NARX	0.11 ±0.048	23 ±5.9	0.59 ±0.23	0.67 ±0.22	0.65 ±0.26
Mejor ARIMA	0.13 ±0.050	30 ±9.1	0.76 ±0.27	0.83 ±0.26	0.81 ±0.30
Mejor kTrayec.	0.11 ±0.069	26 ±15	0.69 ±0.46	0.70 ±0.33	0.69 ±0.43
Lemke	0.11 ±0.052	23 ±7.8	0.57 ±0.25	0.65 ±0.22	0.63 ±0.29

SOMSM usando la función de combinación mediana obtiene resultados en promedio mejores sobre las series del conjunto NN5, La Tabla 5.20 muestra el resultado de la prueba de hipótesis nula Wilcoxon. Igual que en tablas anteriores un valor igual a cero significa que la diferencia no es significativa; cuando la diferencia si es significativa, el valor asignado corresponde al *p-value* con el que la prueba devuelve una diferencia significativa

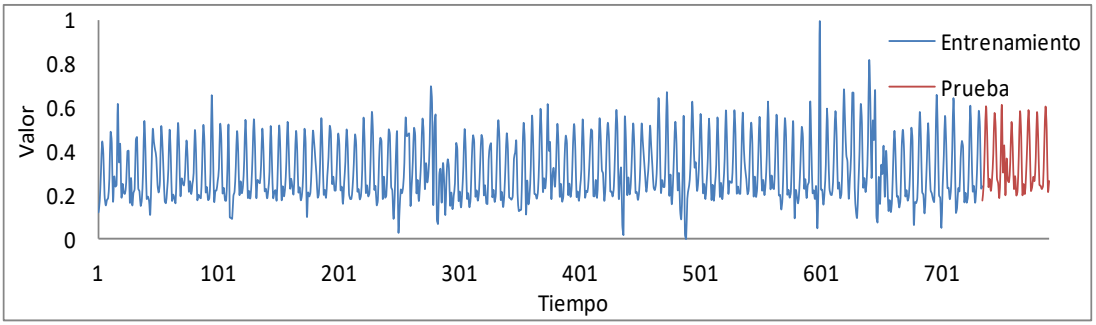
con una probabilidad de 95%. El poder estadístico está junto al *p-value*, es bajo porque la diferencia del efecto es pequeña y son pocas series de tiempo.

Tabla 5.20. Prueba significancia estadística Wilcoxon comparando SOMSM-mediana con los otros modelos, con las series del conjunto NN5. Un valor cero indica que la diferencia no es significativa, en caso contrario se muestra el p-value respectivo y el poder estadístico.

	RMSE	SMAPE	GMRAE	RELMAE	MASE
SOMSM - Prom.	0	4.88E-02 (57%)	0	0	4.88E-02 (50%)
Mejor	0	0	5.86E-03 (77%)	1.95E-02 (73%)	1.95E-02 (63%)
Mejor NARX	1.37E-02 (82%)	0	0	3.22E-02 (61%)	3.22E-02 (59%)
Mejor ARIMA	9.77E-04 (86%)	4.88E-03 (88%)	9.77E-03 (83%)	4.88E-03 (89%)	4.88E-03 (90%)
Mejor kTray.	0	0	0	1.95E-02 (34%)	2.73E-02 (27%)
Lemke	0	0	0	0	0

SOMSM mostró su mejor desempeño con el conjunto de series NN5, donde la mayoría de las series tienen un comportamiento caótico. La Figura 5.8 (a) muestra la serie Id. 20 que corresponde con la 9 serie del conjunto NN5, y la Figura 5.8 (b) muestra solamente la porción de la secuencia de prueba de la serie Id 20. Dada su naturaleza, las series NN5 pueden contener patrones de series de tiempo como: múltiple traslape de estacionalidades, tendencias locales, rupturas estructurales y *outliers*. Estos efectos son consecuencia de una combinación de fuerzas causales no-observadas y desconocidas, conducidas por el año calendario subyacente, como períodos estacionales re-ocurrentes, feriados bancarios, o eventos especiales de diferente longitud, y magnitud de impacto, con diferentes efectos de dirección y retardo (Crone, 2010).

(a)



(b)

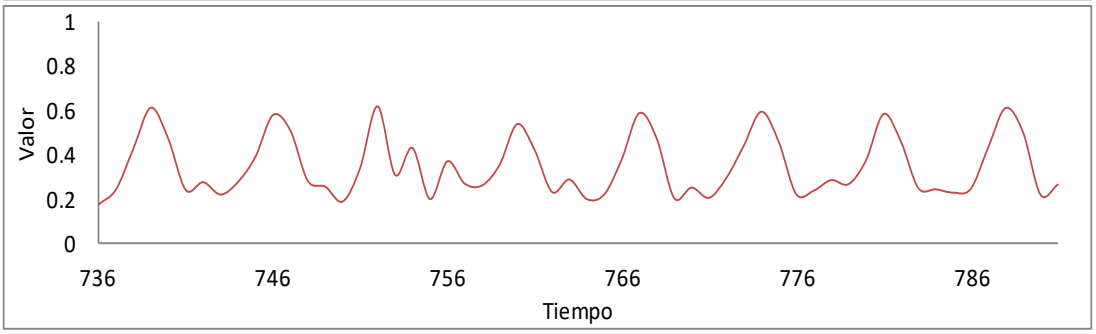


Figura 5.8. Serie Id 20, correspondiente a la 9 serie del conjunto NN5. (a) secuencias de entrenamiento y de prueba. (b) serie de prueba aumentada. Se aprecian cambios bruscos.

La Figura 5.9 (a) muestra el histograma de los datos de la serie Id 20, donde se aprecia que la serie tiene un sesgo a la izquierda, es decir la mayoría de sus valores se encuentran cercanos al mínimo. Sin embargo, un importante conjunto de valores se encuentran a la derecha. Esto implica que al menos existe un salto brusco en los datos; al observar la Figura 5.8 se aprecia que se presentan varios saltos bruscos. Una herramienta útil para analizar si existe una relación entre los valores del pasado y los valores del futuro es el mapa de retorno, donde un eje contiene valores del pasado con un retraso definido y el otro eje tiene los valores sin ningún retraso. La Figura 5.9 (b) muestra el mapa de retorno de la serie Id 20; aquí se aprecia que la figura resultante tiene una forma definida aunque irregular, esto quiere decir que existe un sistema complejo que genera los datos pero que es difícil de predecir.

(a) Histograma serie

(b) Mapa de retorno

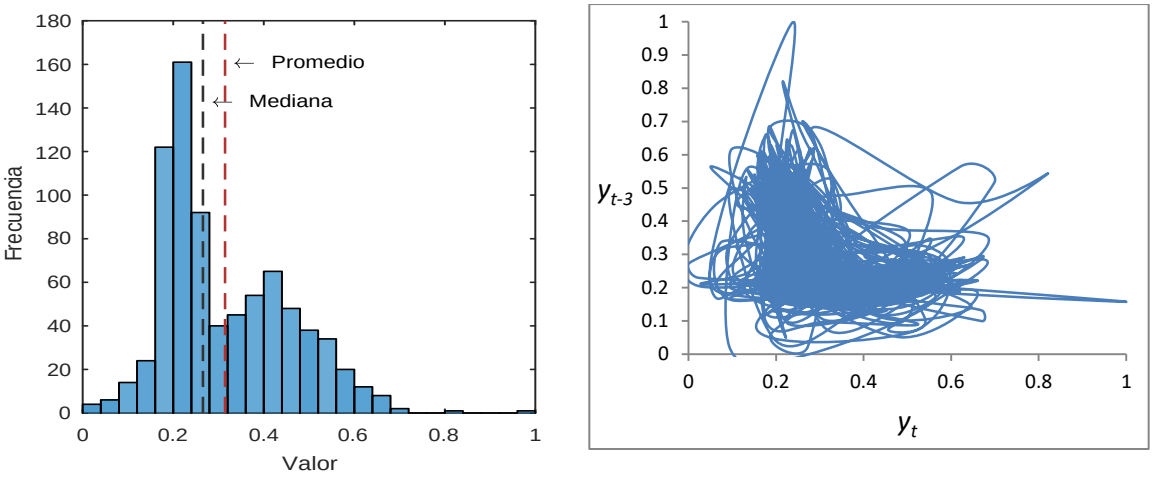


Figura 5.9. Histograma y mapa de retorno de la serie Id=20, 9na del conjunto NN5. Se aprecia sesgo en los datos y un patrón de relación entre el pasado y el futuro con un retraso t-3.

La Figura 5.10 muestra el mapa auto-organizado creado por SOMSM para la serie con Id. 20, utilizando la secuencia de entrenamiento. Cada hexágono en la figura representa un grupo de modelos, y dentro de éstos, en la primera línea se muestra el identificador de la neurona y el número de modelos agrupados en esta neurona, el cual está encerrado entre corchetes y también es representado por el tamaño del hexágono interno. La parte derecha la figura incluye una barra de código de color asociada al valor de SMAPE posible. La segunda línea muestra el SMAPE promedio y la desviación estándar respectiva de los modelos contenidos en ese grupo; este valor también está representado con el color del hexágono interno. La tercera línea muestra los valores de SMAPE mínimo y máximo de los modelos contenidos en ese grupo, si la neurona solo tiene un modelo no se muestra esta línea. La neurona que contiene al modelo con el menor SMAPE de todos está marcada con la letra *A*, el vecindario a un enlace de distancia alrededor de la neurona *A* está marcado con una línea en negrita. En la Figura 5.10 puede apreciarse que el desempeño de los modelos en las neuronas lejanas con respecto a la neurona *A* disminuye, es decir tienen un mayor SMAPE. La Figura 5.11 muestra la predicción devuelta por SOMSM con la serie Id 20 comparándola con los valores correctos, esto es, el segmento de prueba de la serie.

