



INSTITUTO POLITÉCNICO NACIONAL



CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

*"TÉCNICA DE APRENDIZAJE DE SERIES DE TIEMPO
ESTACIONARIAS A PARTIR DE LA INFORMACIÓN DE SU
IMAGEN"*

Tesis que para obtener el grado de:

Maestro en Ciencias de la Computación

Presenta:

Ing. Karinne Ramírez Amaro

Director de Tesis:

Dr. Juan Carlos Chimal Eguía

México, D.F., Noviembre 2007

Índice general

1. Introducción	3
1.1. Planteamiento del problema	4
1.2. Solución propuesta	6
1.3. Objetivos	7
1.3.1. Objetivos generales	7
1.3.2. Objetivos específicos	7
1.4. Contribución	8
1.4.1. Trabajos publicados	8
1.5. Organización	10
2. Marco teórico	13
2.1. Series de tiempo	13
2.1.1. Objetivos del análisis de series de tiempo	15
2.1.2. Técnicas para la descripción de las series de tiempo	15
2.1.3. Comportamientos dinámicos de las series de tiempo	16
2.1.4. Series de tiempo estacionarias	18
2.2. Aprendizaje de series de tiempo	21
2.2.1. Redes neuronales	23
2.2.2. Máquinas de soporte vectorial (Support Vector Machines)	24
2.3. Predicción de series de tiempo	25
2.3.1. Técnica de predicción con Multi-paso.	26
3. Descripción de la técnica de aprendizaje basado en imagen	31
3.1. Introducción	31
3.2. Representación de los datos de entrada	32
3.2.1. Vector de intervalos	34
3.2.2. Método de submetas aplicado a series de tiempo	36
3.3. Técnica de aprendizaje de ST	44
3.3.1. Cálculo del error de aprendizaje	58
3.4. Comparación con otras técnicas de aprendizaje	59
4. Aplicaciones de la técnica	63
4.1. Predicción de series de tiempo	63
4.1.1. Información almacenada para la predicción	64
4.1.2. Pasos para realizar la predicción	70

5. Análisis de resultados	81
5.1. Descripción del conjunto experimental	81
5.2. Resultados del aprendizaje	99
5.2.1. Primera tipo: anchura constante	101
5.2.2. Segundo tipo: división por frecuencias	110
5.2.3. Tercera tipo: división por porcentaje de información total	116
5.3. Análisis de los resultados del seguimiento de ST	129
5.4. Resultados de la predicción	133
5.4.1. Primer tipo: división por anchura	135
5.4.2. Segundo tipo: división por frecuencias	141
5.4.3. Tercer tipo: división por porcentaje de información total	147
5.5. Comparación de los resultados de predicción con otras técnicas	161
5.5.1. Máquinas de soporte vectorial vs. Predicción basada en imagen	162
5.5.2. Redes neuronales vs. Predicción basada en imagen	165
6. Conclusiones y trabajo futuro	167
6.1. Trabajo a futuro	168
6.1.1. Mapeo de series de tiempo a redes complejas	168
6.1.2. Agregar información global a la predicción basada en imagen	171
7. Artículos Publicados	177

Índice de figuras

1.1. Representación de los datos de entrada de dos técnicas de “Machine Learning” para el aprendizaje del comportamiento dinámico de series de tiempo. En a) se observa el vector de entrada para la técnica de redes neuronales y en b) se observa el vector de entrada para la técnica de máquinas de soporte vectorial.	5
1.2. Se observa la nueva representación de los datos de entrada de la técnica de aprendizaje propuesta, ésta se obtiene a través del análisis de la imagen de la serie de tiempo. Observando la figura podemos notar que el recuadro amarillo representa el vector de entrada, el cual, corresponde a los puntos que pertenecen dentro de la caja 2 en el primer instante (visita 1), posteriormente, observamos un segundo recuadro amarillo, el cual, indica que estamos analizando los puntos que visitan la caja 2 por segunda ocasión (visita 2) y así sucesivamente. La información contenida en cada caja va cambiando según evoluciona en el tiempo esta serie.	6
2.1. Ejemplo de una serie de tiempo discreta, donde se observa cómo la función está descrita por puntos discretos, $x(t_1), x(t_2), \dots, x(t_n)$	14
2.2. Ejemplo de la prueba de estacionaridad. En a) se observa la ST Ruido Blanco y como se subdivide en ventanas de 100 elementos sobre el eje del tiempo, en b) se observa la distribución de la media de cada ventana y en c) se observa la varianza de cada ventana, se puede observar que siguen una distribución uniforme en ambos casos.	20
2.3. Ejemplo cuando no cumple con la prueba de estacionaridad. En a) se muestra la ST “Down Jones” dividida en 10 ventanas de 500 puntos cada una, en b) se observa la media de los puntos de cada ventana y en c) se observa la varianza de los puntos de cada ventana. Como se observa en b) y c) todas las ventanas tienen distinta distribución de puntos al variar el tiempo, por lo tanto la ST es no estacionaria.	20
2.4. Ejemplo de arquitectura clásica para una red neuronal con tres entradas (valores de la ST con cierto retardo, esto es, x_{t-1} y x_{t-4} y una constante) una capa oculta de dos neuronas y una salida (por ejemplo el punto predicho).	24

2.5.	Ejemplo de cómo se lleva paso a paso la predicción iterada, a) primero se toman los últimos patrones aprendidos o entrenados (puntos negros) y posteriormente se genera la predicción de un punto (punto rojo), b) representa la nueva entrada de patrones el cual incluye el último predicho y los últimos tres patrones entrenados y a partir de esta nueva entrada se genera el nuevo punto predicho, c) representa el siguiente patrón de entrada, el cual ahora tiene la mitad de puntos predichos en etapas anteriores y la otra mitad representa los últimos dos patrones entrenados y este proceso se realiza hasta que se cumpla con los puntos que se desean predecir y como se observa en d) el patrón de entrada eventualmente estará conformado por todos los puntos predichos.	27
3.1.	Ejemplo de la definición de función como una gráfica.	32
3.2.	Ejemplo de una caja $[Box] = [\underline{x}_1, \overline{x}_1] \times [\underline{y}_1, \overline{y}_1]$ de $I\mathbb{R}^2$	34
3.3.	Ejemplo de como se encapsulan los puntos dentro de una caja y cómo a partir de ésta se obtiene el $\text{área}_f [Box]$ correspondiente. Además se observa que para esta figura el $\text{área}_f [Box] = 86$ puntos.	35
3.4.	Secuencia de puntos que conforman una ST. Se observa que la imagen de la ST está dividida en 10 <i>cajas</i> de igual anchura, esto es, $a [y_1, y_2] = 0.1$. Es debido a esto, que las cajas para esta ST serían las siguientes $[0, 250] \times [0, 0, 1] \times [0, 1, 0, 2] \times [0, 2, 0, 3] \times \dots \times [0, 9, 1]$ y como se observa los intervalos que cambian pertenecen a la imagen de la ST.	36
3.5.	Ejemplo de como se dividió la ST con respecto a sus amplitudes. Como se puede observar en este análisis la información con la que trabajamos es la que se encuentra dentro de cada <i>caja</i> la cual representa la información únicamente de la amplitud.	37
3.6.	Ejemplo de la normalización de una ST. En a) se observa la ST “Lorenz” original, la cual se observa que los valores de su imagen están dentro del intervalo $[-20, 20]$ y en b) se observa la serie de tiempo normalizada dentro del intervalo $[-1, 1]$. Es claro observar como la dinámica de la serie es preservada en a) y b).	39
3.7.	a) ST dividida en 10 <i>cajas</i> con $a [y_1, y_2] = 0.1$ y b) Distribución Gaussiana de los puntos de la ST dentro de cada <i>caja</i> . Como se observa dentro de las <i>cajas</i> 5 y 6 se tiene la mayor frecuencia de puntos y en las <i>cajas</i> 1, 2 y 10 se observa la menor frecuencia de puntos.	40
3.8.	Diagrama de flujo utilizado para el cálculo de las cotas superior e inferior de las nuevas <i>cajas</i> con distribución uniforme de puntos.	42
3.9.	Ejemplo de la división de la imagen de f de acuerdo al nuevo principio definido por área_f . a) Se observan las nuevas <i>cajas</i> enumeradas del 1 al 10, en donde las centrales están más estrechas que las del los costados. b) Se observa la distribución que se obtiene de las nuevas divisiones, la cual es uniforme.	44
3.10.	Acercamiento de los primeros 150 puntos de la ST original, la cual como se observa en a) ésta consta de más de 16,000 puntos y en b) se observa este acercamiento de la ST “Qperiodic2”.	46

3.11. Proceso de aprendizaje. En a) se observa cómo se van obteniendo los puntos de la ST original para posteriormente en b) extraer la información de la caja a la que pertenece dicho punto y el sentido de éste. En c) se observa el procedimiento aleatorio para aprender el comportamiento dinámico de la ST, así como el cálculo del error entre el valor generado menos el original, después en d) se observa como se almacena el valor generado en su matriz correspondiente y finalmente en e) se observa como se almacena el error obtenido. 47

3.12. Ejemplificación de las matrices utilizadas para almacenar los valores generados con la técnica de aprendizaje. A la izquierda se observan los puntos calculados de la ST, donde los primeros tres puntos tienen una tendencia descendente y los otros ascendente, a la derecha se muestra el almacenamiento de los valores calculados dentro de la matriz A, finalmente en la parte inferior derecha se observa cómo obtener los datos almacenados en esta matriz. 48

3.13. Proceso de aprendizaje de los primeros 8 puntos. En a) se observa como se van obteniendo los puntos de la ST original para posteriormente en b) extraer la información de la caja a la que pertenece dicho punto y el sentido de éste. En c) se observa el procedimiento aleatorio para aprender el comportamiento dinámico de la ST, así como el cálculo del error entre el valor generado menos el original, después en d) se observa como se almacena el valor generado en su matriz correspondiente y finalmente en e) se observa como se almacena el error obtenido. 50

3.14. Ejemplo de cómo se van generando las matrices de datos Ua y la de errores Ue . En este caso, debido a que únicamente se han calculado los primeros 9 puntos, la primera columna de ambas matrices muestra los resultados obtenidos para la caja 1 y la segunda columna muestra los resultados de los primeros dos puntos dentro de la caja 2. 51

3.15. Esta figura muestra paso a paso cómo se lleva a cabo el proceso de aprendizaje, el inciso a) nos muestra un acercamiento de los primeros puntos de la ST a analizar, además se observa en un rectángulo los puntos que pertenecen a la caja 2 durante la primera visita cuando tienen el sentido ascendente (puntos negros), b) nos muestra la estructura de las matrices que utilizamos para almacenar la información, c) muestra el cálculo de los primeros puntos estimados, d) muestra cómo se almacenan los resultados en su respectiva matriz, los cuales se observan en a) como puntos grises, en e) se observa la segunda visita que hacen los puntos sobre la misma caja, posteriormente en f) se ajustan los nuevos intervalos de decisión y los resultados finalmente se almacenan en g). 53

3.16. Ejemplo de cómo se reajustan los valores calculados con la técnica propuesta dentro de la matriz Ua . En la Matriz de la izquierda observamos dentro del recuadro rojo el valor del punto ($i = 8$) calculado durante la primera visita, mientras que en la Matriz de la derecha se observa el recuadro rojo, el que ahora pertenece al primer valor generado durante la segunda visita ($i = 126$) y se observa que éste obtuvo un valor más pequeño que la visita anterior. 54

3.17. Diagrama de flujo del proceso de aprendizaje del comportamiento dinámico de series de tiempo. 55

3.18. Resultado del aprendizaje de la ST cuando está decrecentando, esto es, cuando $band = 0$. En a) se observa la matriz $Ua(:, :, 0)$ que se refiere a los puntos que fueron aprendidos y en b) se observa la matriz de errores $Ue(:, :, 0)$	56
3.19. Resultado del aprendizaje de la ST cuando su bandera es igual a 1, es decir está incrementando. En a) se observa la matriz $Ua(:, :, 1)$ que almacena los valores generados y en b) se observa la matriz $Ue(:, :, 1)$ la cual almacena sus respectivos errores.	57
3.20. Ejemplo del proceso de aprendizaje del comportamiento de la serie de tiempo "Seno".	58
3.21. Esquema de cómo funcionan las redes neuronales y de cómo se trabajan los puntos de la serie de tiempo durante su proceso de aprendizaje. Además, se observa el vector de entrada correspondiente a dicha red neuronal.	60
3.22. Esquema explicativo de cómo funcionan las máquinas de soporte vectorial y cómo se obtiene el vector de entrada de los datos de la serie de tiempo para su aprendizaje.	61
3.23. Esquema de TABI y también el vector de entrada de los puntos de la serie de tiempo en ventanas de su imagen para su aprendizaje.	62
4.1. Ejemplo de cómo se obtiene la longitud de los puntos (recuadro naranja) de la serie de tiempo durante el proceso de aprendizaje.	65
4.2. Esta figura muestra cómo se genera la matriz <i>longitud</i> . En a) se observa la ST dividida en 10 <i>cajas</i> , además se observan las visitas y las longitudes durante cada visita y en b) se observa el llenado de la matriz <i>longitud</i> cuando los puntos de la ST están subiendo, esto es, con $band = 1$	66
4.3. Se muestra cómo se genera la matriz <i>saltos</i> . En a) se observan los puntos de la ST y las <i>cajas</i> que éstos van visitando y en b) se observa como se llena la matriz <i>saltos</i> cuando los puntos de la ST tienen $band = 1$	67
4.4. Fusión de las matrices <i>longitud</i> y <i>saltos</i> . En a) se observa la matriz <i>longitud</i> , en b) se observa la matriz <i>saltos</i> y en c) se observa la fusión de a) y b). Se observa que se encerró dentro de un círculo azul un ejemplo de cómo se construyó la nueva matriz <i>longSaltos</i> a partir de las matrices a y b, donde $longSaltos(7, 2, 1, 1) = 1$	68
4.5. Obtención de la matriz <i>longSaltos</i> después de terminar el proceso de aprendizaje (entrenamiento). En a) se observa la serie de tiempo original, en b) se observan los primeros 500 puntos de la serie de tiempo, también se observan dos rectángulos uno de color morado el cual indica la situación de los puntos que estando dentro de la caja 5 brincan a la caja 6, y la caja verde indica la situación en donde apesar de estar en la misma caja 5 los siguientes puntos brincan a la misma caja 5 pero cambian su sentido, entonces, en c) se observa el resultado de la matriz <i>longSaltos</i> y se enfatizan las dos situaciones anteriores mencionadas con los colores respectivos.	69
4.6. Resultado de buscar en la matriz <i>longSaltos</i> la frecuencia de los puntos en la <i>caja</i> 10 y con el sentido de los puntos ascendente. Dentro de la matriz <i>longSaltos</i> $(:, :, 10, 1)$ se observan las frecuencias de los puntos con longitud igual a 9 puntos, la cual se encuentra encerrada dentro de un rectángulo azul y éste es igual a 16 cuando brinca a la siguiente <i>caja</i> 10.	71