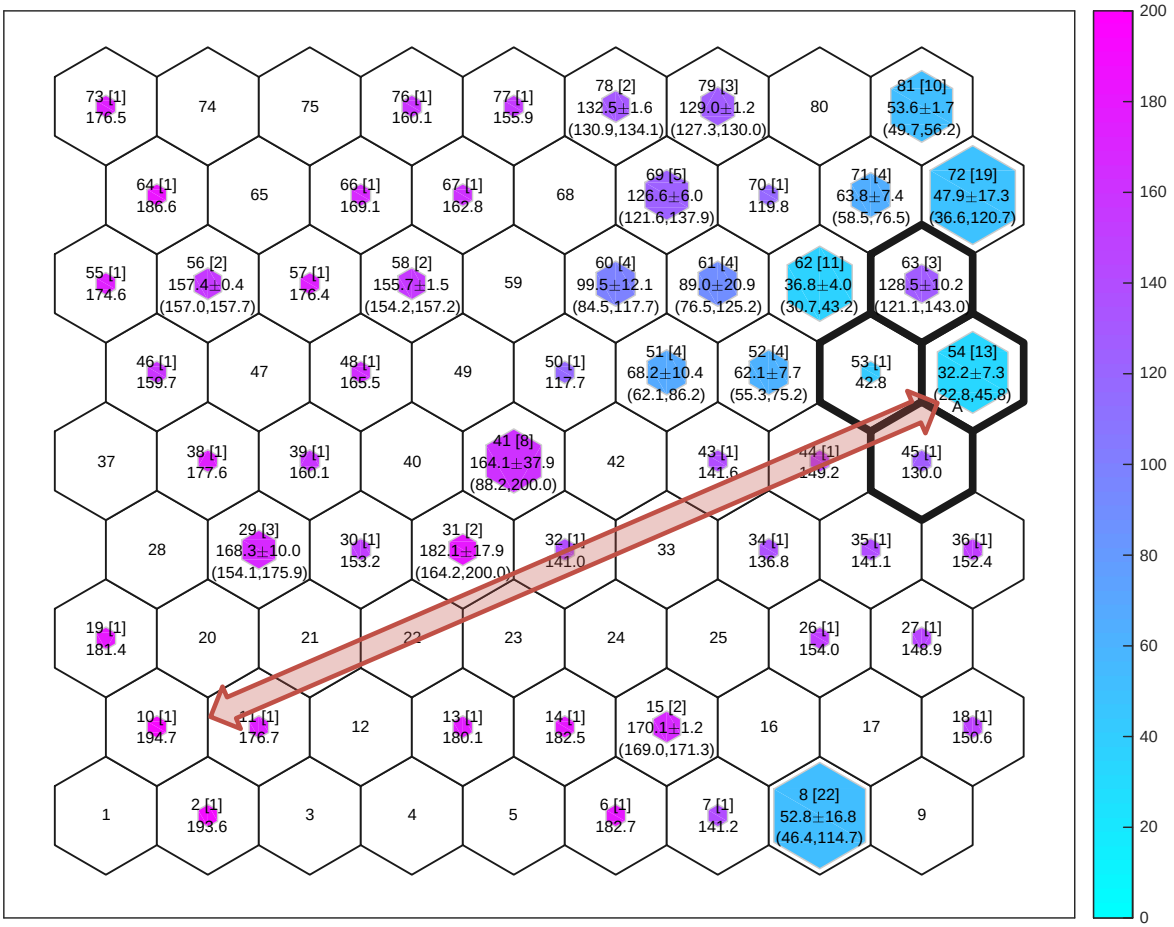


Figura 5.10. Mapa auto-organizado de los modelos de predicción en función de su error representativo con la serie Id 20, SMAPE validado con MCCV.

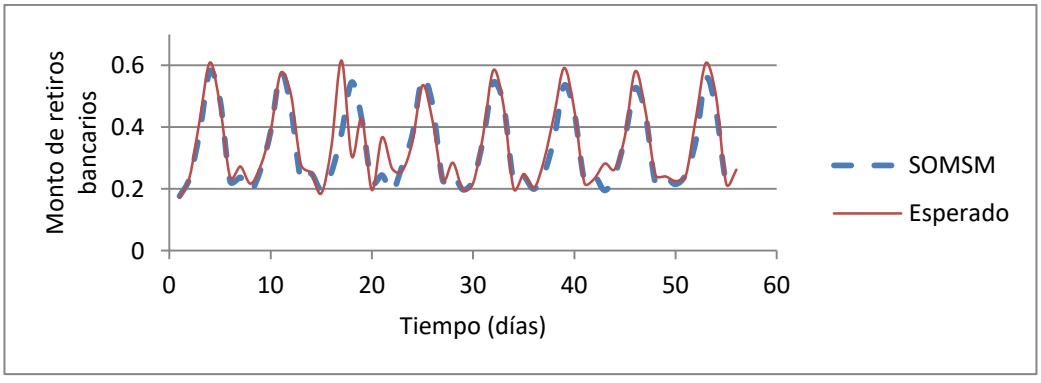
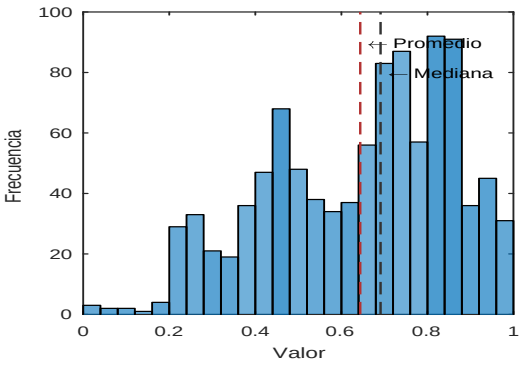


Figura 5.11. Predicción de SOMSM y valor esperado con la serie Id. 20, 9na del conjunto NN5.

5.4.7 Casos extremos de Predicciones con SOMSM

Las siguientes figuras muestran una comparación entre las mejores y peores predicciones obtenidas por SOMSM. Las figuras de las predicciones están acompañadas de figuras que muestran el histograma de los datos y el mapa de retorno; el histograma de los datos nos da una idea de la distribución de los datos. Un ejemplo de excelentes resultados se obtuvo con la serie caótica Mackey-Glass. La Figura 5.12 muestra el histograma y mapa de retorno de esta serie y los resultados de la predicción se muestran en la Figura 5.13. El caso contrario, es decir, un ejemplo imposible de predecir fue el de la serie generada con el modelo ARMA definido, lo cual se debe a la aleatoriedad de los datos presentes en esa serie. El histograma y mapa de retorno se muestran en la Figura 5.14; la Figura 5.15 muestra el resultado de la predicción.

(a) Histograma serie



(b) Mapa de retorno

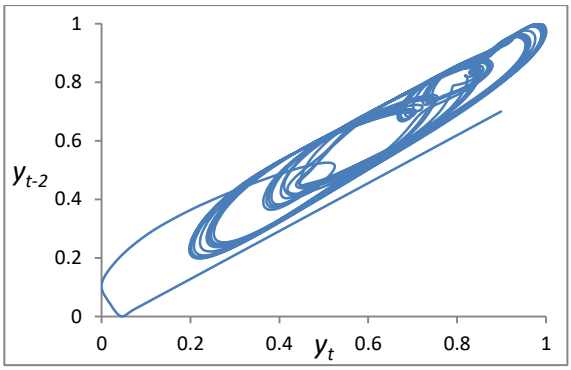


Figura 5.12. Histograma y mapa de retorno de la serie caótica Mackey-Glass. Se aprecia no normalidad de datos y patrones de relación entre el pasado y el futuro con un retraso t-2.

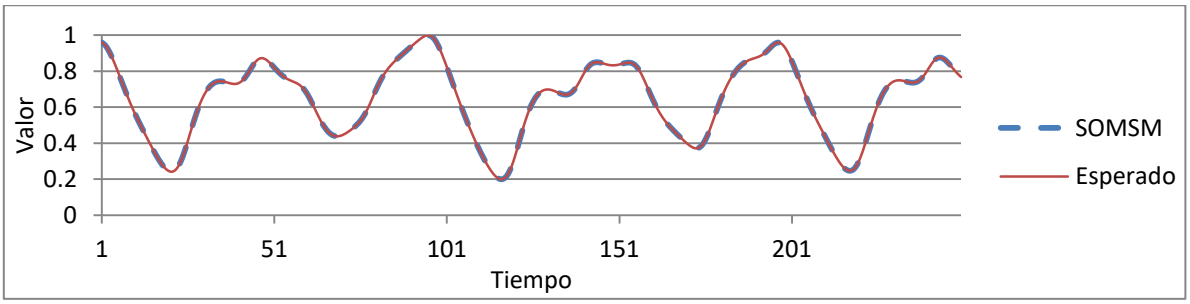
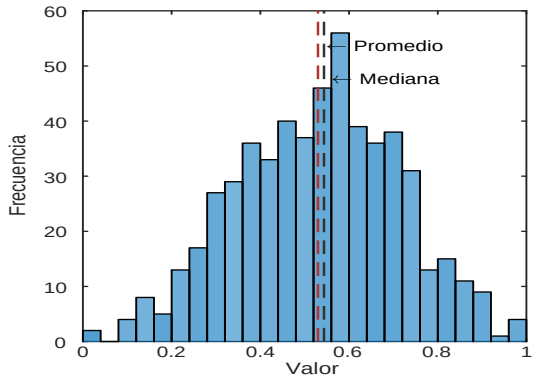


Figura 5.13. Predicción de SOMSM y valor esperado con la serie caótica Mackey-Glass.

(a) Histograma serie



(b) Mapa de retorno

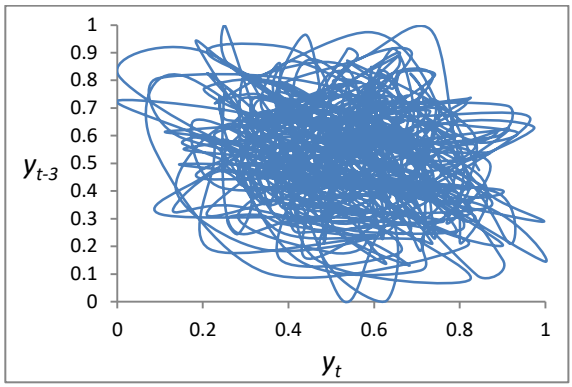


Figura 5.14. Histograma y mapa de retorno de la serie sintética generada con un modelo ARMA definido. Se aprecia normalidad en los datos, pero predomina un comportamiento aleatorio en el mapa de retorno.

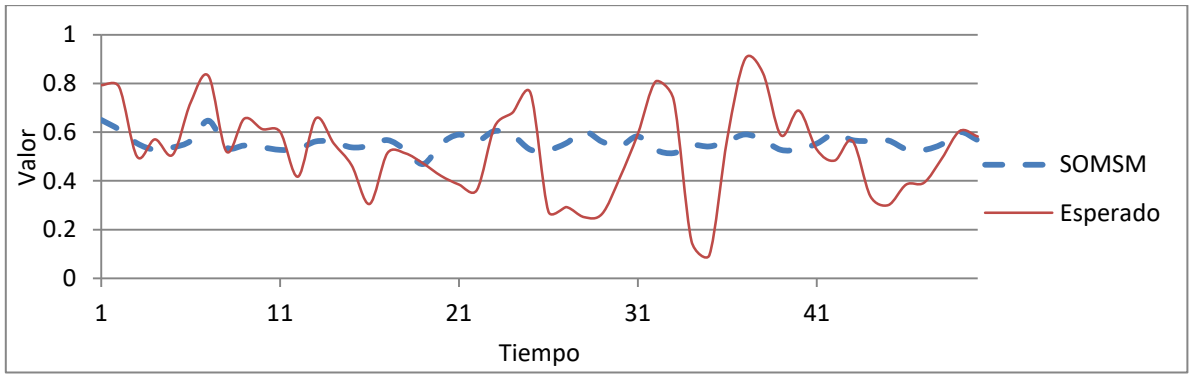


Figura 5.15. Predicción de SOMSM y valor esperado con la serie sintética generada con un modelo ARMA definido.

5.5 Evaluación con Diferentes Métricas

Las siguientes tablas resumen los resultados de los métodos TVC y SOMSM utilizando las funciones de combinación promedio y mediana, comparados con otros métodos; las series de tiempo están agrupadas por su origen. Los métodos propuestos son ajustados con la función SMAPE. Cada tabla muestra los resultados evaluados con diferentes métricas: la Tabla 5.21 RMSE, la Tabla 5.22 SMAPE, la Tabla 5.23 GMRAE, la Tabla 5.24 RelMAE y la Tabla 5.25 MASE.

Tabla 5.21. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con RMSE, comparado con otros métodos.

	Conjunto de series de tiempo			
Método	**NN3**	**NN5**	**Sintéticas**	**Aplicaciones**
TVC Promedio	0.11 ±0.042	0.12 ±0.064	**0.13 ±0.11**	0.15 ±0.089
TVC Mediana	0.11 ±0.052	0.11 ±0.054	0.14 ±0.13	0.17 ±0.10
SOMSM Promedio	0.13 ±0.053	0.10 ±0.048	0.19 ±0.19	0.15 ±0.087
SOMSM Mediana	0.12 ±0.049	**0.10 ±0.047**	0.17 ±0.18	0.16 ±0.10
Mejor	0.13 ±0.055	0.11 ±0.056	0.20 ±0.20	0.22 ±0.22
Mejor NARX	0.15 ±0.051	0.11 ±0.048	0.23 ±0.23	0.28 ±0.21
Mejor ARIMA	0.12 ±0.059	0.13 ±0.050	0.22 ±0.17	**0.13 ±0.08**
Mejor kTrayectorias	0.12 ±0.051	0.11 ±0.069	0.20 ±0.16	0.22 ±0.11
Lemke	**0.10 ±0.035**	0.11 ±0.052	0.15 ±0.13	0.22 ±0.12

Al evaluar con RMSE se aprecian diferencias pequeñas entre todos los métodos en especial con los conjuntos NN3 y NN5; por lo que, esta métrica dificulta la selección de un método en particular sobre los otros métodos. En función de los resultados, el método de Lemke obtiene un mejor RMSE con el conjunto NN3, mientras que SOMSM-mediana obtiene el mejor RMSE en el conjunto NN5, el método TVC-promedio obtiene el mejor RMSE con las series sintéticas y en las aplicaciones el mejor ARIMA obtiene el mejor RMSE; estos resultados obedecen al teorema “No-free-lunch”.

Tabla 5.22. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con SMAPE, comparado con otros métodos.

	Conjunto de series de tiempo			
Método	**NN3**	**NN5**	**Sintéticas**	**Aplicaciones**
TVC Promedio	**32 ±17**	26 ±10	**17 ±18**	42 ±29
TVC Mediana	32 ±18	22 ±7.4	19 ±21	44 ±28
SOMSM Promedio	36 ±17	23 ±6.5	30 ±29	44 ±31
SOMSM Mediana	34 ±18	**21 ±7**	26 ±25	44 ±31
Mejor	38 ±21	24 ±8.4	28 ±27	48 ±37
Mejor NARX	44 ±26	23 ±5.9	36 ±35	59 ±38
Mejor ARIMA	35 ±20	30 ±9.1	42 ±41	**40 ±27**
Mejor kTrayectorias	33 ±20	26 ±15	28 ±23	51 ±27
Lemke	32 ±18	23 ±7.8	32 ±30	52 ±30

Al evaluar con SMAPE aumenta la diferencia entre las evaluaciones en comparación con RMSE, esto facilita la elección de un método. Los métodos propuestos obtienen mejor desempeño con la mayoría de las series salvo con las series de aplicaciones donde el método Mejor ARIMA obtiene el mejor resultado.

Tabla 5.23. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con GMRAE, comparado con otros métodos.

	Conjunto de series de tiempo			
Método	**NN3**	**NN5**	**Sintéticas**	**Aplicaciones**
TVC Promedio	0.56 ±0.34	0.66 ±0.43	0.37 ±0.45	**0.53 ±0.44**
TVC Mediana	0.56 ±0.34	0.56 ±0.26	**0.34 ±0.46**	0.54 ±0.41
SOMSM Promedio	0.63 ±0.36	0.60 ±0.31	0.45 ±0.53	0.67 ±0.56
SOMSM Mediana	0.61 ±0.41	**0.54 ±0.28**	0.40 ±0.47	0.64 ±0.51
Mejor	0.70 ±0.47	0.64 ±0.32	0.38 ±0.45	0.95 ±1.10
Mejor NARX	0.73 ±0.34	0.59 ±0.23	0.49 ±0.45	1.00 ±1.00
Mejor ARIMA	0.63 ±0.47	0.76 ±0.27	0.61 ±0.55	0.54 ±0.53
Mejor kTrayectorias	0.57 ±0.32	0.69 ±0.46	0.42 ±0.43	0.76 ±0.61
Lemke	**0.53 ±0.27**	0.57 ±0.25	0.51 ±0.42	0.87 ±0.79

La métrica GMRAE devuelve diferencias entre métodos pequeñas en comparación con SMAPE pero mayores que RMSE. Es interesante que con esta métrica el mejor resultado en las series de aplicaciones lo alcance TVC-promedio, aunque es muy cercano al resultado del Mejor ARIMA, esto se debe a que las series de aplicaciones tienen varios valores extremos y en comparación con las otras series el horizonte de predicción es pequeño.

Tabla 5.24. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con RelMAE, comparado con otros métodos.

	Conjunto de series de tiempo			
Método	**NN3**	**NN5**	**Sintéticas**	**Aplicaciones**
TVC Promedio	0.62 ±0.33	0.75 ±0.44	**0.41 ±0.44**	0.57 ±0.41
TVC Mediana	0.63 ±0.30	0.64 ±0.26	0.43 ±0.48	0.60 ±0.43
SOMSM Promedio	0.70 ±0.31	0.65 ±0.26	0.58 ±0.53	0.57 ±0.41
SOMSM Mediana	0.66 ±0.33	**0.61 ±0.25**	0.53 ±0.46	0.60 ±0.44
Mejor	0.73 ±0.38	0.67 ±0.27	0.63 ±0.60	0.86 ±0.88
Mejor NARX	0.78 ±0.27	0.67 ±0.22	0.73 ±0.66	0.95 ±0.82
Mejor ARIMA	0.69 ±0.40	0.83 ±0.26	0.77 ±0.53	**0.50 ±0.39**
Mejor kTrayectorias	**0.62 ±0.27**	0.70 ±0.33	0.57 ±0.42	0.74 ±0.47
Lemke	0.57 ±0.21	0.65 ±0.22	0.55 ±0.43	0.80 ±0.57

Los resultados de RelMAE al igual que RMSE muestran diferencias pequeñas que dificultan la discriminación entre modelos.

Tabla 5.25. Resultados de TVC y SOMSM combinando con mediana y promedio, evaluación con MASE, comparado con otros métodos.

	Conjunto de series de tiempo			
Método	**NN3**	**NN5**	**Sintéticas**	**Aplicaciones**
TVC Promedio	0.82 ±0.37	0.71 ±0.41	1.2 ±2.0	1.4 ±0.88
TVC Mediana	0.85 ±0.42	0.63 ±0.31	**1.2 ±1.9**	2.0 ±1.60
SOMSM Promedio	0.92 ±0.36	**0.63 ±0.29**	2.0 ±2.6	1.4 ±0.99
SOMSM Mediana	0.86 ±0.37	0.59 ±0.28	2.0 ±2.7	1.8 ±1.40
Mejor	0.98 ±0.50	0.65 ±0.32	2.9 ±4.7	2.5 ±3.10
Mejor NARX	1.10 ±0.47	0.65 ±0.26	3.3 ±4.9	3.7 ±3.80
Mejor ARIMA	0.92 ±0.53	0.81 ±0.30	3.7 ±4.0	**1.2 ±0.51**
Mejor kTrayectorias	0.80 ±0.27	0.69 ±0.43	3.4 ±3.4	2.8 ±2.5.
Lemke	**0.74 ±0.19**	**0.63 ±0.29**	2.3 ±2.8	2.9 ±2.6.

De los resultados observados se confirma que la selección de la métrica varía considerablemente que método alcanza el mejor resultado; sin embargo, una selección apropiada de esta métrica depende de los objetivos del dominio. De aquí podemos concluir que a pesar de que se han realizado varios trabajos para definir métricas generales, se

requiere investigar en una métrica que se adapte automáticamente a las necesidades de los dominios; o en su defecto extender los métodos propuestos para que realicen la selección de los modelos considerando más de una métrica y esto debe ser flexible de acuerdo al dominio.

El método SOMSM mostró en general un mejor desempeño con todas las series, y en particular con las series NN5, las cuales son afectadas por diversos sistemas desconocidos. Como se comentó con anterioridad, la ventaja de incrementar la diversidad de modelos es que cada modelo captura información diferente, entonces al combinarlos se obtiene una buena predicción. Adicionalmente, el mapa generado por SOMSM permite analizar el comportamiento de los modelos entre ellos y su desempeño fácilmente. El método TVC mostró mejor desempeño en las series con cambios bruscos; este tipo de series es difícil de predecir el momento en será el cambio de comportamiento, por lo tanto TVC que sigue un cambio fijo guiado por tres ventanas obtuvo buenos resultados con estas series.

Capítulo 6. Conclusiones

Esta investigación giró alrededor de la búsqueda de mejores métodos para la predicción a largo plazo de series de tiempo, que son difíciles de predecir por métodos convencionales estadísticos sin la ayuda de un experto humano. Como resultado, se proponen dos nuevos métodos de predicción de varios pasos a futuro, basados en la construcción automática de ensambles, usando meta-características y modelos de redes neuronales. El primer método propuesto se denomina TVC, por las siglas en inglés de Combinación Temporal Validada (*Temporal Validated Combination*). Este método analiza el desempeño de los modelos posibles a incluir en el ensamble, usando una validación cruzada Monte Carlo en tres ventanas de predicción; las ventanas de predicción consecutivas se forman dividiendo el horizonte de predicción en tres porciones de igual tamaño, que corresponden con el corto, el mediano y el largo plazo de predicción. Esta estrategia emula a la realizada por expertos humanos.

El segundo método propuesto se denomina SOMSM, por las siglas en inglés de "Mapa Auto-organizado para la Selección de Modelos" (*Self-Organizing Map for Selection of Models*). Este método analiza el desempeño de cada modelo posible en cada instante de tiempo en el horizonte de predicción; el resultado del análisis es una meta-característica representada por un vector, denominado "error representativo", con el cual es posible agrupar los modelos de acuerdo a su comportamiento. Utilizando redes neuronales de mapas auto-organizados y el error representativo de cada modelo, SOMSM construye un mapa de dos dimensiones, donde es posible apreciar el desempeño de los modelos y su diversidad de comportamiento. Inspirados en el teorema de descomposición sesgo-varianza-covarianza (Ueda & Nakano, 1996), SOMSM utiliza el mapa generado para seleccionar un conjunto de modelos de predicción a combinar, tomando en cuenta su desempeño y diversidad de comportamiento. En los experimentos realizados, SOMSM presentó mejores resultados al procesar series de tiempo conocidas como complejas para

predecir, por saberse resultantes de sistemas dinámicos desconocidos. En la predicción de este tipo de series complejas, la diversidad de modelos de predicción permite capturar características diferentes de la serie de tiempo; al combinar estos modelos se obtienen mejores predicciones en promedio, con relación a otros métodos de selección de modelos.

TVC y SOMSM fueron evaluados con un conjunto de treinta y cinco series de tiempo de diferentes tipos, incluyendo series caóticas, no caóticas, extremos de linealidad y no-linealidad, series resultantes de fenómenos presentes en la industria, etc. Los modelos disponibles para la construcción de ensambles se generaron a partir de los modelos base NARX, ARIMA y k-Trayectorias. Los resultados de los métodos propuestos se compararon con los resultados obtenidos por un método basado en meta-características, propuesto por Lemke y Gabrys (2010). Asimismo, se compararon los resultados con los obtenidos por el modelo de mejor desempeño de todos los modelos disponibles. La evaluación del desempeño se realizó usando varias métricas, entre ellas SMAPE. Adicionalmente, se comparó SOMSM con las predicciones obtenidas por el modelo de mejor desempeño por cada familia de modelos de predicción: mejor NARX, mejor ARIMA y mejor k-Trayectorias.

6.1 Conclusiones

Con respecto a las ventajas y desventajas de uso de los métodos propuestos, se encontró que las series de tiempo con las que se recomienda utilizar TVC son aquellas que presentan cambios bruscos de comportamiento, dónde no es posible para un método automático el predecir a tiempo el cambio de comportamiento. La estrategia seguida por TVC permite al ensamble forzar el análisis en tres ventanas definidas, lo que reportó un mejor desempeño que modelos individuales, en los experimentos realizados. Los expertos humanos recomiendan graficar las series, para visualmente determinar si estos cambios bruscos se presentan. En esta investigación se analizó el comportamiento de todas las series utilizadas y se generó automáticamente una regla basada en sus características y en el desempeño de los modelos con las series. A fin de validarla, se aplicó esta regla a todas las series involucradas en los experimentos y se encontró que, al aplicar TVC cuando la regla lo recomendó, el desempeño de TVC fue estadísticamente mejor que el mejor NARX, el

mejor ARIMA y el mejor modelo, evaluando los resultados con las métricas GMRAE y RelMAE. Estos resultados permiten recomendar TVC en este tipo de series, aunque, como en todos los casos, los usuarios siempre deben considerar que los posibles *outliers* de las predicciones no afecten al dominio.