4.7. Código para extraer el vector <i>extract</i> de la matriz <i>longSaltos</i> ($l, :, k, i$).	72
4.8. Se muestran tres posibles situaciones en las que el sentido de los puntos se cambia.	73
4.9. Código para generar la distribución de probabilidades de cada futura <i>caja</i>	74
4.10. Un ejemplo de cómo se genera la matriz <i>table</i> . Se observa que en la primera columna se almacena la(s) caja(s) a la(s) que posiblemente brinquen los puntos, en la columna dos se almacena la frecuencia de la(s) caja(s), en la columna tres se acumula la probabilidad de ocurrir de cada caja y finalmente en la cuarta columna se almacena la suma de probabilidades de la(s) caja(s) visitada(s) por los puntos durante el proceso de aprendizaje.	75
4.11. Código para escoger de manera aleatoria ponderada el valor de la siguiente <i>caja</i> a la que los siguientes puntos van a brincar.	76
4.12. Pasos que se siguieron para elegir la longitud de los siguientes puntos. Se observa que en a) se tiene la matriz que se extrae de <i>longSaltos</i> , b) muestra cómo se genera el vector <i>suma</i> y c) muestra la tabla de probabilidades para cada longitud pasada.	77
4.13. Esquema de los pasos a seguir para la predicción de series de tiempo. Como se observa el primer paso es extraer la última información de los puntos de la serie de tiempo, posteriormente se estima la siguiente caja a brincar, después se debe de estimar la siguiente longitud de los puntos y finalmente buscar en la matriz <i>Ua</i> los puntos con las características estimadas y por último se dibujan estos puntos estimados y este proceso se repite hasta se cumpla con el número de puntos que se desea predecir.	79
5.1. Serie de tiempo “Seno”.	82
5.2. Serie de tiempo “Vanderpol”.	83
5.3. Serie de tiempo “Qperiodic2”.	83
5.4. Serie de tiempo “Qperiodic3”.	84
5.5. Serie de tiempo “Mackey & Glass”.	84
5.6. Serie de tiempo “Logistic”.	85
5.7. Serie de tiempo “Lorenz”.	86
5.8. Serie de tiempo “Rossler”.	86
5.9. Serie de tiempo “Ikeda”.	87
5.10. Serie de tiempo “Henon”.	88
5.11. Serie de tiempo “Tent”.	88
5.12. Serie de tiempo “A1” (Concurso Santa fe).	89
5.13. Serie de tiempo “D1” (Concurso Santa Fe).	89
5.14. Serie de tiempo “Laser”.	90
5.15. Serie de tiempo del “Niño”.	90
5.16. Serie de tiempo “Lovaina”.	91
5.17. Serie de tiempo “S&P 500”.	91
5.18. Serie de tiempo “Star”.	92
5.19. Serie de tiempo “Brown Motion”.	92
5.20. Serie de tiempo “White Noise” o también conocida como Ruido Blanco.	93
5.21. Serie de tiempo del Precio Diario del Oro.	93
5.22. Serie de tiempo del Flujo Diario del Rio Jokulsa.	94
5.23. Serie de tiempo del Flujo del Rio Colorado.	94

5.24. Serie de tiempo del nivel del agua del lago Erie.	95
5.25. Serie de tiempo del agua del rio Nilo.	95
5.26. Serie de tiempo de Manchas Solares.	96
5.27. Serie de tiempo de la temperatura máxima en Merlburne.	96
5.28. Serie de tiempo de la temperatura mínima en Merlburne.	97
5.29. Serie de tiempo Southern Oscillation.	97
5.30. Esquema de los pasos a seguir para el aprendizaje de series de tiempo: a) se observa que en el paso 1 se realiza la prueba de estacionaridad, b) paso 2 correspondiente a la normalización de la serie de tiempo en el intervalo $[0, 1]$ ó $[-1, 1]$, c) el paso 3 que corresponde a la división de la imagen en cajas y finalmente d) se observa el proceso de aprendizaje de series de tiempo.	100
5.31. Resultados de la primera etapa para la serie Seno, la cual como se observa presenta un comportamiento dinámico periódico. En a) se observa la gráfica del error de aprendizaje de cada una de las veces que se ejecutó el algoritmo, en b) se observa la distribución de estos errores, en c) se observa el resultado promedio de la ST Seno para todos sus puntos y finalmente en d) se observa un acercamiento de los últimos 200 puntos de la ST original y de la ST generada.	103
5.32. Resultados de la primera etapa para la ST Qperiodic2, la cual presenta un comportamiento cuasi-periódico, en esta figura observamos lo siguiente: a) muestra el resultado de los errores de aprendizaje que se obtienen para cada uno de los 50 experimentos, en b) se observa la distribución de los errores de aprendizaje, en c) se observa el resultado promedio de la serie aprendida y la ST original y en d) se observa un acercamiento de los últimos 1000 puntos de las ST.	104
5.33. Resultados de la primera etapa para la ST Qperiodic3, la cual presenta un comportamiento cuasi-periódico. Observamos en a) la gráfica del error de aprendizaje para cada experimento, en b) se observa la distribución de estos errores de aprendizaje, c) muestra el resultado promedio para todos sus puntos y finalmente d) muestra el acercamiento de los últimos 500 puntos de la ST original (línea continúa azul) y la predicha (línea punteada roja).	105
5.34. Resultados de la primera etapa para la ST “Lorenz”, la cual presenta un comportamiento caótico. Se observa en a) la gráfica de los errores obtenidos durante cada experimento, b) muestra la distribución de estos errores, en c) observamos el aprendizaje de la ST “Lorenz” y finalmente d) muestra un acercamiento de los 500 puntos.	106
5.35. Resultados de la primera etapa de la ST “tent”, la cual tiene un comportamiento caótico. Se ilustra, en a) los errores obtenidos en cada experimento, en b) se observa la distribución de los errores, c) muestra el aprendizaje de toda la ST y finalmente d) muestra el acercamiento de los últimos 100 puntos.	107
5.36. Resultado de la primera etapa de la ST “S&P 500”. a) exhibe los resultados obtenidos por cada experimento, b) muestra la distribución de estos errores, c) ilustra el resultado del aprendizaje para esta ST y d) muestra un acercamiento de los últimos 500 puntos.	108

5.37. Resultado de la primera etapa de la ST “Brown Motion”, la cual presenta un comportamiento estocástico, en a) se observa el resultado del error de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) se observa el aprendizaje para la ST completa y finalmente en d) se observa un acercamiento de los últimos 300 puntos. 109

5.38. Resultado de la segunda etapa para la ST “Seno”. Se muestra en a) se observa el resultado del error de aprendizaje para cada experimento, b) muestra la distribución del error de aprendizaje, c) muestra el resultado de la ST que obtuvo el error promedio y finalmente en d) muestra un acercamiento de los 200 puntos. 111

5.39. Resultados de la segunda etapa para la ST “Qperiodic3”, que tiene un comportamiento cuasi-periódico, en a) se observa los resultados de los errores de aprendizaje obtenidos para cada experimento, en b) observamos la distribución de estos errores, en c) se muestra la ST generada que obtuvo el error promedio y en d) se observan los últimos 400 puntos. 112

5.40. Resultado de la segunda etapa para la ST “Lorenz”, la cual tiene un comportamiento caótico, en a) se observa el resultado de los errores por experimento, en b) observamos la distribución de estos errores, en c) se muestra el resultado de la ST que obtuvo el error promedio y en d) se observan los últimos 1,500 puntos. 113

5.41. Resultado de la segunda etapa para la ST “tent”, la cual tiene un comportamiento caótico, en a) se observa el resultado del error de aprendizaje para cada experimento, en b) se observa la distribución de este error, en c) observamos la ST que obtiene el error promedio y en d) se muestran los últimos 100 puntos. . 114

5.42. Resultado de la segunda etapa para la ST “S&P 500”, la cual presenta un comportamiento complejo, en a) se observa el resultado del error de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) se observa la ST que obtuvo el error promedio y en d) se observan los últimos 500 puntos. 115

5.43. Resultado de la segunda etapa de la ST “Brownian Motion”, el cual presenta un comportamiento estocástico, en a) se observa el error de aprendizaje obtenido para cada experimento, en b) se observa la distribución de este error, en c) observamos la ST que obtuvo el error promedio y en d) observamos los últimos 300 puntos. 116

5.44. Resultado de la tercera etapa para el 10% de la información total, en a) se observan los resultados del error de aprendizaje obtenido de cada experimento, en b) se observan la distribución de estos errores, en c) se muestra la ST que obtiene el error de aprendizaje promedio y d) muestra los últimos 200 puntos. . 117

5.45. Resultado de la tercera etapa para el 5% de la información total, en a) se observan los resultados del error de aprendizaje obtenido de cada experimento, en b) se observan la distribución de estos errores, en c) se muestra la ST que obtiene el error de aprendizaje promedio y d) muestra los últimos 200 puntos. . 118

5.46. Resultado de la ST “Qperiodic3” durante la etapa de la división de la imagen con el 10% del total de la información, en a) se observan los resultados de los errores de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) se observa la ST que generó el error promedio y en d) observamos los últimos 400 puntos. 119

5.47. Resultados obtenidos durante la tercera etapa al dividir la imagen en al 5 % de la información total, en a) observamos el resultado de los errores de aprendizaje durante cada experimento, en b) se muestra la distribución de estos puntos, en c) observamos la ST que obtuvo el error promedio y en d) se muestran los últimos 400 puntos. 120

5.48. Resultado de aplicar la tercera etapa para el 10 % del total de la información a la ST Lorenz, en a) se observa el resultado de los errores de aprendizaje obtenidos durante cada experimento, en b) se observa la distribución de estos errores, en c) se muestra la ST que obtuvo el error promedio y en d) se observan los últimos 1,500 puntos. 121

5.49. Restados obtenidos de la tercera etapa, pero con la división del 5 % del total de la información, en a) se observan los errores de aprendizaje para cada experimento, en b) se observa la distribución de dichos errores, en c) se observa la ST que obtuvo el error promedio y en d) se muestran los últimos 1,500 puntos. 122

5.50. Resultado obtenido de aplicar la tercera etapa con el 10 % del total de la información, en a) se muestran los errores de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) observamos la ST que obtuvo el error promedio y en d) se observan los últimos 100 puntos. 123

5.51. Resultado obtenido de aplicar la tercera etapa con el 5 % del total de la información, en a) se muestran los errores de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) observamos la ST que obtuvo el error promedio y en d) se observan los últimos 100 puntos. 124

5.52. Resultado obtenido de la tercera etapa para el 10 % de la información de la ST S&P 500, en a) se observan los resultados de los errores de aprendizaje obtenidos durante cada experimento, en b) se observa la distribución de estos errores, en c) se muestra la ST que obtuvo el error de aprendizaje promedio y d) muestra los últimos 500 puntos. 125

5.53. Resultado obtenido de la tercera etapa para el 5 % de la información de la ST S&P 500, en a) se observan los resultados de los errores de aprendizaje obtenidos durante cada experimento, en b) se observa la distribución de estos errores, en c) se muestra la ST que obtuvo el error de aprendizaje promedio y d) muestra los últimos 500 puntos. 126

5.54. Resultado de la tercera etapa para el 10 % del total de la ST Brownian Motion, en a) se observan los resultados de los experimentos obtenidos para cada experimento, en b) se observa la distribución de estos, en c) se observa la ST que obtuvo el error de aprendizaje promedio y en d) se muestran los últimos 300 puntos. 127

5.55. Resultado de la tercera etapa para el 5 % del total de la ST Brownian Motion, en a) se observan los resultados de los experimentos obtenidos para cada experimento, en b) se observa la distribución de estos, en c) se observa la ST que obtuvo el error de aprendizaje promedio y en d) se muestran los últimos 300 puntos. 128

5.56. División de la imagen de la ST “Santa fe”, en 10 cajas con la misma anchura (primera etapa). 132

5.57. Resultado de dividir los puntos de la ST “Santa fe” en 10 cajas, las cuales se observan en la primera columna de esta tabla, en la segunda columna se observan los intervalos definidos para cada caja y finalmente en la tercera columna observamos los puntos que pretencen dentro de cada caja. 133

5.58. Esquema de la predicción iterada multipaso que se utilizó en este trabajo, la cual predice de manera iterada multiples puntos (puntos rojos pequeños). . . . 134

5.59. Resultado de aplicar el algoritmo de predicción a la ST “Seno”. En a) se observa el comportamiento dinámico de la ST completa, en b) se observa la distribución de los errores de predicción después de ejecutar 50 veces este algoritmo y en c) se observa la predicción de los últimos 50 puntos de la ST. 136