En el análisis experimental, el método SOMSM mostró buenos resultados en general con todas las series de tiempo experimentadas, encontrándose que éste es una buena alternativa cuando no se cuenta un experto humano, que pueda recomendar un tipo de modelo de predicción apropiado para el dominio tratado. También se encontró que el desempeño de SOMSM fue mejor que los otros modelos con que se comparó con las series de tiempo de la competencia NN5. Estas series son en su mayoría caóticas y provienen de una aplicación económica real, que es afectada por sistemas desconocidos. Se observó que con estas series de tiempo, el aumentar la diversidad permitió que los diferentes modelos de predicción consideren las diferentes características de las series de tiempo y combinados obtengan una buena predicción. Con el conjunto NN5, el método SOMSM mostró un desempeño estadísticamente mejor que el mejor modelo ARIMA, el cual es ampliamente utilizado en el dominio económico (Crone, et al., 2008), (De-Gooijer & Hyndman, 2006)

6.2 Aportaciones

Las principales aportaciones de esta investigación son:

- Un nuevo método de predicción de series de tiempo, llamado "Combinación Temporal Validada" (*Temporal Validated Combination, TVC*), basado en ventanas de predicción.

- Un nuevo método de predicción de series de tiempo, llamado "mapa auto-organizado para la selección de modelos" (*Self-Organizing Map for Selection of Models, SOMSM*) basado en meta-características y mapas neuronales auto-organizados.

- Un nuevo método para construir meta-características, llamado "error representativo" que permite adquirir la información de comportamiento de modelos de predicción para la construcción de ensambles, balanceando diversidad y desempeño de los componentes del ensamble.

- Una nueva regla para determinar la pertinencia del uso de TVC, dependiendo de las características de la serie de tiempo involucrada en la predicción.

Los métodos desarrollados pueden trabajar con modelos de predicción de cualquier tipo, gracias a la abstracción que permite la representación a través de la meta-característica propuesta. Los mejores resultados fueron obtenidos sobre series de tiempo con cambios bruscos de comportamiento y con series afectadas por sistemas desconocidos; esto debido a la apropiada selección de modelos de predicción considerando diferentes ventanas de predicción y auto-organizando la selección de modelos, en función de la diversidad de comportamiento y la exactitud de los modelos de predicción.

Los métodos desarrollados cumplieron con el objetivo general originalmente planteado. El conocimiento generado fue reportado en congresos internacionales y en una revista registrada en el índice JCR, como se detalla en el capítulo uno de este documento.

6.3 Trabajo Futuro

Este proyecto presenta la posibilidad de ampliarse en varios aspectos, dentro de las áreas de diseño de ensambles, redes neuronales artificiales y predicción de series de tiempo. Algunas posibilidades se describen a continuación.

Los mapas generados por SOMSM con las series de tiempo utilizadas para la validación de nuestros métodos, sugieren que en algunos casos se forman “islas”, esto es, grupos aislados de modelos que presentan un comportamiento diferente entre sí, pero con un buen desempeño, comparado con los demás modelos de predicción. Se propone analizar con detalle los casos que presentan este comportamiento y extender el método de selección, para mejorar la predicción respectiva.

Es claro que el desempeño de los métodos propuestos depende fuertemente de los modelos disponibles para la construcción de los ensambles. Como trabajo futuro, se propone incluir en los métodos propuestos algoritmos de adaptación para la definición de

los modelos base, sus parámetros y los datos de entrenamiento, de acuerdo a las características particulares de cada serie.

Los patrones que pueden encontrarse en predicciones de varios pasos a futuro, pueden ser utilizados en diversos dominios para la toma de decisiones. Por esa razón, se propone automatizar el análisis de los patrones encontrados, a fin de disparar alertas tempranas en la toma de decisiones en sistemas dinámicos. Ejemplo de uso de esta característica sería para guiar la compra de divisas internacionales, alertas a pacientes de ataques cardiacos, etc.

Existen varios dominios en los que las series de tiempo son multi-dimensionales, es decir, los datos provienen de diversas fuentes y se desea una única predicción, pero basada en varias variables. Un ejemplo muy importante de este caso es la predicción de la velocidad del viento en la superficie terrestre, donde las fuentes que se consideran para la predicción son la presión atmosférica y la temperatura, entre otras variables. La relación entre las fuentes de datos sugiere que, el realizar una predicción conjunta entre los valores del pasado de la velocidad del viento y las otras fuentes de datos, puede presentar mejores resultados que predecir usando únicamente los valores previos de la velocidad del viento. Se propone extender los métodos desarrollados para trabajar con series de tiempo multi-dimensionales.

Creemos que se podría mejorar los resultados si se consideran espacios de búsqueda de mayor tamaño y para manejarlos eficientemente incluir un proceso de optimización basada en sistemas difusos como sugieren (Pulido, et al., 2013) y (Melin, et al., 2012). Adicionalmente, creemos que como trabajo futuro los métodos desarrollados en esta investigación podrían extenderse para ser tolerantes a ruido en los datos incluyendo sistemas difusos tipo 2, como lo reportan en (Gaxiola, et al., 2014). Para las personas que toman decisiones es muy importante poder explicar los argumentos que motivan su decisión; como trabajo futuro también se podría explorar la capacidad de explicar los resultados aprovechando las reglas difusas encontradas.

Referencias

Adhikari, R., 2015. A mutual association based nonlinear ensemble mechanism for time series forecasting. *Applied Intelligence,* 43(2), pp. 233-250.

Adhikari, R. & Agrawal, R., 2012. Forecasting strong Seasonal time series with artificial neural networks. *Journal of Scientific & Industrial Research,* October, Volume 71, pp. 657-666.

Adhikari, R. & Agrawal, R., 2014. A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing and Applications,* 24(6), pp. 1441-1449.

Adhikari, R. & Agrawal, R., 2014. A linear hybrid methodology for improving accuracy of time series forecasting. *Neural Computing and Applications,* 25(2), pp. 269-281.

Adhikari, R. & Verma, G., 2016. Time Series Forecasting Through a Dynamic Weighted Ensemble Approach. In: A. Nagar, P. D. Mohapatra & N. Chaki, eds. *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics: ICACNI 2015, Volume 1*. New Delhi: Springer India, pp. 455-465.

Adya, M., Collopy, F., Armstrong, J. S. & Kennedy, M., 2001. Automatic identification of time series features for rule-based forecasting. *International Journal of Forecasting* , 17(2), pp. 143-157.

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.

Arinze, B., Kim, S.-L. & Anandarajan, M., 1997. Combining and selecting forecasting models using rule based induction. *Computers & Operations Research,* 24(5), pp. 423-433.

Arlot, S. & Celisse, A., 2010. A survey of cross-validation procedures for model selection. *Statistics Surveys,* Volume 4, pp. 40-79.

Armstrong, J. & Collopy, F., 1992. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting* , 8(1), pp. 69-80.

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

Bao, Y., Xiong, T. & Hu, Z., 2014. Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing* , 129(0), pp. 482-493.

Barrow, D. K. & Crone, S. F., 2013. *Crogging (Cross-Validation Aggregation) for Forecasting – a novel algorithm of Neural Network Ensembles on Time Series Subsamples.* s.l., s.n., pp. 214-221.

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., 2015. *Neural Network Toolbox User's Guide R2015a.* s.l.:MathWorks.

Bergmeir, C. & Benitez, J. M., 2012. On the use of cross-validation for time series predictor evaluation. *Information Sciences,* 191(0), pp. 192-213.

Biggs, J. B., 1985. The role of metalearning in study processes. *British Journal of Educational Psychology,* Volume 55, pp. 185-212.

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. pp. 1509-1526.

Brassard, G. & Bratley, P., 1996. *Fundamentals of Algorithmics.* s.l.:Prentice-Hall.

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

Brazdil, P., Carrier, C. G., Soares, C. & Vilalta, R., 2009. *Metalearning: Applications to Data Mining.* s.l.:Springer Science & Business Media.

Breiman, L., 1996. Bagging predictors. *Machine learning,* 24(2), pp. 123-140.

Breiman, L., 1996. Stacked Regressions. *Machine Learning,* Volume 24, pp. 49-64.

Breiman, L., 2000. Randomizing Outputs to Increase Prediction Accuracy. *Machine Learning,* 40(3), pp. 229-242.

Brockwell, P. J. & Davis, R. A., 2006. *Time Series: Theory and Methods.* Second ed. s.l.:Springer.

Brown, G., Wyatt, J. L. & Tiňo, P., 2005. Managing Diversity in Regression Ensembles. *J. Mach. Learn. Res.,* Volume 6, pp. 1621-1650.

Calinski, T. & Harabasz, J., 1974. A dendrite method for cluster analysis. *Communications in Statistics,* 3(1), pp. 1-27.

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.

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. In: N. Jankowski, W. Duch & K. Grabczewski, eds. *Meta-Learning in Computational Intelligence.* s.l.:Springer Berlin Heidelberg, pp. 157-177.

Chandra, A. & Yao, X., 2006. Ensemble Learning Using Multi-Objective Evolutionary Algorithms. *Journal of Mathematical Modelling and Algorithms,* 5(4), pp. 417-445.

Chappell, G. J. & Taylor, J. G., 1993. The temporal Kohonen map. *Neural Networks,* 6(3), pp. 441-445.

Cohen, W. W. & Singer, Y., 1999. *A simple, fast, and effective rule learner.* s.l., s.n., pp. 335-342.

Collopy, F. & Armstrong, J. S., 1992. Rule-Based Forecasting: Development and Validation of an Expert Systems Approach to Combining Time Series Extrapolations. *Management Science,* Volume 38, pp. pp. 1394-1414.

Crone, S. F., 2010. *Competition Instructions.* Disponible en: http://www.neural-forecasting-competition.com/instructions.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.

Deutsch, M., Granger, C. W. J. & 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.

Diaconescu, E., 2008. The use of NARX neural networks to predict chaotic time series. *WSEAS Trans. Comp. Res.,* 3(3), pp. 182-191.

Dzeroski, S. & Zenko, B., 2004. Is Combining Classifiers with Stacking Better than Selecting the Best One?. *Machine Learning,* Volume 54, pp. 255-273.

Fonseca-Delgado, R. & Gomez-Gil, P., 2014. *Selecting and combining models with self-organizing maps for long-term forecasting of chaotic time series.* s.l., s.n., pp. 2616-2623.

Franses, P. H., van Dijk, D. & Opschoor, A., 2014. *Time Series Models for Business and Economic Forecasting.* Second ed. s.l.:Cambridge University Press.

Freund, Y., Schapire, R. E. & others, 1996. *Experiments with a new boosting algorithm.* s.l., s.n., pp. 148-156.

Garcia-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.

García-Rois, J. & Burguillo, J. C., 2015. Topology-based analysis of self-organizing maps for time series prediction. *Soft Computing,* pp. 1-18.

Gaxiola, F., Melin, P., Valdez, F. & Castillo, O., 2014. Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction. *Information Sciences ,* Volume 260, pp. 1-14.

Gibbons, J. D. & Chakraborti, S., 2003. *Nonparametric Statistical Inference.* Fourth ed. s.l.:Marcel Dekker.

Giraud-Carrier, C., 2008. *Metalearning-a tutorial.* s.l., s.n.