5.60. Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3” se observa en a) la ST completa y además se observa un rectángulo rojo el cual indica los 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 137

5.61. Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción. 138

5.62. Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 139

5.63. Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 140

5.64. Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 141

5.65. Resultados de la predicción de los últimos 50 puntos de la ST “seno”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 142

5.66. Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 143

5.67. Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.	144
5.68. Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	145
5.69. Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	146
5.70. Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	147
5.71. Resultados de la predicción de los últimos 50 puntos de la ST “seno”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	148
5.72. Resultados de la predicción de los últimos 50 puntos de la ST “seno”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	149
5.73. Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	150
5.74. Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).	151
5.75. Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.	152
5.76. Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.	153

5.77. Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 154

5.78. Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 155

5.79. Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 156

5.80. Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 157

5.81. Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 158

5.82. Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul). 159

5.83. Resultados de la mejor predicción obtenida para la ST “tent”, en a) se observa el resultado que se obtuvo con la técnica de SVM y b) muestra los resultados obtenidos con la técnica de predicción que proponemos en este trabajo. 163

5.84. Resultados de predicción de la ST “seno”, en a) se observan los últimos 50 puntos predichos con la técnica de SVM y en b) se observa los resultados obtenidos de la técnica propuesta. 164

6.1. Metodología empleada para obtener los primeros resultados de la hipótesis de analizar las ST con técnicas del mundo pequeño. 170

6.2. Esquema explicativo de como agregar la información global para la toma de decisiones en la predicción. 171

Índice de cuadros

5.1. Tabla del conjunto experimental de series de tiempo.	98
5.2. Resultados del proceso de aprendizaje para cada tipo de división.	129
5.3. Se observa el número total de puntos analizados durante este proceso de aprendizaje para cada ST. Además, se observa el número de puntos dentro de cada caja según la etapa empleada. Como se comentó, el resultado de la primera división correspondiente a la misma anchura genera cajas con distinto número de puntos, es por ello que se muestra el promedio que se calculó de los puntos dentro de cada caja.	131
5.4. Se observan los errores de predicción (RMSE) para una predicción de los últimos 50 puntos de la ST obtenidos mediante la aplicación de las distintas metodologías para dividir la imagen de la ST . Los mejores resultados se observan resaltados en “negritas”.	160
5.5. Se observan los errores de predicción (RMSE) obtenidos mediante la aplicación de las distintas metodologías para dividir la imagen de la ST aplicados para una predicción de los últimos 30 puntos de las ST. Los mejores resultados se observan resaltados en “negritas”.	161
5.6. Se observan los errores de predicción (RMSE) obtenidos mediante la aplicación de las técnicas máquinas de soporte vectorial y la técnica propuesta para la predicción de los últimos 50 puntos.	165
5.7. Se observan los errores de predicción (RMSE) obtenidos mediante la aplicación de la técnica de redes neuronales con respecto a la técnica propuesta para la predicción de los últimos 30 puntos de las ST.	166

Resumen

Existen algunos fenómenos en la naturaleza que debido a la complejidad de sus dinámicas, no es posible obtener modelos matemáticos que los representen de manera adecuada, es por eso,,,,, que para poder estudiar dichos fenómenos se cuentan con una secuencia de observaciones medidas de manera experimental a través del tiempo, dichas secuencias de datos son conocidas como series de tiempo, como por ejemplo la temperatura en la ciudad de Chetumal, la venta de zapatos de alguna empresa, los datos de la bolsa de valores, solo por mencionar algunos. La importancia de las series de tiempo radica en que a través de su adecuado análisis, es posible obtener información valiosa para poder definir la dinámica de dichos fenómenos. En este trabajo se presenta una nueva técnica para aprender el comportamiento dinámico de series de tiempo a través de la información obtenida de su imagen. Un factor de gran importancia para el correcto desempeño de las técnicas de aprendizaje es la adecuada representación de los datos de entrada de las series de tiempo. Así entonces, en este trabajo, además proponemos una nueva manera de extraer información de dichas series de tiempo mediante la división de la imagen o amplitud de ésta en pequeñas regiones llamadas *cajas*. Cada caja proporciona información sobre el comportamiento de un segmento de la imagen de la serie de tiempo objetivo. Es entonces, a partir de esta nueva manera de representación de la información, que proponemos una nueva técnica de aprendizaje de series de tiempo. Esta nueva técnica, aprende el comportamiento dinámico de una serie de tiempo objetivo, mediante la imitación o seguimiento de ésta a través de mecanismos estadísticos, esto es, generamos una nueva serie a partir de valores aleatorios acotados y generados mediante ciertas reglas, los cuales imitan o siguen el comportamiento de la serie original. Los resultados experimentales realizados en este trabajo indican que con esta nueva técnica es posible aprender el comportamiento de series de tiempo con buena precisión.

La información que se obtiene de esta nueva técnica de aprendizaje puede servir para diversas aplicaciones como por ejemplo, en predicción o caracterización de series de tiempo. En este trabajo utilizamos dicha información para aplicarla al complicado pero interesante problema de predicción, lo cual resulta ser una tarea extremadamente compleja en el área del estudio de series de tiempo, pero que, a su vez, dicha predicción es muy útil en diversas áreas del conocimiento, tales como: economía, control, visión por computadora, medicina, entre otras. Los resultados que se obtuvieron con esta nueva representación de datos y la nueva técnica propuesta en el proceso de predicción de series de tiempo fueron comparados con los resultados obtenidos con técnicas modernas como las redes neuronales y máquinas de soporte vectorial, y los errores de predicción que se obtuvieron con nuestra técnica fueron menores que las antes mencionadas.

Abstract

In nature, there are certain natural phenomena with very complex dynamic where a mathematical model that represents the behaviour of this phenomena in a suitable way is not possible to obtain. Is because of this problem that this sort of phenomena are represented by a sequence of observation measured experimentally through time, that sequence of ordered data is called time series, for instance: the temperature in Chetumal city, the sale of shoes in a store, the data of the stock market, among others. The importance of the time series is that through an adequate analysis, it is possible to recover valuable information in order to define the dynamic of a phenomenon. In this work we present a new approach to learn the dynamical behaviour of time series through its image or amplitude information. In order to obtain a better performance of this learning technique, a new input data representation is proposed in this work which is different from the classical ones. This new representation obtains the information of the times series by dividing the image of time series into small sections called “boxes”. This boxes will provide information of the image axis behavior of the time series in order to learn its behavior trough its own experience. On the basis of the information from this procedure the dynamical behaviour of time series can be tracked. The experimental results indicate that, with this new approach the behaviour of the time series can be learned with high accuracy.

This learning information is applied to the complicate but interesting problem of forecasting. During the learning approach the information about the dynamical of the times series is been recorded and then a probabilistic technique to recover the appropriate information in order to forecast h unknown values of the time series is designed. The time series forecasting could be applied into several fields such as: economy, control theory, computer vision, medicine, among others. The results indicate that it is possible to use this new approach to forecasting time series with high accuracy. The experimental results that we obtain with this new representation are compared against modern techniques such as neuronal networks and support vector machines.

Capítulo 1

Introducción

Los modelos teóricos de aprendizaje natural desarrollados en la psicología del aprendizaje son parecidos a los sistemas de aprendizaje desarrollados en inteligencia artificial. Pero a diferencia de los sistemas artificiales, los modelos de aprendizaje natural tienen como objetivo el lograr simular y predecir el aprendizaje que realizan los seres humanos y los demás animales teniendo en cuenta no sólo sus limitaciones de memoria y de capacidad de procesamiento, sino también el tiempo limitado del que disponen a la hora de responder a las demandas de su ambiente [1].

Los humanos y demás animales aprendemos de manera muy flexible, adaptándonos a cambios que tienen lugar en nuestro medio ambiente. Es precisamente esta capacidad de cambiar y de adaptarnos al ambiente lo que nos permite sobrevivir. Además, debemos hacerlo constantemente y de manera muy rápida. Precisamente la capacidad de adaptación radica ahí, en la ausencia de comportamientos rígidos, en la capacidad de asumir riesgos y de dar respuestas más o menos correctas a partir de informaciones incompletas o ruidosas [2].

Es cierto que aún no es posible construir una máquina que sea capaz de aprender de la misma forma que los humanos y los demás animales, esto posiblemente se deba a que sencillamente no tenemos aún una teoría lo suficientemente buena como para que pueda ser aplicada con éxito a todas las áreas del aprendizaje natural. Sin embargo, sí es posible aplicar algunas teorías psicológicas del aprendizaje natural a un número cada vez mayor de situaciones concretas presentes en el área de la inteligencia artificial. Como por ejemplo, el proceso de aprendizaje más básico es la imitación, o también conocido como aprendizaje por observación, es decir, es el proceso de observar la repetición de un comportamiento personal, como por ejemplo, una sonrisa, esto es, nosotros no nacemos sabiendo sonreír, si no que a través de observar repetidas veces este proceso lo intentamos imitar y decimos que empezamos a aprender a sonreír. En otras palabras, el aprendizaje observacional sucede cuando el sujeto contempla la conducta de un modelo repetidas veces y la intenta imitar. Este tipo de comportamiento es parecido al aprendizaje supervisado utilizado en el área de Máquinas de Aprendizaje (“Machine Learning”), la cual es un subconjunto de la inteligencia artificial.

Este tipo de aprendizaje, es el que se emplea en esta tesis para aprender el comportamiento dinámico de series de tiempo estacionarias. Las series de tiempo (ST) son fenómenos en la naturaleza que no son analizados a través de modelos matemáticos, si no que, por el contrario, éstos dependen de observaciones o de experimentos, los cuales describen el comportamiento de dichos fenómenos, como por ejemplo, mediciones del clima, monitoreo de los datos de la bolsa de valores, el número de manchas solares, la venta de zapatos de alguna tienda, entre otros

fenómenos. Estas ST contienen información sobre las variables independientes de un sistema, las cuales determinan su dinámica¹. Existen al menos dos principales objetivos del análisis de series de tiempo:

- identificar la naturaleza de algún fenómeno representado por una secuencia de observaciones y
- la predicción de valores futuros de dicha secuencia de observaciones.

En este trabajo presentamos una nueva técnica para aprender el comportamiento dinámico de series de tiempo estacionarias a través de la información obtenida de su imagen y el principio sobre el cual se basa esta nueva técnica es el conocido como submetas. Este principio establece que es más fácil dividir un problema difícil en varios problemas fáciles y resolverlos de manera independiente en donde la solución del problema completo puede ser la suma de las soluciones independientes o la suma de un subconjuntos de éstas [5]. La clave de esta técnica es dividir un problema secuencial y en donde estas divisiones secuenciales se pueden resolver de manera independiente, obteniendo así una solución global del problema. Usando esta idea dividimos la amplitud de la serie de tiempo en pequeñas regiones las cuales llamamos *cajas*, y éstas proveen información sobre el comportamiento de una sección de la amplitud de la serie de tiempo y con esto generamos una serie de tiempo que aprenda y por lo tanto, siga el comportamiento de alguna serie de tiempo objetivo, demostrando así que esta nueva serie aprendió dicho comportamiento.

Mediante la adecuada recuperación del conocimiento obtenido a través de este proceso de aprendizaje, es posible aproximarnos a una solución al problema de predicción de ST, lo cual resulta ser una tarea extremadamente compleja en el área del estudio de series de tiempo, pero que, a su vez, dicha predicción es muy útil en diversas áreas del conocimiento, tales como: en economía, donde se utiliza para predecir los puntos de la bolsa de valores, o también para predecir la venta de zapatos de alguna tienda; en control, se utiliza para generar la información del estado del sistema a controlar necesaria para la retroalimentación; en visión, se utiliza para retroalimentar la información visual retardada; entre otras. Con lo anterior podemos definir el problema a resolver en esta investigación.

1.1. Planteamiento del problema

Uno de los problemas dentro del área de “Machine Learning” es construir programas computacionales que automáticamente se mejoren con la experiencia [20]. En estos últimos años se han realizado varias aplicaciones exitosas en el área de “Machine Learning”, como por ejemplo, en la minería de datos se han realizado programas que aprenden a detectar transacciones fraudulentas con tarjetas de crédito, otro ejemplo son los programas para vehículos autónomos los cuales aprenden a manejar en autopistas públicas, por mencionar algunos [20].

Por otro lado, dentro del área de análisis de ST uno de los principales problemas es la predicción de ST, el cual pudiera ser resuelto a través de técnicas de aprendizaje. Como podemos observar es posible tratar de resolver el problema del análisis de ST a través de la aplicación de técnicas de “Machine Learning”.

¹La dinámica de un sistema muestra la evolución de sus variables a través del tiempo.

Existen algunas técnicas de “Machine Learning” para aprender el comportamiento dinámico de ST, desde técnicas lineales hasta técnicas no lineales [45]. Sin embargo, este proceso de aprendizaje se basa en obtener la información de los datos de la ST a través de ventanas temporales, esto es, por medio de un corrimiento de ventanas sobre el eje del tiempo (en el caso de la técnica de máquinas de soporte vectorial, ver Figura 1.1b)) o a través de la representación de la información de los datos de la ST en nuevos vectores los cuales contienen los datos correspondientes a ciertos puntos de la ST con un determinado retardo (como es el caso de las redes neuronales, ver Figura 1.1a)) [11]. En otras palabras, se busca la mejor representación de los datos para que las técnicas de aprendizaje funcionen mejor. Sin embargo, estas técnicas cuentan con el problema de que automáticamente se mejoren a través de la experiencia.