Haken, H., 2006. *Information and Self-Organization A Macroscopic Approach to Complex Systems.* Third ed. s.l.:Springer Berlin.

Han, L., Romero, C. E. & Yao, Z., 2015. Wind power forecasting based on principle component phase space reconstruction. *Renewable Energy ,* 81(0), pp. 737-744.

Haykin, S., 1999. *Neural Networks A Comprehensive Foundation.* Second Edition ed. s.l.:Pearson Prentice Hall.

Haykin, S., 2009. *Neural Networks and Learning Machines.* Third ed. Hamilton(Ontario): Pearson Education.

Helbing, D., 2012. *Social Self-Organization Agent-Based Simulations and Experiments to Study Emergent Social Behavior.* s.l.:Springer-Verlag.

Hénon, M., 1976. A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics,* 50(1), pp. 69-77.

Ho, T. K., 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Aug, 20(8), pp. 832-844.

Hyndman, R., 2016. *Time Series Data Library.* Disponible en: https://datamarket.com/data/list/?q=provider:tsdl.

Hyndman, R. J. & Billah, B., 2003. Unmasking the Theta method. *International Journal of Forecasting,* 19(2), pp. 287-290.

Hyndman, R. J. & Koehler, A. B., 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting ,* 22(4), pp. 679-688.

Kalousis, A. & Theoharis, T., 1999. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis,* 3(5), pp. 319-337.

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.

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, Volume 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., 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics,* 43(1), pp. 59-69.

Kohonen, T., 2001. *Self-Organizing Maps.* Third ed. s.l.:Springer-Verlang.

Kohonen, T., 2013. Essentials of the self-organizing map. *Neural Networks*, Volume 37, pp. 52-65.

Koskela, T., Varsta, M., Heikkonen, J. & Kaski, K., 1998. *Temporal sequence processing using recurrent SOM.* s.l., s.n., pp. 290-297 vol.1.

Kourentzes, N., Petropoulos, F. & Trapero, J. R., 2014. Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting,* 30(2), pp. 291-302.

Kugiumtzis, D., 1996. State space reconstruction parameters in the analysis of chaotic time series the role of the time window length. *Physica D: Nonlinear Phenomena* , 95(1), pp. 13-28.

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., Budka, M. & Gabrys, B., 2013. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review,* pp. 1-14.

Lemke, C. & Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. *Neurocomputing,* 73(10-12), pp. 2006-2016.

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.

Martinetz, T., Berkovich, S. & Schulten, K., 1993. Neural-gas network for vector quantization and its application to time-series prediction. *Neural Networks, IEEE Transactions on,* Jul, 4(4), pp. 558-569.

Matijas, M., Suykens, J. A. & Krajcar, S., 2013. Load forecasting using a multivariate meta-learning system. *Expert Systems with Applications*.

May, R. M., 1976. Simple mathematical models with very complicated dynamics. *Nature,* 261(5560), pp. 459-467.

McNames, J., 1998. *A nearest trajectory strategy for time series prediction.* s.l., Katholieke Universiteit Leuven, pp. 112-128.

Meade, N., 2000. Evidence for the selection of forecasting methods. *Journal of Forecasting,* 19(6), pp. 515-535.

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.

Melin, P., Soto, J., Castillo, O. & Soria, J., 2012. A new approach for time series prediction using ensembles of ANFIS models. *Expert Systems with Applications* , Volume 39, pp. 3494-3506.

Mendes-Moreira, J., Soares, C., Jorge, A. M. & Sousa, J. F. D., 2012. Ensemble Approaches for Regression: A Survey. *ACM Comput. Surv.*, Volume 45, pp. 10:1-10:40.

Milligan, G. & Cooper, M., 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika,* 50(2), pp. 159-179.

Miranian, A. & Abdollahzade, M., 2013. Developing a Local Least-Squares Support Vector Machines-Based Neuro-Fuzzy Model for Nonlinear and Chaotic Time Series Prediction. *Neural Networks and Learning Systems, IEEE Transactions on,* Feb, 24(2), pp. 207-218.

Montgomery, D. C., Jennings, C. L. & Kulahci, M., 2008. *Introduction to Time Series Analysis and Forecasting.* s.l.:John Wiley & Sons.

Nau, R., 2015. *Statistical forecasting: notes on regression and time series analysis.* s.l.:s.n.

Ni, H. & Yin, H., 2009. A self-organising mixture autoregressive network for FX time series modelling and prediction. *Neurocomputing,* 72(16-18), pp. 3529-3537.

Ouyang, Y. & Yin, H., 2014. A neural gas mixture autoregressive network for modelling and forecasting FX time series. *Neurocomputing,* 135(0), pp. 171-179.

Papagiannaki, K., Taft, N., Zhang, Z.-L. & Diot, C., 2003. *Long-term forecasting of Internet backbone traffic observations and initial models.* s.l., s.n., pp. 1178-1188.

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.

Picard, R. R. & Cook, R. D., 1984. Cross-Validation of Regression Models. *Journal of the American Statistical Association,* 79(387), pp. 575-583.

Prudêncio, R. B. C. & Ludermir, T. B., 2004. Meta-learning approaches to selecting time series models. *Neurocomputing,* Volume 61, pp. 121-137.

Pulido, M., Melin, P. & Castillo, O., 2013. *Optimization of ensemble neural networks with type-2 fuzzy response integration for predicting the Mackey-Glass time series.* s.l., s.n., pp. 16-21.

Rahman, M., Islam, M., Murase, K. & Yao, X., 2015. Layered Ensemble Architecture for Time Series Forecasting. *Cybernetics, IEEE Transactions on,* PP(99), pp. 1-1.

Ren, Y., Suganthan, P. & Srikanth, N., 2015. Ensemble methods for wind and solar power forecasting - A state-of-the-art review. *Renewable and Sustainable Energy Reviews* , Volume 50, pp. 82-91.

Ren, Y., Zhang, L. & Suganthan, P., 2016. Ensemble Classification and Regression-Recent Developments, Applications and Future Directions. *Computational Intelligence Magazine, IEEE,* Feb, 11(1), pp. 41-53.

Rice, J. R., 1976. The Algorithm Selection Problem* . In: M. Rubinoff & M. C. Yovits, eds. s.l.:Elsevier, pp. 65-118.

Ronco, E. & Gawthrop, P., 1995. Modular neural networks: a state of the art. *Rapport Technique csc-95026.*

Sano, M. & Sawada, Y., 1985. Measurement of the Lyapunov Spectrum from a Chaotic Time Series. *Phys. Rev. Lett.,* Sep, 55(10), pp. 1082-1085.

Sapankevych, N. & Sankar, R., 2009. Time Series Prediction Using Support Vector Machines: A Survey. *Computational Intelligence Magazine, IEEE,* may , 4(2), pp. 24-38.

Sauer, T. D., 2006. *Attractor reconstruction.* Disponible en: http://www.scholarpedia.org/article/Attractor_reconstruction.

Sharkey, A., 1999. Multi-Net Systems. In: A. Sharkey, ed. *Combining Artificial Neural Nets.* s.l.:Springer London, pp. 1-30.

Soman, S., Zareipour, H., Malik, O. & Mandal, P., 2010. *A review of wind power and wind speed forecasting methods with different time horizons.* s.l., s.n., pp. 1-8.

Sorjamaa, A. et al., 2009. Sparse Linear Combination of SOMs for Data Imputation: Application to Financial Database. In: J. C. Principe & R. Miikkulainen, eds. *Advances in Self-Organizing Maps.* s.l.:Springer Berlin Heidelberg, pp. 290-297.

Sorjamaa, A. et al., 2007. Methodology for long-term prediction of time series. *Neurocomputing,* 70(16-18), pp. 2861-2869.

Strickert, M. & Hammer, B., 2005. Merge SOM for temporal data. *Neurocomputing*, Volume 64, pp. 39-71.

Strickert, M., Hammer, B. & Blohm, S., 2005. Unsupervised recursive sequence processing. *Neurocomputing*, Volume 63, pp. 69-97.

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.

Takagi, T. & Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, Jan, Volume SMC-15, pp. 116-132.

Takens, F., 1981. Detecting strange attractors in turbulence. In: D. Rand & L. Young, eds. *Dynamical Systems and Turbulence, Warwick 1980*. s.l.:Springer Berlin Heidelberg, pp. 366-381.

Tan, C., Li, M. & Qin, X., 2008. Random Subspace Regression Ensemble for Near-Infrared Spectroscopic Calibration of Tobacco Samples. *Analytical Sciences*, 24(5), pp. 647-653.

Ting, K. M. & Witten, I. H., 1999. Issues in Stacked Generalization. *Journal of Artificial Intelligence Research*, Volume abs/1105.5466, pp. 271-289.

Ueda, N. & Nakano, R., 1996. *Generalization error of ensemble estimators*. s.l., s.n., pp. 90-95 vol.1.

Vanschoren, J., 2010. *Understanding machine learning performance with experiment databases*, s.l.: s.n.

Vilalta, R. & Drissi, Y., 2002. A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review*, 18(2), pp. 77-95.

Voegtlin, T., 2002. Recursive self-organizing maps. *Neural Networks* , 15(8-9), pp. 979-991.

Vokurka, R. J., Flores, B. E. & Pearce, S. L., 1996. Automatic feature identification and graphical support in rule-based forecasting: a comparison. *International Journal of Forecasting,* 12(4), pp. 495-512.

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.

Yang, Z. R., Lu, W. & Harrison, R. G., 2001. Evolving Stacked Time Series Predictors with Multiple Window Scales and Sampling Gaps. *Neural Processing Letters,* Volume 13, pp. 203-211.

Yu, L., Dai, W., Tang, L. & Wu, J., 2015. A hybrid grid-GA-based LSSVR learning paradigm for crude oil price forecasting. *Neural Computing and Applications,* pp. 1-23.

Yuxia, H. & Hongtao, Z., 2012. Chaos Optimization Method of SVM Parameters Selection for Chaotic Time Series Forecasting. *Physics Procedia ,* 25(0), pp. 588-594.

Zadeh, L. A., 1975. The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences,* Volume 8, pp. 199-249.

Anexo A. Acrónimos

La Tabla A.1 presenta en la primera columna el acrónimo y en la segunda columna muestra la definición respectiva.

Tabla A.1 Lista de acrónimos y definiciones utilizados en esta investigación.

Acrónimo	Definición
ARIMA	*Autoregressive Integrated Moving Average*
BDD	Base de datos
NARX	*Non-lineal Autoregressive with eXogenous inputs*
MSE	*Mean Square Error*
RMSE	*Root Mean Square Error*
MAE	*Mean Absolute Error*
MdAE	*Median Absolute Error*
MAPE	*Mean Absolute Percentage Error*
MdAPE	*Median Absolute Percentage Error*
RMSPE	*Root Mean Square Percentage Error*
RMdSPE	*Root Median Square Percentage Error*
SMAPE	*Symmetric Mean Absolute Percentage Error*
SMdAPE	*Symmetric Median Absolute Percentage Error*
RAE	*Relative Absolute Error*
MdRAE	*Median Relative Absolute Error*
GMRAE	*Geometric Mean Relative Absolute Error*
RelMAE	MAE Relativo
MASE	*Mean Absolute Scaled Error*
SOM	*Self-Organizing Maps*
VRC	*Variance Ratio Criterion*

Anexo B. Lista de Símbolos

La Tabla B.1 presenta en la primera columna el símbolo y en la segunda columna muestra la interpretación respectiva.

Tabla B.1 Lista de variables y definiciones utilizadas en esta investigación.