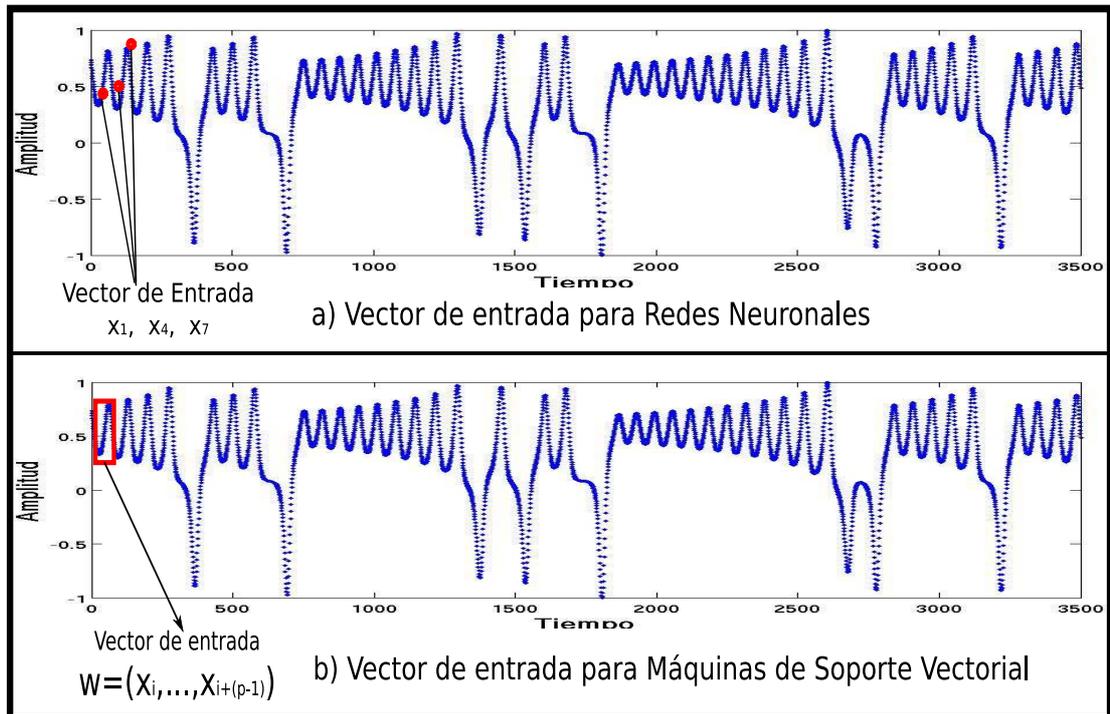


Figura 1.1: Representación de los datos de entrada de dos técnicas de “Machine Learning” para el aprendizaje del comportamiento dinámico de series de tiempo. En a) se observa el vector de entrada para la técnica de redes neuronales y en b) se observa el vector de entrada para la técnica de máquinas de soporte vectorial.

De tal manera que al resolver la problemática del análisis de ST a través de técnicas de “Machine Learning” se generan las siguientes problemáticas:

1. Encontrar una mejor representación de los datos de entrada de la ST para las técnicas de aprendizaje.
2. Generar una técnica adecuada de aprendizaje para esta representación de datos de ST, y que además se mejore automáticamente con la experiencia.

3. Recuperar de manera adecuada el conocimiento obtenido a través del proceso de aprendizaje, con la finalidad de tener una aproximación al problema de predicción. Esta problemática puede ser vista como una aplicación del proceso de aprendizaje (problemática 2).

Si se resuelven de manera adecuada estas problemáticas, entonces, se dice que el aprendizaje del comportamiento dinámico de ST se resolverá de manera adecuada.

1.2. Solución propuesta

A continuación presentaremos la manera en la que abordaremos las problemáticas anteriores. Se puede observar que dentro de éstas se tratará de dar una aproximación al problema de predicción, el cual es un problema complicado del análisis de ST. La solución que ofrecemos toma como base algunas técnicas estadísticas de “Machine Learning”. La metodología que usaremos para resolver las problemáticas que se atacarán en este trabajo serán las siguientes:

- Primero definimos una nueva manera de representar los datos de entrada de la técnica de aprendizaje de ST a analizar, diferente a las que se encontraron en la literatura (ver Figura 1.1). En esta nueva manera de representar los datos de entrada, proponemos lo siguiente: dividir la imagen o amplitud de la ST en n ventanas, llamadas “*cajas*” donde cada una de éstas contenga una cierta anchura y un número adecuado de puntos dentro de cada *caja*. Una vez definidas las regiones (cajas), analizaremos de manera independiente la evolución en el tiempo de los puntos que pertenecen a cada *caja* (en cada visita), como se ejemplifica en la Figura 1.2.

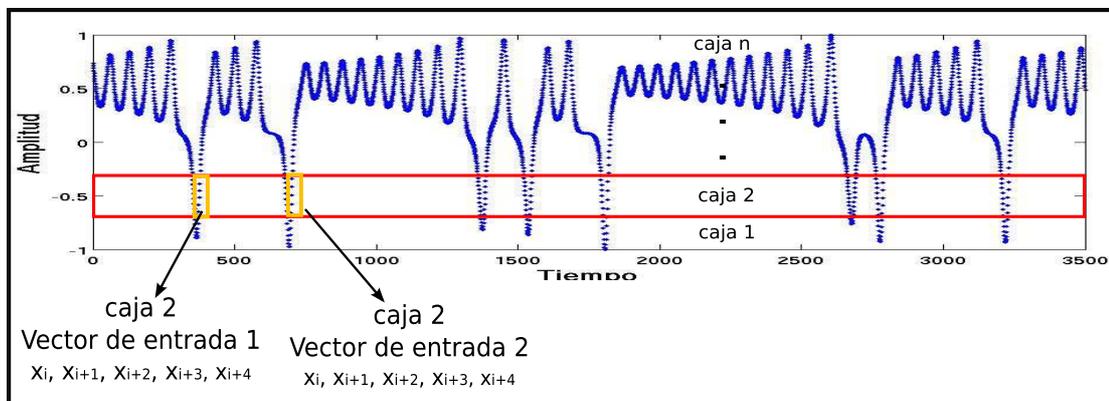


Figura 1.2: Se observa la nueva representación de los datos de entrada de la técnica de aprendizaje propuesta, ésta se obtiene a través del análisis de la imagen de la serie de tiempo. Observando la figura podemos notar que el recuadro amarillo representa el vector de entrada, el cual, corresponde a los puntos que pertenecen dentro de la caja 2 en el primer instante (visita 1), posteriormente, observamos un segundo recuadro amarillo, el cual, indica que estamos analizando los puntos que visitan la caja 2 por segunda ocasión (visita 2) y así sucesivamente. La información contenida en cada caja va cambiando según evoluciona en el tiempo esta serie.

- Posteriormente, generamos una nueva técnica de aprendizaje del comportamiento dinámico de ST, la cual llamaremos “Técnica de Aprendizaje Basado en Imagen” (TABI). Esta nueva técnica se basa en el principio conocido como *submetas* (subgoals) [5], recordemos que la clave de esta técnica es dividir un problema que se resuelve de manera secuencial, mientras que los subproblemas que lo conforman se resuelven de manera independiente aparentemente. En otras palabras, por una parte todos los subproblemas tienen un objetivo en particular, y en este caso, sería aprender el comportamiento de los puntos que pertenecen a una determinada caja, pero por otro lado, todos los subproblemas siguen un objetivo global, y en este caso se refiere al aprendizaje del comportamiento dinámico de series de tiempo.
- Finalmente, emplearemos un mecanismo de indexación para recuperar la información obtenida durante el proceso de aprendizaje, también emplearemos técnicas probabilísticas con la finalidad de tomar decisiones sobre los futuros pasos de los puntos de la ST de acuerdo a lo aprendido, y de esta manera ofrecer una primera aproximación hacia el problema de predicción y a esta nueva técnica la llamaremos “Predicción Basada en Imagen” (PBI).

1.3. Objetivos

La solución de los problemas mencionados anteriormente son la principal motivación de este trabajo de maestría, el cual pretende contribuir al estudio del análisis de series de tiempo a través del empleo de técnicas nuevas de aprendizaje para series de tiempo. A continuación se presentan los objetivos generales y específicos de este trabajo de investigación.

1.3.1. Objetivos generales

1. **Proponer una nueva representación de los datos de entrada de la información de las series de tiempo.**
2. **Generar una técnica estadística que aprenda el comportamiento dinámico de series de tiempo estacionarias y que además se mejore con la experiencia.**
3. **Generar una técnica de indexación que recupere el conocimiento obtenido durante el proceso de aprendizaje.**
4. **Desarrollar una técnica probabilística para la toma de decisiones para la predicción de puntos.**
5. **Comparar los resultados obtenidos entre la técnica propuesta y las técnicas de aprendizaje encontradas en la literatura, en este caso en particular, con las redes neuronales y las máquinas de soporte vectorial.**

1.3.2. Objetivos específicos

- Estudiar y entender las técnicas actuales de aprendizaje de series de tiempo, como por ejemplo, las redes neuronales y las técnicas de máquinas de soporte vectorial y además las diferentes manera de representación de los datos de entrada de la información a aprender.

- Al estudiar las técnicas actuales de aprendizaje y predicción de ST, fue posible observar que el conjunto de ST empleadas en dichas técnicas primero eran normalizadas dentro del intervalo $[-1, 1]$, por lo que fue necesario agregar este paso a la técnica que se propone para el aprendizaje e implementar un algoritmo para dicha normalización.
- Encontrar el número adecuado de cajas, en las cuales se almacena la información apropiada obtenida de la amplitud de la ST, de tal manera que todas las cajas manejen la misma cantidad de información.
- Generar a partir de números aleatorios una secuencia que siga el comportamiento dinámico parecido a la ST original ó ST objetivo y calcular el error de aprendizaje entre la serie original y la serie calculada, así como también calcular el error de predicción entre los puntos de la serie de tiempo original y los puntos predichos.
- Estudiar e implementar soluciones para otras problemáticas que se pueden resolver a través del aprendizaje de la ST como por ejemplo, la predicción y además comparar los resultados obtenidos entre la técnica propuesta de predicción contra otras técnicas encontradas en la literatura.

1.4. Contribución

Dentro de los resultados que se obtuvieron del desarrollo de este trabajo de tesis se tienen las siguientes contribuciones:

- Se desarrolló una nueva representación de los datos de entrada de las técnicas de aprendizaje, a través de la información del eje imagen de las series de tiempo.
- Se generó una técnica nueva, la cual utiliza como base el principio de submetas que dice, que es más fácil resolver un problema difícil si éste se divide en varios subproblemas sencillos que estén ligados de manera secuencial y esta técnica encuentra diferente información que las técnicas clásicas.
- Se desarrolló un nuevo algoritmo para la predicción de series de tiempo a partir de la información obtenida de la técnica de aprendizaje y se comprobó que a través de la combinación de una nueva representación de la información y una nueva técnica de aprendizaje se obtuvieron mejores resultados en el problema de predicción de series de tiempo comparado con las técnicas de redes neuronales y máquinas de soporte vectorial.

1.4.1. Trabajos publicados

Durante este trabajo de tesis se obtuvieron las siguientes publicaciones:

1. Revista Indexada por Latindex

- “*On a New Approach to Time Series Tracking*”, Ramírez-Amaro, K, Chimal-Eguía, J.C. Advances in Computer Science and Engineering. eds. Torres S., López I. and Calvo H. Research in Computer Science 27, pp. 205-215, México, ISSN: 1870-4069, 2007.

- “*Automatic Generation of Hypotheses Using the GUHA Method*”, K. Ramírez-Amaro, V. Ortega-González and J. Figueroa-Nazuno. Data Mining and Information. Guzman-Arenas A. and Olivares. Research in Computer Science , pp. 65-72, México, D. F., Noviembre. ISSN: 1870-4069, 2006.

2. Congresos internacionales

- “*New Learning Technique Applied to Time Series Forecastig* ”, K. Ramirez-Amaro, J.C. Chimal-Eguía and N. Sánchez-Salas. 16th International Conference on Computing (CIC’07), Centro de Investigación en Computación del IPN, México, D. F., Noviembre 2007.
- “*Una Nueva Técnica para Aprender el Comportamiento de Series de Tiempo*”, Ramírez-Amaro K., Chimal-Eguía J.C. 6ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI’07), Orlando, Florida, EE.UU (2007).
- “*New Approach to Time Series Tracking*”, Ramírez-Amaro K. and Chimal-Eguía J.C., 27th International Symposium on Forecasting, New York City, USA 24-27, June 2007.
- “*Recurrence Plot Analysis and its Application to Teleconnection Patterns*”, K. Ramirez-Amaro, J. Figueroa-Nazuno. 15th International Conference on Computing (CIC’06), Centro de Investigación en Computación del IPN, pp. 65-72, México, D. F., Noviembre 2006. ISBN: 0-7695-2708-6.

3. Congresos nacionales

- “*Evaluación de modelos de predicción de Series de Tiempo*”, E. Castillo-Montiel, M. A. Paredes-Farrera, C. Bustillo-Hernández, K. Ramírez-Amaro & J. Figueroa-Nazuno. ROC&C 2006, Acapulco Guerrero, Diciembre 2006.
- “*Técnicas de Dinámica No Lineal para el Estudio de la Respuesta del Sitio en Sismo*”, K. Ramírez-Amaro, O. Domínguez- Acevedo, S. R. García-Benitez, J. Figueroa-Nazuno, F. Correa-Giraldo, A. Angeles-Yreta, M. P. Romo-Organista, M. Ortega-Sanchez, M. Rupit-Olvera. ROC&C 2005, Acapulco Guerrero, Noviembre 2005.
- “*Extracción Automática de Modelos, Empleando Técnicas de Mapa Recurrente*”, K. Ramírez-Amaro, J. Figueroa-Nazuno, A. Angeles-Yreta, V. Landassuri-Moreno, ROC&C 2005, Acapulco Guerrero, Noviembre 2005.
- “*Búsqueda de Semejanza entre Objetos 3D por Indexado*”, A. Angeles-Yreta, J. Figueroa-Nazuno & K. Ramírez-Amaro. ROC&C 2005, Acapulco, Guerrero. Noviembre 2005.
- “*Generación Automática De Modelos Para Series De Tiempo*”, K. Ramírez-Amaro, O. Domínguez-Acevedo y J. Figueroa-Nazuno. Congreso Nacional de Física 2005, 3MC4, pág. 97, Guadalajara, Jalisco. Octubre 2005.
- “*Dinámica no lineal para el modelado de la respuesta de sitio*”, K. Ramírez-Amaro, F. Correa, S. García, y J. Figueroa-Nazuno, Marco, Oscar, H. Solís-Estrella y A. Angeles-Yreta. XV congreso nacional de ingeniería sísmica, Septiembre 2005.

- “*Análisis de los Índices de Teleconexión Atmosféricos Empleando Técnicas de Mapas Recurrentes Y Modelos Casuales*”, K. Ramírez-Amaro, H. Jiménez-Hernández, J. Figueroa-Nazuno; IEEE Reunión de Otoño de Comunicaciones y Computación, Noviembre 2004.
- “*Predicción de Series de Tiempo por el Método de Regresiones Logísticas*”, O. Delfín, K. Ramírez-Amaro, J. Figueroa-Nazuno, XLVII Congreso Nacional de Física, Memorias en CD, Octubre 2004, ISSN: 01874713.

4. Seminarios de Investigación

- “*On a New Approach to Time Series Tracking*”, Karinne Ramírez-Amaro. Seminario de Termodinámica y Sistemas Complejos, organizado por la Escuela Superior de Física y Matemáticas del Instituto Politécnico Nacional (ESFM-IPN), el día 14 de junio del 2007.

5. Reportes técnicos

- “*Empleo de la técnica Mapa Recurrente en el análisis de los Índices de Teleconexión atmosféricos*”, K. Ramírez-Amaro, J. Figueroa-Nazuno K. Ramírez-Amaro, Reporte Técnico editado por Centro de Investigación en Computación del IPN, 2006. ISBN: 970-36-0330-0.
- “*Sobre el Problema de Semejanza*”, J. Figueroa-Nazuno, A. Angeles-Yreta, J. Medina-Apodaca, V. Ortega-González, K. Ramírez-Amaro, M. Mirón-Bernal, V. Landassuri-Moreno, Reporte Técnico editado por el Centro de Investigación en Computación del IPN, 2006. ISBN: 970-36-0343-2.

1.5. Organización

Este trabajo está organizado de la siguiente manera:

- En el capítulo 2, se describirán los fundamentos teóricos utilizados a lo largo de este documento, los cuales son: las series de tiempo, los objetivos del análisis de series de tiempo, las técnicas que describen los datos de las ST, los tipos de comportamientos dinámicos que presentan las ST, así como también la definición de ST estacionarias y la descripción de algunas técnicas estadísticas que se emplean para evaluar la estacionaridad de las ST. En este mismo capítulo, se introduce la definición de aprendizaje y cómo es posible emplearlo en ST, también se estudia el aprendizaje que realizan las técnicas de redes neuronales y máquinas de soporte vectorial. Además, se introduce el concepto de predicción y se define el tipo de predicción que se estudia en este trabajo.
- En el capítulo 3, se describe la manera en cómo será la nueva representación de los datos de entrada de la(s) técnica(s) de aprendizaje de ST. También se definirán algunos conceptos relacionados con las ST. Posteriormente, se describen los pasos de la técnica de aprendizaje propuesta para ST. Finalmente, se realiza una comparación de la complejidad de las decisiones o pasos que se llevan a cabo para la “Técnica de Aprendizaje Basado en Imagen” con respecto a las técnicas de redes neuronales y máquinas de soporte vectorial.

- En el capítulo 4, se describe la aplicación de la técnica propuesta de aprendizaje hacia el problema de la predicción y ésta se refiere a la adecuada recuperación del conocimiento acumulado durante el proceso de aprendizaje, para la toma de decisiones. También se describen los pasos a seguir para obtener la predicción de ST a través de un mecanismo probabilístico propuesto.
- En el capítulo 5, se muestra la descripción del conjunto experimental de ST empleado en este trabajo, así como también los resultados de la técnica de aprendizaje de series de tiempo y los resultados de sus respectivas etapas de desarrollo. También se observan los resultados de la técnica de predicción de ST y finalmente, se observa la comparación de los resultados de la “Predicción Basada en Imagen” con respecto a los resultados obtenidos de las redes neuronales y las máquinas de soporte vectorial.
- Finalmente, en el capítulo 6 se ofrecen las conclusiones generales obtenidas como resultado del desarrollo de este trabajo, así como el trabajo futuro a realizar siguiendo esta línea de investigación.

Capítulo 2

Marco teórico

Hace cuatro siglos, Issac Newton observó que las descripciones matemáticas nos ayudan a entender la naturaleza de los fenómenos. Sin embargo, existen fenómenos que no son analizados a través de modelos matemáticos debido a que el cálculo de un modelo matemático que describa por completo la dinámica de un fenómeno natural, la mayoría de las veces resulta ser una tarea muy compleja y en algunos casos imposible. Una forma de resolver este problema es describir la dinámica del fenómeno natural directamente de la observación del mismo. Esto es, medir una característica que defina al fenómeno durante un intervalo con una frecuencia definida. Estas observaciones o experimentos, pueden describir el comportamiento de los fenómenos y con esto definir su dinámica al menos de una forma cualitativa. Los datos de cualquier fenómeno obtenidos a través de estas observaciones o experimentos son llamados: Series de Tiempo (ST). Existen al menos dos objetivos principales del análisis de series de tiempo; (a) identificar la naturaleza de algún fenómeno representado por una secuencia de observaciones y (b) la predicción de valores futuros de dicha secuencia de observaciones.

Como se puede suponer, la predicción de ST es una tarea muy compleja, dicha tarea se puede tratar de resolver a través de técnicas de “Machine Learning”. Tomando esta idea, en este trabajo proponemos una solución para este interesante problema.

A lo largo de este capítulo se tratarán algunas definiciones básicas de ST y se harán referencias a trabajos previos relacionados al análisis de series de tiempo. También se introduce la definición de aprendizaje y además se define el tipo de aprendizaje que se realizará en esta tesis, así como estudiar el aprendizaje que realizan las técnicas de redes neuronales y máquinas de soporte vectorial, para finalmente compararlas con la técnica de aprendizaje que proponemos en este trabajo. Además, se introduce el concepto de predicción y se define el tipo de predicción que se estudia en este trabajo.

2.1. Series de tiempo

Una ST es un conjunto de observaciones medidas generalmente de manera secuencial a través del tiempo [8]. Estas mediciones se pueden obtener de manera continua o de forma discreta a través del tiempo. Si el conjunto de puntos es medido en tiempo continuo entonces se dice que la ST es continua, pero sí el conjunto de puntos es medido u observado en tiempo discreto entonces se dice que la ST es discreta. En otras palabras, una definición más formal de ST discreta sería la siguiente [11], [15] y [6]:

Definición 1 Una ST discreta es una secuencia de valores $x(t)$ de un sistema medido sobre el tiempo y esta secuencia de valores $x(t)$ son obtenidos de manera experimental,

$$x(t_1), x(t_2), x(t_3), \dots, x(t_n), \quad (2.1)$$

para algún intervalo $t = n$, que cumpla con lo siguiente:

$$t_1 < t_2 < t_3 < \dots < t_n. \quad (2.2)$$

En este trabajo consideraremos únicamente ST discretas, donde las observaciones se toman de un cierto intervalo fijo $h = [t_1, t_n]$. Un ejemplo de ST discretas se observa en la Figura 2.1.

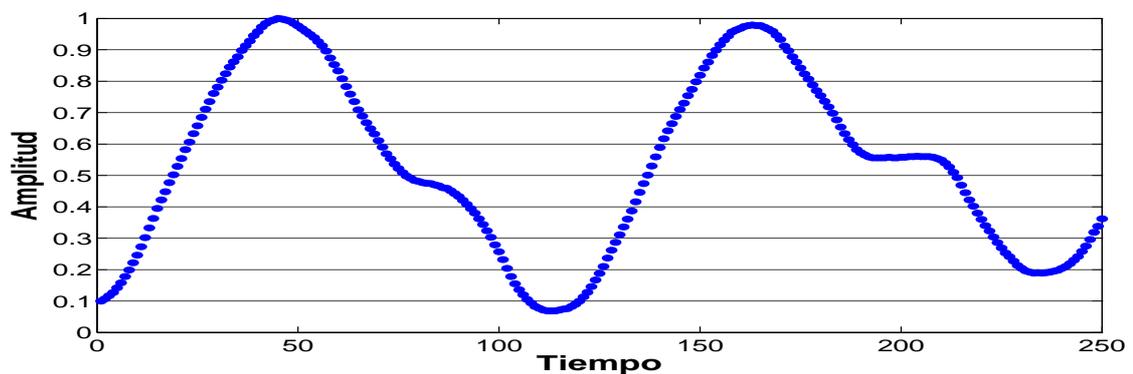


Figura 2.1: Ejemplo de una serie de tiempo discreta, donde se observa cómo la función está descrita por puntos discretos, $x(t_1), x(t_2), \dots, x(t_n)$.

Las ST contienen información sobre las variables independientes de un sistema, las cuales determinan su dinámica. La dinámica de sistemas muestra la evolución de las variables a través del tiempo [24]. La extracción de esta información es un problema que se estudia mediante el análisis (caracterización), predicción y modelado de las series de tiempo.

Las ST discretas se pueden obtener de la siguiente manera:

1. Por el *muestreo* de una ST continua; por ejemplo, la temperatura medida en intervalos de cada hora.
2. Por el *acumulamiento* de una variable sobre un período de tiempo; por ejemplo, la lluvia, la cual se acumula usualmente cada día o cada mes, o las ventas que se pueden acumular cada día, mes o año.
3. O como series *discretas por naturaleza*; por ejemplo, el dividendo del pago de una compañía en años consecutivos.

Para cualquier tipo de ST discreta, los datos son almacenados típicamente en *intervalos de tiempo iguales*, esto con el fin de analizar el comportamiento dinámico de las series de tiempo. Recuerde que la dinámica de un sistema puede ser expresada como ecuaciones diferenciales de segundo orden cuyas ecuaciones de diferencias se definen en períodos constantes.

Entonces, se entiende por ST, al conjunto de datos numéricos que se obtienen en períodos regulares a través del tiempo para un fenómeno. Por otro lado, existe una analogía entre ST y señal ya que podemos decir que: la energía transmitida detectable que puede ser usada para transportar información. Pero cuando trabajamos con señales en las computadoras, es necesario realizar un muestreo y cuantificar con valores, ya que las computadoras no pueden trabajar en el dominio continuo. Es por eso, que podemos ver a los conceptos ST y señal de manera semejante, hablando en términos computacionales, ya que ambos se determinan con datos discretos de algún fenómeno.

2.1.1. Objetivos del análisis de series de tiempo

Una de las características especiales de series de tiempo es que las observaciones secuenciales usualmente *no son independientes* y debido a lo anterior el análisis debe tomar en cuenta el *orden* en el cual las observaciones fueron tomadas.

Los objetivos principales del análisis de series de tiempo son:

Descripción. Describir los datos usando técnicas estadísticas y/o métodos gráficos. Una graficación de los datos es particularmente valioso.

Modelación. Encontrar un modelo que describa el proceso de generación de datos. Un modelo *univariado* para una variable dada está basado sólo en los valores pasados de dicha variable, mientras que un modelo *multivariado* para una variable dada puede ser basado no solo en valores pasados de dicha variable sino también en valores presentes y pasados de otras variables.

Predicción. Estimar los valores futuros de una ST. Existe una distinción evidente entre lo que es la predicción de un estado estable “steady-state” donde se espera que el futuro sea muy parecido al pasado y una predicción hipotética “what-if” donde un modelo multivariado es utilizado para explorar el efecto al cambiar la política de las variables [23].

En este trabajo se estudia la predicción de ST tomando como base de predicción la nueva técnica de aprendizaje propuesta en este documento.

2.1.2. Técnicas para la descripción de las series de tiempo

Antes de realizar cualquier tipo de análisis sobre los datos, como por ejemplo, un modelado ó predicción de éstos, primero es necesario tener un conocimiento previo de los datos y su evolución para tener un “presentimiento” sobre éstos, con lo cual se puede tener una idea general de sus principales características. Graficar los datos en el tiempo es una de las herramientas más importantes, pero las técnicas estadísticas también ayudan a tener un conocimiento previo sobre los datos a analizar.

A continuación, se presentará un breve resumen de las variaciones de ST de acuerdo a los siguientes componentes para permitir un mejor análisis [23]:

Variación temporal. Este tipo de variación se presenta generalmente en períodos anuales y aparecen durante varios años, sin importar si fue medido semanal, mensual o trimestralmente. En este tipo de variación se observan patrones similares de comportamiento en

períodos particulares de tiempo durante varios años. Un ejemplo de este tipo de variación, serían los patrones encontrados en las ventas de helado, los cuales siempre son altos durante la temporada de verano. Es importante mencionar que, sí una ST es medida únicamente cada año (esto es, una vez al año), entonces, no es posible afirmar que existe una variación temporal en esa ST.

Tendencia. Este tipo de variación está presente cuando una serie exhibe una estabilidad creciente o decreciente al menos durante varios períodos de tiempo sucesivos. Un ejemplo de este tipo de variación, es la serie del índice industrial Down Jones de Nueva York mostrada en la Figura 2.3b) y como se observa hay un incremento de este índice a través de los años. La tendencia puede ser definida de manera general como: “el cambio del nivel de la media a largo plazo”, pero esa no es una definición matemática satisfactoria. La definición de tendencia depende en parte de la longitud de los datos muestreados y debido a eso, la descomposición de la variación de los datos de la ST no es única.