Símbolo	Definición
h	Horizonte de predicción.
Y	Serie de tiempo.
t	Instante de tiempo.
N	Número de observaciones en una serie de tiempo.
y_t	Valor de la serie de tiempo en un instante de definido.
$\hat{Y}$	Secuencia predicha por un modelo o ensamble de modelos.
$\hat{y}_t$	Valor predicho en un instante de tiempo definido.
m	Número de valores del pasado utilizados para estimar el siguiente valor.
F, f	Función.
$i(x)$	Neurona "ganadora" cuyo vector prototipo es el más semejante a la entrada x.
n_N	Número de neuronas o número de grupos en red SOM.
$w_j(n+1)$	Vector de pesos de la neurona j en la iteración siguiente $(n+1)$.
$\eta(n)$	Coeficiente de aprendizaje en la iteración n.
$v_{j,i(x)}(n)$	Influencia de la neurona ganadora sobre su neurona vecina j en la iteración n.
VRC	Criterio de razón de varianza.
$Trace_B$	Varianza entre grupos.
$Trace_W$	Varianza dentro de los grupos.
M	Número de modelos.
n_E	Número de elementos agrupados.
C_j	Grupo de elementos asociados con la neurona j.
$\bar{\omega}$	Promedio de los vectores centrales de cada grupo, es igual al promedio de los prototipos de las neuronas de la red SOM.
x	Ejemplo o elemento, también utilizado para indicar la coordenada x.

Símbolo	Definición
$\|\|a-b\|\|$	Distancia euclidiana entre los vectores a y b.
$GErr\left(f_{ens}^{(M)}\right)$	Error de generalización de la función ensamble de M modelos.
Bias	Sesgo o diferencia entre el valor esperado y el valor obtenido.
Var	Varianza, media aritmética del cuadrado de las desviaciones respecto a la media.
Covar	Covarianza, es la media aritmética de los productos de las desviaciones de cada una de las variables respecto a sus medias respectivas.
k	Número entero constante, asociado a número de vecinos, número de iteraciones, o número de modelos.
L	Número de puntos en la reconstrucción del espacio de fases.
Δt	Paso de avance en la predicción, generalmente igual a 1.
ρ	Número del meta atributo del método de Lemke y Gabrys.
f_ρ	Característica normalizada del método de Lemke y Gabrys.
s_i	Identificadores de la serie de acuerdo al índice i.
c_i	Modelo identificado por el índice i.
n_S	Número de series.
$Y^{tr(j)}$	Serie de entrenamiento para la iteración j.
$Y^{test(j)}$	Serie de prueba respectiva para la iteración j.
$r(j)$	Tamaño del conjunto de entrenamiento de la iteración j.
r_0	Número mínimo de valores requerido para entrenar un modelo, lo define el usuario.
$\{y^s\}$, $\{y^m\}$, y $\{y^l\}$	Predicción a corto, mediano y largo plazo.
ΔT	Tamaño de las ventanas corto, mediano y largo plazo.
b	Modelo con la mejor expectativa de exactitud, es decir el menor valor de error esperado.
q_b	Expectativa de exactitud del modelo b.
A	Neurona que contiene al modelo b.
l	Número de saltos en el mapa SOM que se tiene que dar para conectar cualquier neurona con la neurona A.
$p\%$	Porcentaje de modelos seleccionados.
α	Factor de tolerancia aplicado en poda del método SOMSM.
C_M	Conjunto de modelos disponibles para el método SOMSM.

Símbolo	Definición
S	Conjunto de posibles tamaños de la red SOM.
$N_C \times N_F$	Número de columnas y número de filas de la red SOM, como utilizamos matrices cuadradas $N_C = N_F$.
c	Identificador de un modelo.
G	Vecindario de la neurona A.
Γ	Lista de modelos que constituyen el ensamble.
N_a	Neurona actual, grupo de modelos actual.
N_p	Grupo de modelos igual al $p\%$ de los modelos con mejor desempeño de Na.
$\mathbb{mm}$	Dimensión del ejemplo de entrada x.
e_t	Error en instante t, diferencia entre el valor predicho y el observado.
p_t	Porcentaje de error en un instante t.
r_t	Error absoluto relativo en instante t.
Q_t	Error escalado en instante t.

Anexo C. Predicción a Largo Plazo

La definición de largo plazo depende de la frecuencia de muestreo y del horizonte de predicción, lo cual a su vez es inherente a cada dominio; a continuación se citan algunos ejemplos donde diversos autores definen largo plazo para sus respectivos dominios:

“Predecir es un problema importante que abarca varios campos incluyendo negocios e industria, gobierno, económicos, ciencias ambientales, medicina, ciencias sociales, políticas, y finanzas. Los problemas de predicción son a menudo clasificados como corto plazo, mediano plazo y largo plazo. Los problemas de predicción a corto plazo involucran predecir eventos únicamente en períodos de tiempo pequeños (días, semanas, meses) en el futuro. Predicciones de mediano plazo se extienden más allá desde uno a dos años en el futuro, y los problemas de predicción a largo plazo pueden extenderse más allá de muchos años”, tomado de la página 1 de(Montgomery, et al., 2008). Estos autores catalogan el horizonte de predicción de acuerdo al tiempo, es importante notar que los datos a los que se refieren son adquiridos cada mes, cada seis meses o cada año.

En la predicción del tráfico de internet se considera mediano plazo un mes y largo plazo seis meses en adelante(Papagiannaki, et al., 2003). En este trabajo las observaciones utilizadas tenían una frecuencia de muestreo de una semana.

En la competencia de predicción NN3, donde utilizan series de tiempo tomadas mensualmente de la industria, se considera corto plazo de uno a tres meses, el mediano plazo va de tres a doce meses, y el largo plazo va de trece a dieciocho meses (Crone, et al., 2011).

En la predicción de la velocidad del viento el corto plazo se considera de 30 minutos a 6 horas, el mediano plazo va de 6 horas a un día, y el largo plazo de 1 día a 1 semana(Soman, et al., 2010). Soman y colaboradores revisaron un conjunto de trabajos, de los cuales las observaciones fueron registradas en diferentes intervalos: los intervalos más pequeños asociados con el corto plazo van entre 5 minutos y 6 horas, por otro lado los intervalos más grandes relacionados con el largo plazo van entre 12 horas y 1 día.

En conclusión el horizonte de predicción puede categorizarse en corto, mediano y largo plazo, donde estas categorías dependen de la frecuencia de muestreo que a su vez está definida de acuerdo a cada dominio.

Anexo D. Ejemplos de Series de Tiempo Caóticas

El siguiente ejemplo muestra la dificultad de predecir series de tiempo caóticas, por esta dificultad el horizonte de predicción está definido de acuerdo al dominio del sistema caótico.

Robert May(May, 1976) plantea que en varios contextos así como en las poblaciones biológicas en particular, existe una tendencia para una variable Y de incrementar de aumentar de una generación a la siguiente cuando es pequeña, y de decrecer cuando es grande. Un ejemplo de ecuación con este comportamiento es la ecuación logística descrito por (May, 1976):

$$y_{t+1} = a y_t (1 - y_t) \tag{D.1}$$

la ecuación exhibe un comportamiento dinámico no trivial si $1 < a < 4$. La Figura D.1 muestra la ecuación logística con $a = 3.8284$, variando t desde 1 a 20. En este ejemplo, se puede ver claramente el notable cambio de comportamiento al variar en una milésima las condiciones iniciales en el tiempo $t = 1$.

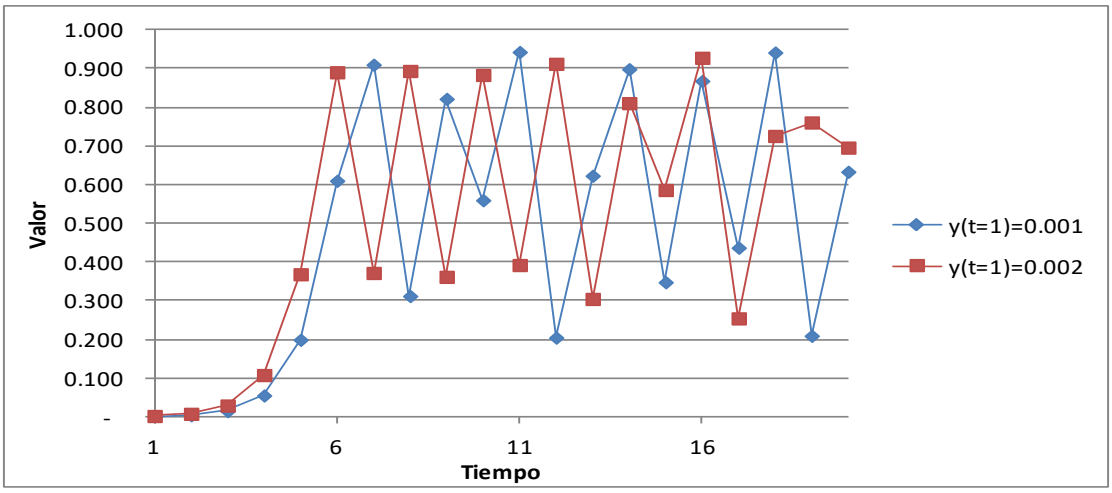


Figura D.1. Ecuación logística iniciando con un pequeño cambio en las condiciones iniciales, la diferencia es de una milésima.

Aún cuando la ecuación que genera los datos está definida, si se inicia con condiciones iniciales ligeramente diferentes, los valores obtenidos divergen notablemente al aumentar el horizonte de predicción; esto nos da la idea intuitiva de que no se pude predecir hacia un futuro indeterminado una serie proveniente de un sistema caótico. En varios dominios que trabajan con sistemas caóticos buscan mejorar sus predicciones consientes de que el horizonte de predicción no puede ser infinito; esta investigación busca encontrar combinaciones de modelos que se adapten a los datos capturados del sistema caótico para predecir hasta horizontes de predicción previamente definidos.

Anexo E. El Error de Predicción Crece Exponencialmente

Dada una predicción $\hat{y}_t$ en un instante t y la observación esperada en el mismo instante y_t, es posible definir un error de predicción en ese instante igual a:

$$e_t = y_t - \hat{y}_t \tag{E.1}$$

En este proyecto de investigación utiliza una predicción iterativa, es decir el valor predicho es utilizado en la predicción del siguiente elemento. Considerando que las series de tiempo caóticas son sensibles a cambios en las condiciones iniciales, podemos argumentar que el error de predicción en un instante es directamente proporcional al error obtenido en el instante anterior.

$$e_t \propto e_{t-1} \tag{E.2}$$

En este mismo sentido podemos asumir que existe una constante $\varepsilon > 1$ de tal modo que:

$$e_t = \varepsilon e_{t-1} \tag{E.3}$$

Siguiendo una demostración iterativa podemos calcular el error en el instante h con respecto al error en el instante inicial $t = 1$.

1) $e_h = \varepsilon e_{h-1}$
2) $e_h = \varepsilon(\varepsilon e_{h-2})$
3) $e_h = \varepsilon\big(\varepsilon(\varepsilon e_{h-3})\big)$ (E.4)
4) ...
5) $e_h = \varepsilon^{h-1} e_1$

Por lo tanto, el error de predicción en series de tiempo caóticas crece exponencialmente con el tamaño del horizonte de predicción.

Anexo F. Tipos de Modelos de Predicción

Existe una amplia variedad de tipos de modelos de predicción, al cambiar sus parámetros principales los modelos pueden tener un comportamiento muy diferente. Este anexo describe brevemente los tipos de modelos de predicción utilizados en este proyecto de investigación, así como los parámetros que se variaron para generar diferentes modelos de predicción.

F.1 Modelos Auto-Regresivos Integrados de Medias Móviles

Los modelos ARIMA, por las siglas del inglés "Autoregressive Integrated Moving Average", representan una generalización de varios modelos lineales. ARIMA puede definirse en base a los modelos Auto-regresivos de Medias Móviles (ARMA por sus siglas en inglés), los cuales expresan el promedio condicional de y_t como una función de p observaciones del pasado $\{y_{t-1}, \dots, y_{t-p}\}$ y q innovaciones pasadas $\{\varepsilon_{t-1}, \dots, \varepsilon_{t-q}\}$. La forma general de un ARMA (p, q) es (Box, et al., 1994):

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1}, \dots, \theta_q \varepsilon_{t-q} \quad \text{(F.1)}$$

donde c es una constante, ε_t representa a un proceso no correlacionado con media cero; ϕ_i y θ_i son valores determinados a partir de los datos. Un modelo ARIMA es obtenido integrando d veces un modelo ARMA. Los parámetros principales de ARIMA(p, d, q) indican el orden de los componentes del modelo: p para el auto-regresivo, d para el de integración y q para el de media móvil. Diferentes valores en estos parámetros dan origen a varios tipos de modelos lineales, como por ejemplo: algunos de los más comunes son (Nau, 2015),(Franses, et al., 2014):

- Auto-regresivo de primer orden ARIMA(1,0,0).
- Caminata aleatoria ARIMA(0,1,0).
- Auto-regresivo diferenciado de primer orden ARIMA(1,1,0).
- Suavizado exponencial simple ARIMA(0,1,1) sin constante:.

- Suavizado exponencial simple con crecimiento ARIMA(0,1,1) con constante.
- Suavizado exponencial lineal ARIMA(0,2,1) o ARIMA(0,2,2) sin constante.
- Suavizado exponencial lineal de amortiguado de tendencia ARIMA(1,1,2) con constante.

F.2 Redes Neuronales NARX

El método de predicción No Lineal Auto-Regresivo con Entradas Exógenas (NARX por sus siglas en inglés) fue propuesto por Leontaritis y Billings y se define como (Leontaritis & Billings, 1985):

$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, \dots, y_{t_0}, u_{t+1}, u_t, u_{t-1}, \dots, u_{t_0}\right) \tag{F.2}$$

La implementación del método NARX utilizando redes neuronales se conoce como red neuronal NARX(Beale, et al., 2015). Escogimos a la red neuronal NARX ya que puede simular, con algunas restricciones, a las redes neuronales completamente conectadas, quienes a su vez pueden simular cualquier sistema dinámico y cualquier máquina de estados(Haykin, 1999). Por otra parte, una red NARX es más fácil de implementar y controlar que una totalmente conectada.

La implementación de NARX en el lenguaje de modelado de datos *Matlab*, utiliza una capa llamada "línea de retardos" (TDL por sus siglas en inglés) cuya entrada es la serie de tiempo y su salida es un vector de dimensión m, construido a partir m observaciones previas (neuronas de retardo)(Beale, et al., 2015). La TDL se conecta con la capa oculta de NARX, la cual puede contener varias neuronas; la capa de salida contiene únicamente una neurona. NARX es entrenada de forma supervisada, utilizando los vectores de dimensión m como entrada y el siguiente valor en la serie de tiempo como salida deseada. Después de que la red está entrenada, la red realiza la predicción de forma recursiva, donde la salida es retro-alimentada a la entrada de la red. El modelo recibe los primeros m valores e itera hasta alcanzar el horizonte de predicción (h). Los parámetros que pueden generar modelos NARX con diferente comportamiento son: el número de neuronas de retardos (m), el número de neuronas en la capa escondida y el algoritmo utilizado para entrenar la red.

F.3 Trayectorias Más Cercanas

La estrategia de las trayectorias más cercanas se basa en la teoría del caos para reconstruir el sistema embebido. Takens mostró que el estado de muchos sistemas dinámicos pueden ser exactamente reconstruidos por una ventana finita de la serie de tiempo(Takens, 1981). Esta ventana se denomina embebido de retardo de tiempo (McNames, 1998):

$$x_t \triangleq \left[y_t, y_{t-\tau}, \dots, y_{t-(m-1)\tau}\right] \tag{F.3}$$

donde m es la dimensión embebida y τ es el embebido de retardo. Los embebidos de retardo de tiempo son ampliamente utilizados como vector de entrada en modelos dinámicos, tanto lineales como no-lineales. El teorema de Takens provee una profunda base teórica para este enfoque; el teorema asume que la serie de tiempo tiene longitud infinita y está libre de ruido, estas condiciones nunca son alcanzadas y la selección de m y τ puede afectar críticamente cuan precisa es la reconstrucción del estado del sistema. El parámetro τ influye en el costo computacional al momento de reconstruir el sistema y m debe ser lo suficientemente grande para capturar el comportamiento del sistema. Kugiumtzis sugiere una estrategia para determinar estos parámetros a través del tamaño de una ventana definido por(Kugiumtzis, 1996):

$$\tau_w = m\tau \tag{F.4}$$

Un buen valor para el tamaño de la ventana, τ_w, puede escogerse por inspección visual de la serie de tiempo. Generalmente la serie tiene oscilaciones aparentemente periódicas. τ_w debe ser escogido para contener varias de estas oscilaciones.

La variante del método de trayectorias más cercanas implementado en este proyecto de investigación crea un conjunto Γ con todas las ventanas posibles de tamaño τ_w de la serie de tiempo, asociando cada ventana con el respectivo siguiente valor esperado en la serie de tiempo. Sacrificando costo computacional en este proyecto fijamos $\tau = 1$, al momento de predecir se reciben una secuencia inicial de τ_w valores iniciales, el modelo busca en Γ las k secuencias más semejantes a la secuencia inicial utilizando una distancia euclidiana; estas secuencias semejantes se denominan trayectorias más cercanas. Luego, los valores

siguientes asociados con cada trayectoria más cercana son promediados para obtener el valor predicho. El proceso de predicción utiliza el valor predicho para estimar el siguiente valor e itera hasta alcanzar el horizonte de predicción. Los parámetros que pueden generar modelos de trayectorias más cercanas con diferente comportamiento son: el tamaño de la ventana, la función de distancia y el número de vecinos.

Anexo G. Características Del Método de Lemke

Las tablas a continuación muestran las características utilizadas en el análisis de series de tiempo realizado en el trabajo(Lemke & Gabrys, 2010).

Tabla G.1. Lista de características estadísticas.

No.	Abreviación	Característica
1	std	*Standard deviation of detrended series*
2	skew	*Skewness of series*
3	kurt	*Kurtosis of series*
4	length	*Length of series*
5	trend	*Standard deviation(series)/standard deviation(detrended series)*
6	dw	*Durbin–Watson statistic of regression residuals*
7	turn	*Turning points*
8	step	*Step changes*
9	pred	*Predictability measure*
10	nonlin	*Nonlinearity measure*
11	lyap	*Largest Lyapunov exponent*

Tabla G.2. Lista en el dominio de frecuencia.

No.	Abreviación	Característica
12-14	ff[1-3]	*Power spectrum frequencies of three biggest values*
15	ff[4]	*Power spectrum: maximal value*
16	ff[5]	*Number of peaks not lower than 60% of the maximum*

Tabla G.3. Lista en el dominio de autocorrelación.

No.	Abreviación	Característica
17-18	acf[1,2]	*Autocorrelations at lags one and two*
19-20	pacf[1,2]	*Partial autocorrelations at lags one and two*
21	Season	*Seasonality measure, pacf[7] for NN5, pacf[12] for NN3*

Tabla G.4. Lista en el dominio de diversidad.

No.	Abreviación	Característica
22	div 1	*Mean(SMAPEs)–mean(SMAPEs deviation from average SMAPE)*
23	div 2	*Mean(SMAPEs)/mean(SMAPEs deviation from average SMAPE)*
24	div 3	*Mean(correlation coefficients in the method pool)*
25	div 4	*Std(correlation coefficients in the method pool)*
26	div 5	*Number of methods in top performing cluster*
27	div 6	*Distance top performing cluster to second best*

Anexo H. Técnicas de Re-Muestreo para la Estimación del Error

La estimación de la exactitud de predicción es importante, tanto para comparar estadísticamente modelos como para evaluar el modelo que será finalmente seleccionado. Dado un conjunto de entrenamiento de datos observados de una población desconocida, y un conjunto de modelos construidos para predecir valores futuros, para conocer qué modelo funciona mejor, el primer paso es calcular el error de predicción de cada modelo. Sin embargo, la exactitud dentro de la muestra (*in-sample*), es decir la habilidad para aproximar los datos, ha probado tener poca correlación con la exactitud en una nueva muestra desconocida (*out-sample*), es decir la habilidad para generalizar sobre nuevos datos del mismo proceso generador(Barrow & Crone, 2013).

La técnica de muestreo estadístico de validación cruzada (CV por sus siglas en inglés) evalúa como los resultados de una estimación estadística generalizarán un conjunto de datos independiente(Arlot & Celisse, 2010). CV separa los datos, dejando un subconjunto como conjunto de entrenamiento para entrenar cada modelo, y la parte remanente como un conjunto de validación para estimar el error del modelo de predicción. Esto provee una estimación menos sesgada y más representativa de la verdadera exactitud del modelo. De múltiples modelos candidatos disponibles como alternativas, el modelo con el menor error de predicción es seleccionado como modelo final.

En los métodos basados en división del conjunto de entrenamiento según (Picard & Cook, 1984) se deben realizar dos consideraciones: (1) Asegurar que se repitan las condiciones en la que el modelo será utilizado, en esta investigación el problema es predecir los siguientes h valores. (2) Considerar donde está localizada la información, al tratar con series de tiempo caóticas, que son sensibles a las condiciones iniciales, es importante siempre tomar en cuenta los valores iniciales. Existen diferentes métodos de validación cruzadas: *hold-out* CV, k-pliegues CV, dejar uno fuera CV y Monte Carlo CV; estos se explican a continuación enfocados al problema de esta investigación.

H.1 *Hold-out* CV

Es el caso más simple de CV; la serie de tiempo es dividida una sola vez en dos secuencias mutuamente exclusivas: una para entrenar el modelo Y^{tr} y otra con los últimos h valores para validarlo Y^{test}. El modelo ajustado con el conjunto de entrenamiento $c_{Y^{tr}}$ predice una secuencia de h de valores futuros $\hat{Y}$; la diferencia entre la secuencia de validación y la predicha es evaluada con la métrica previamente definida Q, el resultado de *Hold-out* CV es $Q\left(Y^{test}, \hat{Y}\right)$. La *Hold-out* CV computacionalmente solo demanda un proceso de entrenamiento y uno de predicción, pero por realizar solo una evaluación presenta una varianza alta.

H.2 *k*-pliegues CV

La serie de tiempo es dividida en dos secuencias; la primera contiene un número mínimo de observaciones requeridas para entrenar el modelo r_0 y la secuencia V que a su vez es dividida k secuencias mutuamente exclusivas (pliegues) de tamaño h. En la primera iteración el modelo c es entrenado utilizando las observaciones requeridas r_0 y predice los siguientes h valores, la predicción es evaluada utilizando Q y el primer pliegue $V(1)$, en la siguiente iteración el modelo c es entrenado utilizando la unión del conjunto de entrenamiento anterior con el siguiente pliegue $r_0 \cup V(1)$ y las predicciones respectivas son evaluadas con la misma función Q y el segundo pliegue $V(2)$, este proceso se repite con todos los pliegues. El resultado de *k*-pliegues CV es el promedio de las k evaluaciones medidas con la función Q. La varianza de *k*-pliegues CV es menor que la de *Hold-out* CV, aunque incrementa el costo computacionales dependiendo del número de evaluaciones.