Variación cíclica. Ésta incluye una variación regular cíclica en períodos diferentes a un año. Un ejemplo incluye los ciclos de los negocios sobre un período de tal vez cinco años y el ritmo diario en los comportamientos biológicos de los seres vivos (llamado variaciones diurnas).

Fluctuaciones irregulares. Este tipo es utilizado a menudo para describir cualquier variación que no esté en la variación temporal, tendencia y variaciones cíclicas. Como por ejemplo, los datos podrían ser completamente aleatorios. Sin embargo, estos datos podrían describir correlaciones de corto alcance o incluir una discontinuidad específica.

Los métodos clásicos [23] trabajan muy bien cuando se trata de series en las que domina una variación lineal de tendencia y/o datos con variaciones temporales. Pero no funcionan muy bien cuando los datos presentan tendencia y/o variaciones temporales irregulares, esto es, están cambiando a través del tiempo o cuando valores consecutivos de fluctuaciones irregulares están correlacionados. Cuando valores consecutivos de una ST están correlacionados se está hablando generalmente de una autocorrelación entre los datos [22]. Este tipo de ST se analizan con técnicas no lineales como por ejemplo: con técnicas de mapas recurrentes, técnicas de espacio de fase, exponentes de Lyapunov, mapas de Poincaré, entre otros [8].

2.1.3. Comportamientos dinámicos de las series de tiempo

Existen diferentes características que tienen los datos de las series de tiempo y estas características definen el comportamiento dinámico de las mismas, las cuales pueden ser clasificadas de la siguiente manera:

Periódicas. Una señal periódica es una función en la cual su forma se repite exactamente después de un período o ciclo [6]. Por ejemplo, en la vida diaria existen muchos casos de funciones periódicas cuando la variable es el tiempo; situaciones como el movimiento de las manecillas de un reloj o las fases de la luna, muestran un comportamiento periódico. Un movimiento periódico es un movimiento en el que la(s) posición(es) del sistema se puede(n) expresar en base a funciones periódicas, todas bajo el mismo período, ejemplo $f(x) = \text{seno}(x)$. Cuando una señal está en el dominio de los números reales o de los enteros, significa que la totalidad de su gráfica puede ser representada a partir de copias de una determinada porción de ésta, repetida a intervalos regulares [6].

Cuasi-periódicas. Una función es cuasiperiódica si puede expresarse como una suma de funciones periódicas con períodos, donde al menos algunos de los períodos son inconmensurables (esto es, su proporción es irracional). Un ejemplo de patrón cuasiperiódico son los llamados mosaicos de Penrose [6].

Complejas. Este comportamiento no tiene una definición consistente en la literatura [6]. Algunos autores utilizan este término para definir el comportamiento de alguna señal como no periódica o no constante pero sin llegar a ser aleatorio. En un sistema complejo existen variables ocultas cuyo desconocimiento nos impide analizar el sistema con precisión. Así pues, el análisis total de un sistema complejo posee más información que cada una de sus partes analizadas de manera independiente. Para describir un sistema complejo hace falta no sólo conocer el funcionamiento de las partes sino conocer cómo se relacionan entre sí. Un ejemplo típico de un sistema complejo es la Tierra. La Tierra está formada por varios sistemas que la describen: campo gravitatorio, campo magnético, flujo térmico, ondas elásticas, geodinámica, entre otros. Cada uno de estos sistemas está bien estudiado pero desconocemos la forma en que interactúan y hacen evolucionar el sistema “Tierra”. Entonces, existe mucha más información oculta en esas interrelaciones de sistemas, que en cada uno por separado.

Caóticas. Los sistemas dinámicos se pueden clasificar básicamente en: estables, inestables y caóticos (caos determinista) [46]. Un sistema estable tiende a un punto u órbita a medida que el tiempo transcurre, según su dimensión (atractor). Un sistema inestable se escapa de los atractores, y un sistema caótico manifiesta los dos comportamientos [46]. Por un lado, existe un atractor por el cual el sistema se ve “atraído”, pero a la vez, hay “fuerzas” que lo alejan de éste. De esa manera, el sistema permanece confinado en una zona de su espacio de estados, pero sin tender a un atractor fijo. En los sistemas caóticos, una mínima diferencia en esas condiciones hace que el sistema evolucione de manera totalmente distinta¹ [7]. Ejemplos de tales sistemas incluyen, la atmósfera terrestre, el Sistema Solar, las placas tectónicas, los fluidos en régimen turbulento y los crecimientos de población.

Estocásticas. Si valores futuros de una secuencia de números puede ser predicha exactamente con una función de sus valores pasados, entonces, se dice que la serie es *determinista*. Sin embargo, muchas series son estocásticas o aleatorias en el cual el futuro sólo es parcialmente determinado por valores pasados. Las leyes conocidas de causa-efecto no explican cómo actúa el sistema (y de modo reducido el fenómeno) de manera determinista, sino en función de probabilidades. De tal modo que, en matemáticas, la estocástica resulta ser un conjunto de teorías estadísticas que tratan a los procesos cuya evolución es aleatoria, un ejemplo de ellos son las tiradas de dados [7]. Sí existe un modelo apropiado para este comportamiento aleatorio entonces, el modelo debe determinar una muy buena predicción de valores futuros. Un modelo estocástico de ST es a menudo llamado proceso estocástico [6].

¹Este principio suele llamarse efecto mariposa debido a que, en meteorología, la naturaleza no lineal de la atmósfera ha hecho afirmar a muchos científicos que es posible que el aleteo de una mariposa en un determinado lugar y momento, pueda ser la causa de un huracán varios meses más tarde en otra parte del globo.

2.1.4. Series de tiempo estacionarias

Una señal estacionaria está basada en la suposición de que la señal está en un estado de *equilibrio estadístico* [22]. En el caso de la medición de las ST existen dos nociones diferentes de estacionaridad [6]. La primera se refiere a la forma más débil de estacionaridad, la cual requiere de tener el modelo que produce la señal y que además todos los parámetros del sistema estudiado sean estables y constantes durante su período de medición. Si el proceso en observación es probabilístico, éste será caracterizado por distribuciones probabilísticas de las variables envueltas. Para procesos estacionarios estas probabilidades pudieran no depender del tiempo. Si hay reglas determinísticas que gobiernan la dinámica, estas reglas no deberán cambiar durante el muestreo de la ST.

Desafortunadamente, en la mayoría de los casos, no tenemos acceso directo al modelo del sistema que produce la señal y no es posible establecer si sus parámetros permanecen constantes. Esto nos lleva a formular un segundo concepto de estacionaridad basado únicamente en los datos disponibles. Una señal es llamada estacionaria si todas las probabilidades de transición² son independientes del tiempo dentro de su período de observación, esto es, cuando se calcula de los datos observados. Esto incluye la constancia de ciertos parámetros, pero además requiere que las ST de los fenómenos estén suficientemente muestreadas, para que sus probabilidades u otras reglas puedan ser inferidas de manera más acertada. Esta definición es diferente con respecto a la primera, debido a que hay varios procesos que son formalmente estacionarios cuando el límite de observaciones es infinitamente largo, pero cuando éstos son estudiados en tiempos finitos se comportan como procesos no estacionarios.

Para nuestros propósitos, si la señal observada es por lo general regular la mayor parte del tiempo, pero contiene una ruptura muy irregular de vez en cuando, entonces la ST tiene que ser considerada como no estacionaria para nuestros propósitos, aún cuando todos los parámetros permanezcan constantes y se dice que dicha señal es intermitente [6]. Únicamente si los eventos raros (rupturas irregulares) aparecieran varias veces en la ST, podemos hablar de una independencia de las probabilidades de transición observadas y por lo tanto de estacionaridad.

En otras palabras, una señal se dice que es estrictamente *estacionaria* si sus propiedades no se ven afectadas por algún cambio en su tiempo de origen [22]; esto es, si la distribución de probabilidad de alguna señal asociada con m observaciones $x(t_1), x(t_2), x(t_3), \dots, x(t_m)$, generadas en cualquier conjunto de tiempo $t_1, t_2, t_3, \dots, t_m$, es la misma que la asociada con m observaciones $x(t_{1+k}), x(t_{2+k}), x(t_{3+k}), \dots, x(t_{m+k})$, generada en los tiempos $t_{1+k}, t_{2+k}, t_{3+k}, \dots, t_{m+k}$, entonces la señal es estacionaria. Por lo tanto, para que una señal discreta sea estrictamente estacionaria, la distribución de probabilidad de cualquier conjunto de observaciones no debe de ser afectada por cualquier desplazamiento hacia adelante o hacia atrás del eje del tiempo por alguna constante k .

Cabe mencionar que en este trabajo no se va a trabajar con el concepto de estacionaridad débil, debido a que no se conoce el modelo matemático de las ST que se usarán como base de prueba.

Prueba de estacionaridad

Después de hacer mucho énfasis en el problema de la estacionaridad, es necesario cuestionar cuándo, para un conjunto de datos dado, la no estacionaridad en una ST puede ser detectada,

²Transición se refiere al cambio de estado de un sistema a otro.

como se mencionó anteriormente la estacionaridad es una propiedad que nunca puede ser establecida con precisión. Existen técnicas estadísticas como el cálculo de la media, la varianza, la autocovarianza, entre otras, para probar la estacionaridad de un proceso. En este trabajo nos enfocaremos a las técnicas de media y varianza.

Media y Varianza de una señal. La media de una señal μ puede ser definida como el promedio de un conjunto de números y se puede estimar de la siguiente manera [6]:

$$\mu_{x(t)} = \frac{1}{N} \sum_{i=1}^N x(t_i) \quad (2.3)$$

donde $x(t_i)$ representan los puntos de la ST y N corresponde al número total de puntos de la ST. La varianza de una ST $\sigma_{x(t)}^2$, es un estimador de la dispersión de una variable aleatoria x respecto a su esperanza $\mu_{x(t)}$ y se estima de la siguiente manera [6]:

$$\sigma_{x(t)}^2 = \frac{1}{N} \sum_{i=1}^N (x(t_i) - \mu_{x(t)})^2 \quad (2.4)$$

Para determinar si una ST es estacionaria se evalúan los siguientes pasos:

1. Se define una ventana de m puntos de la ST donde $m \in N$ tal que, $0 < m < N$.
2. Se define una técnica para definir la estacionaridad de ST, esto es, la media, varianza, autocovarianza u otra.
3. Cuando se han cumplido con los pasos anteriores, entonces se procede a aplicar la técnica selecta a cada una de las ventadas definidas para toda la ST y esta ventana se desplaza hacia adelante o hacia atrás a través del eje del tiempo.
4. Finalmente, se grafican los resultados de la técnica seleccionada aplicada a cada ventana. Sí estos valores muestran una distribución uniforme, entonces significa que la ST es estacionaria, pero por el contrario, sí los resultados no muestran una distribución uniforme, entonces la ST es no estacionaria.

Por ejemplo, observe la Figura 2.2 en donde se ejemplifica la prueba de estacionaridad para la ST Ruido Blanco “White Noise”, en donde a través del cálculo de la media y de la varianza se demuestra que dicha señal es estacionaria.

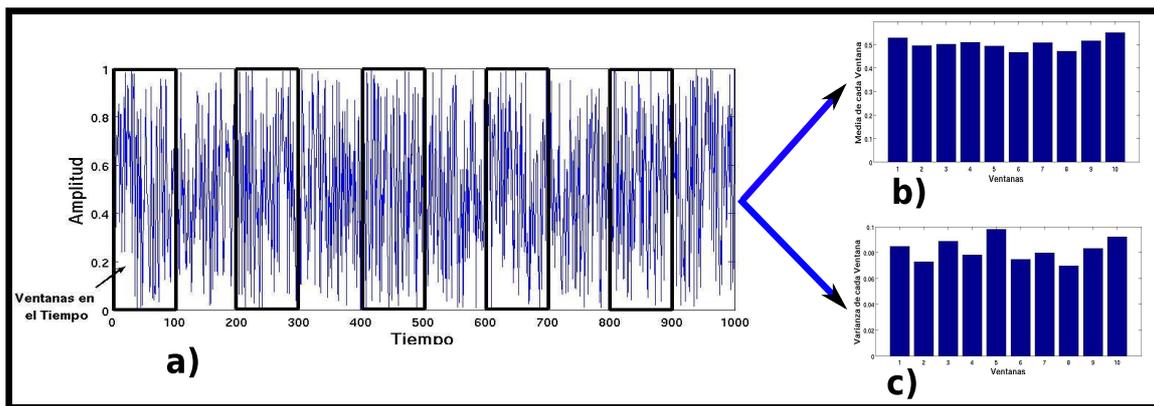


Figura 2.2: Ejemplo de la prueba de estacionaridad. En a) se observa la ST Ruido Blanco y como se subdivide en ventanas de 100 elementos sobre el eje del tiempo, en b) se observa la distribución de la media de cada ventana y en c) se observa la varianza de cada ventana, se puede observar que siguen una distribución uniforme en ambos casos.

Por otro lado, un ejemplo de señal no estacionaria se muestra en la Figura 2.3 y como se observa a través del cálculo de la media y la varianza se demuestra que la señal es no estacionaria. Señales no estacionarias son muy comunes, por ejemplo en particular cuando observamos fenómenos naturales y culturales [6].

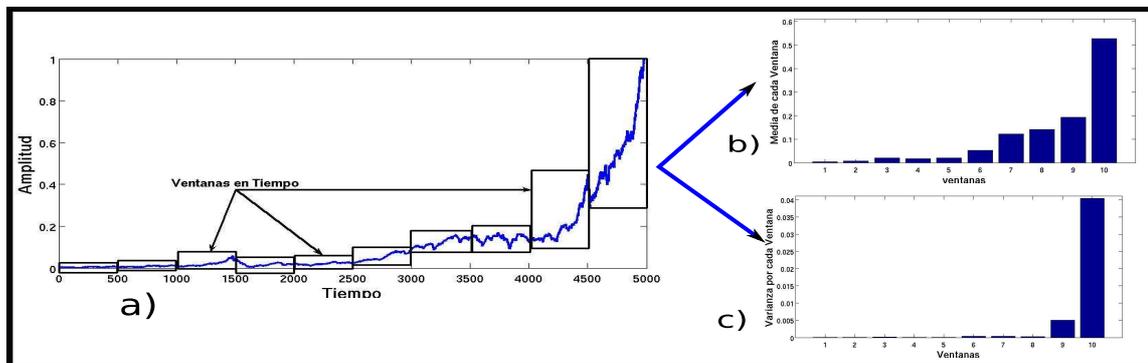


Figura 2.3: Ejemplo cuando no cumple con la prueba de estacionaridad. En a) se muestra la ST “Down Jones” dividida en 10 ventanas de 500 puntos cada una, en b) se observa la media de los puntos de cada ventana y en c) se observa la varianza de los puntos de cada ventana. Como se observa en b) y c) todas las ventanas tienen distinta distribución de puntos al variar el tiempo, por lo tanto la ST es no estacionaria.

2.2. Aprendizaje de series de tiempo

Posiblemente las razones por las cuales no ha sido posible construir una máquina que sea capaz de aprender igual que como lo hacemos los humanos y los demás animales, es posiblemente porque los modelos de aprendizaje natural tienen como objetivo el lograr simular y predecir el aprendizaje que realizan los seres humanos y los demás animales teniendo en cuenta no sólo sus limitaciones de memoria y de capacidad de procesamiento, sino también el tiempo limitado del que disponen a la hora de responder a las demandas de su ambiente y además posiblemente no tenemos aún una teoría lo suficientemente buena como para que pueda ser aplicada con éxito a todas las áreas del aprendizaje natural[1].

Sin embargo, sí es posible aplicar las siguientes teorías psicológicas del aprendizaje natural a un número cada vez mayor de situaciones concretas desarrollados en inteligencia artificial:

Aprendizaje por reforzamiento. Define la manera de comportarse de un agente a un tiempo dado en un tiempo exacto. Puede verse como el mapeo entre los estados del ambiente que el agente percibe y las acciones que toma, cuando se encuentra en esos estados. Corresponde a lo que en psicología se conoce como reglas de estímulo-respuesta o asociaciones.

Aprendizaje por observación. El proceso de aprendizaje más básico es la imitación, es decir, el proceso de observar la repetición de un comportamiento personal, como por ejemplo una sonrisa. Por lo tanto, este proceso pudiera tomar un cierto tiempo (prestar atención en los detalles), espacio (un lugar para aprender), habilidades (o práctica), entre otros. En otras palabras, el aprendizaje observacional sucede cuando el sujeto contempla la conducta de un modelo, aunque se pueda aprender una conducta sin llevarla a cabo. Este proceso consiste de los siguientes pasos:

- adquisición, se refiere a observar un modelo y reconocer sus rasgos característicos de conducta;
- retención, es cuando se almacenan las conductas del modelo en la memoria del observador;
- ejecución, es cuando se decide si se considera la conducta apropiada y si sus consecuencias son positivas entonces, se reproduce la conducta;
- finalmente las consecuencias, se generan cuando al imitar al modelo, el individuo puede ser reforzado por la aprobación de otras personas, este paso implica atención y memoria, pues es de tipo de actividad cognitiva.

Un ejemplo de este tipo de aprendizaje, es cuando los niños aprenden a escribir y este aprendizaje se mejora con la experiencia.

Juegos. Los juegos generalmente describen comportamientos que no tienen una finalidad en particular por sí mismos, sin embargo, mejoran su desempeño en situaciones similares en un futuro. Esto se observa en una gran variedad de vertebrados, además de los humanos, pero es limitado a mamíferos y pájaros. Por ejemplo, los gatos cuando son pequeños en un principio aprenden a jugar con alguna pelota de hilo y ese aprendizaje sin ningún fin más que diversión se convierte en un aprendizaje útil cuando estos animales tienen que cazar a su presa.

Aunque el aprendizaje a menudo se piensa asociado con humanos ó más general con organismos vivientes, las computadoras también son capaces de modificar sus propios comportamientos como resultado de la experiencia. Esto se conoce como “Machine Learning”, el cual es un subconjunto del campo de la inteligencia artificial concerniente con el diseño y desarrollo de algoritmos y técnicas que permiten a las computadoras “aprender”. Uno de los principales objetivos del área de “Machine Learning” es extraer información de los datos de manera automática a través de métodos estadísticos y computacionales. Otra de las características del área de “Machine Learning” es que está muy relacionado con la minería de datos y con las ciencias computacionales teóricas y estadísticas.

El área de “Machine Learning” tiene un amplio espectro de aplicaciones dentro de las áreas de procesamiento de lenguaje natural, reconocimiento de patrones, diagnósticos médicos, bioinformática, entre otras, y dichas aplicaciones se refieren, por ejemplo, a detectar fraudes de las tarjetas de crédito, análisis de la bolsa de valores, clasificación de secuencias de ADN, reconocimiento de voz y escritura, reconocimiento de objetos en visión por computadora.

Algunos sistemas de “Machine Learning” intentan eliminar la necesidad de la interacción del humano en el análisis de los datos, mientras que otros optan por un método colaborativo entre humanos y máquinas. La intuición humana no puede ser del todo eliminada, pues el diseñador del sistema debe de especificar como van a ser representados los datos y que mecanismos deberán de ser utilizados para buscar una caracterización de los datos. “Machine Learning” pudiera ser visto como un intento de automatizar partes del método científico.

Los algoritmos de “Machine Learning” se organizan dentro de una taxonomía, basado en la salida deseada del algoritmo. Algunos algoritmos comunes incluyen los siguientes tipos de aprendizaje [20]:

Aprendizaje supervisado. Es donde el algoritmo genera una función que mapea las entradas con las salidas deseadas. Una tarea clásica de este aprendizaje supervisado es el problema de clasificación, en donde el aprendiz necesita aprender (o aproximarse) al comportamiento de una función, la cual mapea un vector $[X_1, X_2, \dots, X_N]$ dentro de una de las distintas clases, esto mediante la búsqueda de varios ejemplos de entrada-salida de dicha función.

Aprendizaje no supervisado. Este tipo de aprendizaje modela un conjunto de entradas, y la diferencia con el aprendizaje anterior es que no etiqueta las entradas con alguna salida deseada. En otras palabras, este método de “Machine Learning”, genera un modelo que se ajusta a las observaciones (datos) y se distingue del aprendizaje supervisado por el hecho de no tener una salida “a priori”. El aprendizaje no supervisado trata típicamente a los objetos de entrada como un conjunto de variables aleatorias.

Aprendizaje semi-supervisado. Este aprendizaje combina el aprendizaje supervisado y el no supervisado para generar una apropiada función o clasificador.

Aprendizaje por reforzamiento. Durante este tipo de aprendizaje, el algoritmo aprende una política de como actuar dada una observación del mundo. Cada acción tiene algún impacto en el ambiente y el mismo ambiente provee una retroalimentación el cual guía dicho algoritmo de aprendizaje.

Uno de los problemas dentro del área de “Machine Learning” es construir programas computacionales que automáticamente se mejoren con la experiencia [20]. Sin embargo, muy poco se

ha reportado en la literatura sobre el aprendizaje del comportamiento dinámico de ST, el cual es útil para resolver el problema complejo de la predicción de ST ó para clasificación. Debido a lo anterior, en este trabajo realizaremos un aprendizaje supervisado del comportamiento dinámico de ST, tomando en cuenta únicamente la información de la imagen de ésta.

En la literatura el *aprendizaje* del comportamiento dinámico de ST es también conocido como el proceso de *entrenamiento* de ST. Este proceso de entrenamiento sirve para generar un modelo no lineal del conjunto de datos. En este trabajo se hace una comparación entre la “Técnica de Aprendizaje Basado en Imagen” (TABI) con otras dos técnicas de aprendizaje muy conocidas como lo son las redes neuronales y las máquinas de soporte vectorial para distinguir las diferencias de sus metodología con respecto a la que proponemos. Estas técnicas presentan procesos de entrenamiento muy diferentes a la nueva técnica documentada en esta tesis, por un lado el proceso de entrenamiento en la redes neuronales se utiliza para obtener los pesos adecuados de los nodos de la red neuronal de tal manera que durante este proceso se minimice el error de la función de aprendizaje, el entrenamiento se puede llevar a cabo de diferentes formas, como por ejemplo, con el conocido algoritmo Back-Propagation [31]. Por otro lado durante el proceso de entrenamiento de las máquinas de soporte vectorial se busca encontrar una función que minimice el riesgo de pérdida de datos [30]. Una breve explicación del funcionamiento de estas dos técnicas se presenta a continuación.

2.2.1. Redes neuronales

Los primeros trabajos en redes neuronales fueron realizados por McCulloch y Pitts en 1943 [41], donde buscaban la representación computacional de una neurona biológica. Las Redes Neuronales (RN) proveen un modelo no lineal cuya estructura se dice: “imita la estructura del cerebro humano” de alguna manera. Una RN se puede describir como un sistema interconectado de manera no lineal entre un conjunto de entradas con un conjunto de salidas. Con respecto a las ST las salidas de una RN pueden ser los puntos a predecir de la ST y las entradas a la RN pueden ser los puntos de entrenamiento de la ST con un cierto retardo.

La conexión entre los conjuntos de entrada y salida se hace típicamente vía una o más *capas ocultas* o *nodos*. La estructura de una RN es llamada usualmente *arquitectura* [31]. Escoger una arquitectura implica determinar el número de capas, el número de neuronas dentro de cada capa y como están conectados los valores de entrada, las capas ocultas y los valores de salida. La Figura 2.4 muestra un ejemplo clásico de RN con tres entradas y una capa oculta con dos neuronas.

La idea principal detrás de las RN, es que las variables dependientes, es decir, las entradas de la red, se filtren a través de una o más capas ocultas y éstas a su vez se filtren con otras capas ocultas y así sucesivamente hasta alcanzar la salida de la red, esto es, los valores deseados.

La fuerza de cada conexión es calculada mediante el parámetro llamado peso w [31]. Un valor numérico es calculado para cada neurona para cada período de tiempo t . Continuando con el ejemplo de la Figura 2.4, el valor de las entradas son $y_{1,t} = \text{unidad}$, $y_{2,t} = x_{t-1}$ y $y_{3,t} = x_{t-4}$. Sea w_{ij} el peso de las conexiones entre la entrada y_i y la j –ésima neurona. Se asume que este peso es constante a través del tiempo. Entonces, por cada neurona se calcula una suma lineal de pesos de las entradas, es decir, $\sum w_{ij}y_{i,t} = v_{j,t}$, para $j = 1, 2$. El analista debe de escoger una función llamada *función de activación*, para transformar los valores de v_j a un valor final para la neurona. La función que se utilizó en el ejemplo de la Figura 2.4 fue la *función logistic*, $z = 1 / (1 + e^{-v})$, el cual da valores dentro del rango $(0, 1)$.

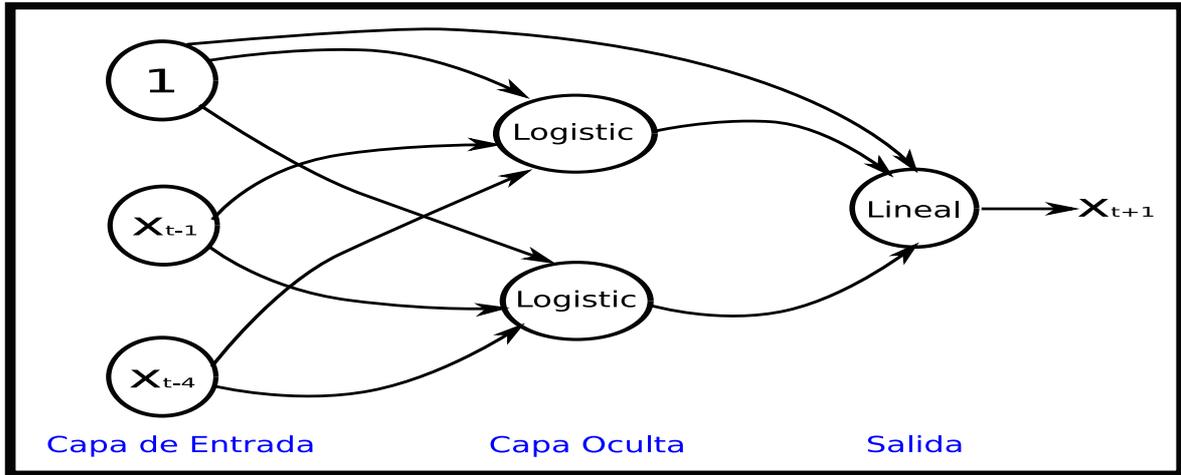


Figura 2.4: Ejemplo de arquitectura clásica para una red neuronal con tres entradas (valores de la ST con cierto retardo, esto es, x_{t-1} y x_{t-4} y una constante) una capa oculta de dos neuronas y una salida (por ejemplo el punto predicho).

Uno de los problemas de las RN es encontrar la arquitectura adecuada para realizar el entrenamiento de los puntos. En la actualidad se utilizan algoritmos evolutivos como una manera de encontrar la mejor arquitectura de la red neuronal y de esta manera garantizar un mejor funcionamiento [31]. Más adelante explicaremos cómo el aprendizaje del comportamiento dinámico de las ST a través de RN se utiliza para la predicción.