H.3 Dejar Uno Fuera CV

Es el caso extremo de *k*-pliegues CV donde el número *k* es determinado por el número de observaciones disponibles en la secuencia de validación V. Dejar Uno Fuera CV realiza muchas más evaluaciones que un clásico 10-pliegues CV y por lo tanto se acerca más al error real del modelo de predicción. El número de evaluaciones es el parámetro más influyente en la complejidad computacional de CV, entonces la complejidad computacional de Dejar Uno Fuera CV es mucho mayor que la de 10-pliegues CV.

H.4 Monte Carlo CV

Divide la secuencia de validación aleatoriamente permitiendo repeticiones; así se pueden evitar sesgos inducidos por el método de validación. La Figura H.1 muestra el proceso, en cada iteración j existe un conjunto de entrenamiento $Y^{tr(j)}$ que tiene los primeros r_0 más un número aleatorio de valores de la serie de tiempo; el conjunto de prueba $Y^{test(j)}$ tiene los siguientes h valores. En cada iteración el modelo a validar es entrenado con la secuencia $Y^{tr(j)}$, una vez entrenado predice los siguientes h valores. La predicción es evaluada con la función Q y la secuencia de prueba respectiva; MCCV retorna el promedio de las evaluaciones de todas las iteraciones. El número de iteraciones lo define el usuario del método.

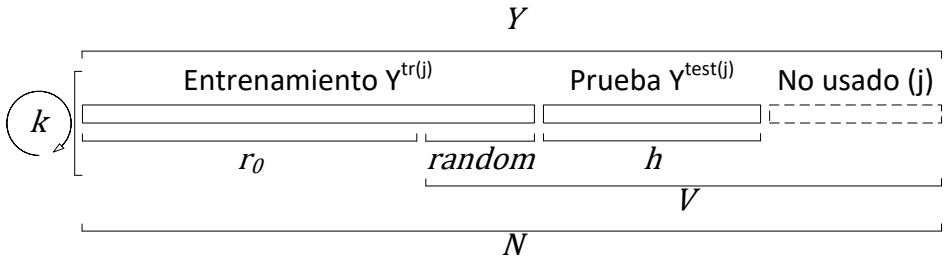


Figura H.1. Esquema de funcionamiento de MCCV aplicado con series de tiempo para predecir *h* valores.

Anexo I. Impacto de la Métrica en la Selección de Modelos en TVC

Esta sección muestra los resultados obtenidos por TVC al utilizar diferentes métricas para seleccionar los modelos con mejor desempeño; las filas corresponden a la función de combinación y la función utilizada como métrica de selección, las columnas corresponden a: (*k*) el número de modelos seleccionados en promedio y los resultados promedio evaluados con las métricas RMSE, SMAPE, GMRAE, RelMAE y MASE. Los valores mínimos están marcados en negrita.

Tabla I.1. Impacto de la métrica para seleccionar modelos en TVC, combinando con promedio o mediana en las series NN3.

		NN3					
F. Combinación	**F. Selección**	***k***	**RMSE**	**SMAPE**	**GMRAE**	**RELMAE**	**MASE**
Promedio	RMSE	40.73 ± 55.79	0.12 ± 0.05	35.74 ± 19.00	0.69 ± 0.45	0.71 ± 0.38	0.91 ± 0.38
Mediana	RMSE	40.27 ± 52.01	0.12 ± 0.06	33.16 ± 19.18	0.61 ± 0.36	0.66 ± 0.33	0.86 ± 0.37
Promedio	SMAPE	26.82 ± 38.03	**0.11 ± 0.04**	32.32 ± 17.60	**0.56 ± 0.35**	**0.62 ± 0.34**	**0.82 ± 0.38**
Mediana	SMAPE	44.36 ± 50.23	0.11 ± 0.05	**31.98 ± 19.02**	**0.56 ± 0.35**	0.63 ± 0.31	0.85 ± 0.45
Promedio	GMRAE	**13.09 ± 17.95**	0.12 ± 0.06	34.19 ± 20.69	0.62 ± 0.45	0.67 ± 0.37	0.89 ± 0.46
Mediana	GMRAE	33.27 ± 42.77	0.12 ± 0.06	34.37 ± 20.13	0.66 ± 0.47	0.69 ± 0.38	0.92 ± 0.47
Promedio	RELMAE	19.27 ± 23.72	0.12 ± 0.06	34.24 ± 21.17	0.63 ± 0.38	0.65 ± 0.31	0.88 ± 0.44
Mediana	RELMAE	32.73 ± 44.18	0.12 ± 0.07	33.31 ± 22.08	0.61 ± 0.40	0.63 ± 0.33	0.85 ± 0.45
Promedio	MASE	30.00 ± 46.63	0.12 ± 0.06	33.87 ± 20.79	0.60 ± 0.36	0.63 ± 0.31	0.85 ± 0.42
Mediana	MASE	36.45 ± 46.32	0.12 ± 0.06	33.53 ± 20.88	0.58 ± 0.40	0.64 ± 0.33	0.87 ± 0.46

Tabla I.2. Impacto de la métrica para seleccionar modelos en TVC, combinando con promedio o mediana en las series NN5.

		NN5					
F. Combinación	**F. Selección**	***k***	**RMSE**	**SMAPE**	**GMRAE**	**RELMAE**	**MASE**
Promedio	RMSE	19.64 ± 27.05	**0.11 ± 0.05**	23.99 ± 8.15	0.61 ± 0.28	0.67 ± 0.25	0.66 ± 0.31
Mediana	RMSE	40.18 ± 41.76	0.11 ± 0.06	23.63 ± 8.24	0.59 ± 0.28	0.67 ± 0.26	0.66 ± 0.32
Promedio	SMAPE	30.09 ± 46.99	0.12 ± 0.07	25.60 ± 10.58	0.66 ± 0.45	0.75 ± 0.46	0.71 ± 0.43
Mediana	SMAPE	24.55 ± 28.37	0.11 ± 0.06	22.24 ± 7.77	0.56 ± 0.27	0.64 ± 0.27	0.63 ± 0.32
Promedio	GMRAE	24.27 ± 41.61	0.12 ± 0.05	24.50 ± 7.82	0.61 ± 0.28	0.70 ± 0.25	0.68 ± 0.30
Mediana	GMRAE	44.64 ± 60.60	0.12 ± 0.06	25.54 ± 9.82	0.63 ± 0.32	0.72 ± 0.29	0.69 ± 0.33
Promedio	RELMAE	21.82 ± 36.72	**0.11 ± 0.05**	22.55 ± 7.33	**0.56 ± 0.25**	**0.64 ± 0.20**	**0.62 ± 0.27**
Mediana	RELMAE	32.55 ± 32.66	0.11 ± 0.06	**22.13 ± 7.29**	**0.56 ± 0.25**	0.64 ± 0.25	0.62 ± 0.31
Promedio	MASE	**8.36 ± 9.86**	**0.11 ± 0.05**	23.28 ± 7.95	0.58 ± 0.27	0.66 ± 0.24	0.64 ± 0.30
Mediana	MASE	15.55 ± 22.98	0.10 ± 0.06	22.44 ± 8.08	0.58 ± 0.31	0.64 ± 0.28	0.62 ± 0.33

Tabla I.3. Impacto de la métrica para seleccionar modelos en TVC, combinando con promedio o mediana en las series Sintéticas.

Sintéticas							
F. Combinación	**F. Selección**	***k***	**RMSE**	**SMAPE**	**GMRAE**	**RELMAE**	**MASE**
Promedio	RMSE	**18.43 ± 19.92**	**0.12 ± 0.12**	17.17 ± 19.16	0.37 ± 0.48	**0.40 ± 0.47**	1.20 ± 2.10
Mediana	RMSE	24.00 ± 39.06	0.13 ± 0.12	17.48 ± 19.98	0.33 ± 0.49	0.41 ± 0.48	1.18 ± 2.03
Promedio	SMAPE	24.00 ± 31.83	0.13 ± 0.12	17.39 ± 19.46	0.37 ± 0.48	0.41 ± 0.48	1.23 ± 2.17
Mediana	SMAPE	29.00 ± 54.10	0.14 ± 0.14	19.01 ± 23.10	0.34 ± 0.50	0.43 ± 0.52	1.20 ± 2.03
Promedio	GMRAE	20.57 ± 38.18	0.27 ± 0.25	70.49 ± 81.71	0.36 ± 0.50	0.80 ± 0.60	4.47 ± 5.96
Mediana	GMRAE	27.00 ± 53.01	0.28 ± 0.25	70.50 ± 82.18	0.33 ± 0.51	0.79 ± 0.61	4.26 ± 5.86
Promedio	RELMAE	24.57 ± 33.00	0.13 ± 0.12	17.63 ± 19.76	0.37 ± 0.49	0.42 ± 0.49	1.22 ± 2.13
Mediana	RELMAE	28.43 ± 50.67	0.14 ± 0.14	19.25 ± 23.12	0.35 ± 0.51	0.44 ± 0.53	1.21 ± 2.03
Promedio	MASE	23.43 ± 33.29	0.13 ± 0.12	**17.16 ± 19.39**	0.33 ± 0.44	0.40 ± 0.48	1.17 ± 1.99
Mediana	MASE	26.71 ± 50.91	0.13 ± 0.14	18.71 ± 22.85	**0.32 ± 0.48**	0.43 ± 0.52	**1.17 ± 1.95**

Tabla I.4. Impacto de la métrica para seleccionar modelos en TVC, combinando con promedio o mediana en las series de Aplicaciones.

Aplicaciones							
F. Combinación	**F. Selección**	***k***	**RMSE**	**SMAPE**	**GMRAE**	**RELMAE**	**MASE**
Promedio	RMSE	35.50 ± 61.57	0.16 ± 0.16	**40.76 ± 35.45**	0.53 ± 0.57	0.60 ± 0.62	1.64 ± 2.05
Mediana	RMSE	**24.17 ± 49.42**	0.19 ± 0.15	44.66 ± 33.99	0.57 ± 0.54	0.67 ± 0.60	2.14 ± 2.06
Promedio	SMAPE	55.83 ± 77.32	**0.15 ± 0.10**	41.81 ± 31.68	**0.53 ± 0.48**	**0.57 ± 0.45**	**1.40 ± 0.96**
Mediana	SMAPE	24.67 ± 45.35	0.17 ± 0.11	43.96 ± 31.00	0.54 ± 0.45	0.60 ± 0.47	1.98 ± 1.73
Promedio	GMRAE	89.00 ± 61.22	0.41 ± 0.59	72.01 ± 68.24	1.62 ± 2.25	1.55 ± 2.27	3.79 ± 3.91
Mediana	GMRAE	59.50 ± 49.83	0.19 ± 0.13	50.75 ± 36.58	0.71 ± 0.67	0.70 ± 0.55	2.25 ± 1.99
Promedio	RELMAE	75.67 ± 55.53	0.21 ± 0.11	51.84 ± 35.65	0.69 ± 0.62	0.71 ± 0.51	2.32 ± 1.80
Mediana	RELMAE	59.00 ± 49.39	0.19 ± 0.13	48.86 ± 35.30	0.68 ± 0.64	0.70 ± 0.56	2.25 ± 2.01
Promedio	MASE	69.33 ± 74.44	0.18 ± 0.13	46.45 ± 36.45	0.57 ± 0.51	0.63 ± 0.51	1.60 ± 1.27
Mediana	MASE	42.67 ± 45.76	0.19 ± 0.14	47.01 ± 35.75	0.55 ± 0.45	0.64 ± 0.52	2.04 ± 1.82

En estas tablas se puede apreciar que los resultados cambian de acuerdo a la métrica que guío la selección de los modelos, de aquí que la selección de la métrica depende del dominio del problema. Con la finalidad de obtener resultados comparables con otros trabajos de investigación, en esta investigación la métrica utilizada para seleccionar modelos es SMAPE.