2.2.2. Máquinas de soporte vectorial (Support Vector Machines)

Esta técnica de Machine Learning aprende el comportamiento dinámico de ST a partir del siguiente esquema numérico [9]: divide la ST $x(t_1), x(t_2), x(t_3), \dots, x(t_n)$ en ventanas $w(x(t_i), \dots, x(t_{i+p-1}))$ de tamaño p , cuyo objetivo es encontrar una función $f: \mathbb{R}^p \rightarrow \mathbb{R}$ tal que $f(x(t_i), \dots, x(t_{i+p-1})) = x(t_{i+p})$ para cada $i \in \{0, N-p\}$ [25]. Utilizando una representación como la anterior es posible aprender el comportamiento dinámico de ST con Máquinas de Soporte Vectorial (SVM, Support Vector Machines), las cuales son máquinas de aprendizaje capaces de realizar clasificación binaria (reconocimiento de patrones) así como tareas de aproximación de funciones (cálculo de regresión).

Las SVM son sistemas de aprendizaje introducidos por Vladimir Vapnik [42] que utilizan un espacio de hipótesis de funciones lineales en un amplio espacio de características, las cuales son entrenadas con un algoritmo de optimización que implementa una tendencia de aprendizaje estadístico [43]. Dicha teoría de aprendizaje trata el problema de encontrar una función f de una clase de funciones $(f_\alpha)_\alpha \in \Lambda$ del tal forma que se minimice el riesgo esperado $R[f]$ respecto a una función de pérdida L , cuando la distribución de probabilidad de los ejemplos $P(x)$ y sus clasificaciones $P(y|x)$ no son conocidos, y tienen que ser estimados de un número finito de ejemplos $(x_i, y_i)_i \in I$.

$$R[f] = \int \int L(y, f(x)) dP(y|x) dP(x) \quad (2.5)$$

El algoritmo de las SVM resuelve este problema minimizando el riesgo regularizado $R_{reg}[f]$ el cual es la suma ponderada del riesgo empírico $R_{emp}[f]$ con respecto a los datos $(x_i, y_i)_{i=1 \dots n}$ y a un término de complejidad $\|w\|^2$.

$$R_{reg}[f] = R_{emp}[f] + \lambda \|w\|^2 \quad (2.6)$$

En su formulación básica, las SVM encuentran una función de decisión $f(x) = \text{sign}(\omega \cdot x + b)$ que minimiza el error de aprendizaje en el conjunto de entrenamiento y procura el mejor desempeño en la generalización.

Una de las principales características de las SVM es el uso de funciones de kernel para extender la clase de funciones de decisión entre las cuales se encuentran: los lineales, las funciones de base radial (RBF) ó Fourier. Para una revisión más completa del tema puede consultar [30][25].

2.3. Predicción de series de tiempo

Predicción es el proceso de estimar situaciones desconocidas, y en el caso particular de ST, $x(t_1), x(t_2), x(t_3), \dots, x(t_n)$, la predicción se refiere a encontrar sus valores futuros como por ejemplo $x(t_{n+h})$. El entero h es conocido como el *tiempo de predicción* o el *horizonte de predicción* (h de horizonte) y la predicción de $x(t_{n+h})$ realizado en el tiempo n para h pasos adelante será denotado por $\hat{x}_n(h)$. Nótese que es indispensable especificar el momento en que empieza la predicción y el tiempo de predicción, pues estos nos indicarán los puntos que se desean conocer en un futuro.

Una técnica de predicción se refiere a un procedimiento para calcular los valores futuros a partir de valores presentes y pasados. La gran mayoría de las técnicas de predicción, asumen que existe una relación entre los datos pasados del fenómeno con los datos futuros. Esta relación puede ser expresada en forma de una función y ésta es muy difícil de encontrar sobre todo si no se tiene un modelo o un conocimiento previo del sistema.

Las técnicas de predicción pueden ser ampliamente clasificadas en los siguientes tres tipos [31]:

1. *Predicción subjetiva* basada en un juicio subjetivo, intuitivo, con conocimiento “previo”, y alguna otra información relevante del fenómeno.
2. *Técnicas univariable* en donde la predicción depende únicamente de los valores presentes y pasados de una sola serie que se va a predecir, posiblemente incrementado por alguna función de tiempo como por ejemplo, la tendencia lineal.
3. *Técnicas multivariables* en donde la predicción de una variable dependiente, al menos de manera parcial, de uno o más valores de variables de otras series de tiempo llamadas variables informativas o predictoras. Las técnicas multivariables podrían depender de un modelo multivariado que involucre a más de una ecuación si las variables son conjuntamente dependientes.

En este trabajo se propone una nueva técnica univariable, la cual se basa únicamente de los valores presentes y no utiliza información extra. Un ejemplo de técnicas univariables son las redes neuronales y las máquinas de soporte vectorial [23].

2.3.1. Técnica de predicción con Multi-paso.

Para una ST, $x(t_1), x(t_2), x(t_3), \dots, x(t_n)$ muestreada de un sistema, es posible predecir los siguientes m puntos de la ST, esto es, los puntos, $x(t_{n+1}), \dots, x(t_{n+m})$, básicamente mediante las siguientes tres formas:

1. Predecir únicamente el siguiente punto en la serie, $x(t_{n+1})$.
2. Predecir directamente $x(t_{n+m})$ (*predicción directa*).
3. Predecir un punto adelante y usar esta predicción como parte del pasado para predecir el segundo punto y así sucesivamente hasta tener m predicciones (*Predicción Iterada*).

Aunque se mencionan tres formas de realizar la predicción, es posible notar que el punto uno es un subconjunto del punto dos, al realizar $m = 1$. Sin embargo, en este trabajo solo manejaremos la predicción iterada debido a que en la literatura únicamente encontramos resultados de esta predicción, y así tener un patrón de comparación para poder medir cuantitativamente la eficiencia de la técnica “Predicción Basada en Imagen” (PBI) respecto a otras técnicas como lo son las redes neuronales y las máquinas de soporte vectorial. Es por eso, que solo nos enfocaremos en la descripción de forma más detallada la predicción iterada a continuación.

Predicción iterada. Este tipo de predicción consiste en usar un conjunto de datos de entrada distinto para predecir cada valor de la serie [31]. Así, la predicción obtenida forma parte de una nueva ventana para predecir el valor siguiente, desplazando de esta forma la ventana una unidad de tiempo en cada predicción. De esta manera, se tiene una relación entre valores pasados como puntos de \mathfrak{R}^m y valores futuros como puntos de \mathfrak{R} . Con este último tipo de predicción los errores se van acumulando en el vector de entrada que forma la ventana, sobre todo en los últimos periodos de tiempo del horizonte de predicción. Así entonces, se trará de encontrar una función $f_i : \mathfrak{R}^m \rightarrow \mathfrak{R}$ con $i = 1, 2, \dots, n$ tal que:

$$\hat{x}(t_{n+1}) = f_1([x(t_n), x(t_{n-1}), x(t_{n-2}), \dots, x(t_{n-(m-1)})]) \quad (2.7)$$

$$\hat{x}(t_{n+2}) = f_2([\hat{x}(t_{n+1}), x(t_n), x(t_{n-1}), \dots, x(t_{n-(m-2)})]) \quad (2.8)$$

$$\hat{x}(t_{n+3}) = f_3([\hat{x}(t_{n+2}), \hat{x}(t_{n+1}), x(t_n), \dots, x(t_{n-(m-3)})]) \quad (2.9)$$

$$\dots \quad (2.10)$$

$$\hat{x}(t_{n+h}) = f_h([\hat{x}(t_{n+(h-1)}), \dots, \hat{x}(t_{n+1}), x(t_n), \dots, x(t_{n-(m-h)})]) \quad (2.11)$$

donde f_i son funciones no lineales que relacionan los valores futuros de la serie temporal con los valores pasados del horizonte de predicción. La longitud de la ventana es un parámetro que hay que determinar de manera óptima. Esto se puede realizar mediante el análisis de los valores pasados de la ST, disponibles en la base de datos histórica, que más influyen en los valores futuros de la variable que se quiere predecir. En la Figura 2.5 se observa un esquema de como se lleva a cabo la predicción iterada.

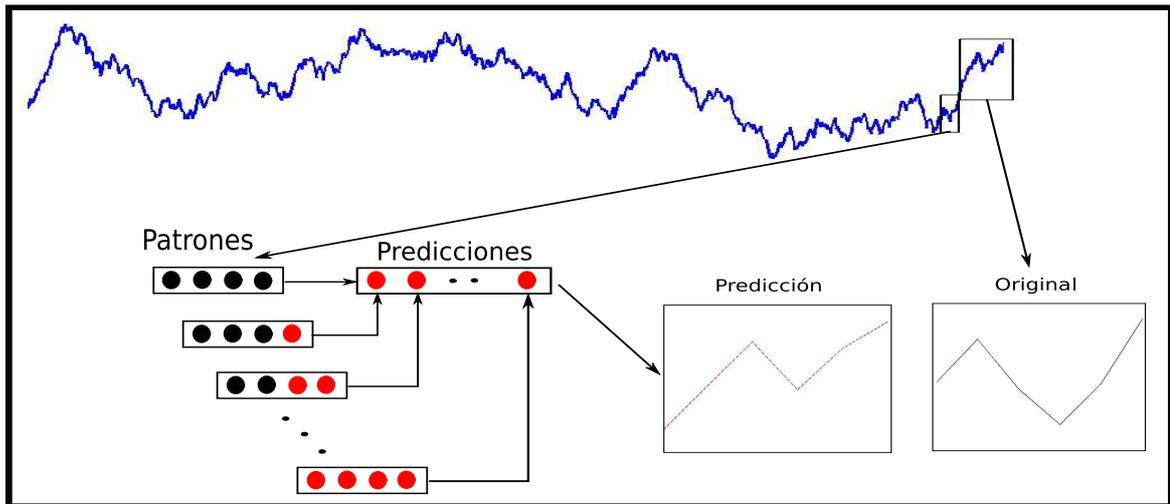


Figura 2.5: Ejemplo de cómo se lleva paso a paso la predicción iterada, a) primero se toman los últimos patrones aprendidos o entrenados (puntos negros) y posteriormente se genera la predicción de un punto (punto rojo), b) representa la nueva entrada de patrones el cual incluye el último predicho y los últimos tres patrones entrenados y a partir de esta nueva entrada se genera el nuevo punto predicho, c) representa el siguiente patrón de entrada, el cual ahora tiene la mitad de puntos predichos en etapas anteriores y la otra mitad representa los últimos dos patrones entrenados y este proceso se realiza hasta que se cumpla con los puntos que se desean predecir y como se observa en d) el patrón de entrada eventualmente estará conformado por todos los puntos predichos.

En la literatura existen muchos y muy diversos trabajos relacionados con la predicción de ST [30]. Ejemplos de estos trabajos son: Gooijer et al. [45] donde publica un trabajo en el cual recopila las técnicas de predicción que se han estado utilizando durante los últimos 25 años. Algunas de estas técnicas utilizan modelos matemáticos o estadísticos para predecir [32][33], otros autores usan RN [34][35] y [36]. También se pueden encontrar trabajos relacionados con la predicción de ST caóticas [32][37], económicas / financieras [33] o con diversas series de diferente naturaleza [11][38][39].

Debido a que nuestro patrón de comparación serán las técnicas de RN y SVM a continuación describiremos el proceso de predicción de estas técnicas no lineales.

Redes Neuronales

La idea de realizar predicción de ST con redes neuronales se remonta alrededor de 1960 [38] donde investigadores y científicos empezaron a darse cuenta de todas las ventajas que podía otorgar una red neuronal para realizar la predicción, como es la capacidad de generalización y la de poder aprender mediante la experiencia.

Aunque no existen reglas para diseñar una red neuronal, se debe de tomar en cuenta el número de capas (capa de entradas, ocultas y de salida), el número de nodos por capa, la interconexión entre éstos (feed-forward, feed-back, nodos completamente conectados u otros tipos de conexión), la presencia de bias, el tipo de función de transferencia en capas intermedias y de

salida; si se normalizarán los datos de entrada, la cantidad de datos de entrada (corresponden directamente a los nodos de entrada), el número de nodos de salida, el tipo entrenamiento (Batch o secuencial), el algoritmo de entrenamiento, el tamaño del conjunto de entrenamiento, los criterios de evaluación en el entrenamiento y en el conjunto de prueba (MSE, SSE, RMSE, etc.).

Por ejemplo, para una red neuronal con un nivel de capa oculta con H neuronas, la ecuación general de predicción para calcular la estimación de la salida $\hat{x}(t_{n+1})$ utilizando una selección de observaciones pasadas, $x(t_{n-j_1}), x(t_{n-j_2}), \dots, x(t_{n-j_k})$ como las entradas, pudiera ser descrita de la siguiente manera:

$$\hat{x}(t_{n+1}) = \phi_o \left(w_{co} + \sum_{h=1}^H w_{ho} \phi_h \left(w_{ch} + \sum_{i=1}^k w_{ih} x_{t-j_i} \right) \right) \quad (2.12)$$

donde $\{w_{ch}\}$ denota el peso de las conexiones entre las entradas constantes y las neuronas escondidas, para $h = 1, \dots, H$, y w_{co} determina el peso de las conexiones directas entre las entradas constantes y las salidas. Los pesos $\{w_{ih}\}$ y $\{w_{ho}\}$ determinan los pesos para otras conexiones entre las entradas y las neuronas escondidas y entre las neuronas y las salidas, respectivamente. Las dos funciones ϕ_h y ϕ_o indican la función de activación utilizada para la capa oculta y de la salida respectivamente.

Cuando la predicción es multipaso iterada se procede de dos maneras distintas. Primero, se puede construir una nueva arquitectura con varias salidas, dando la predicción a uno, dos, tres ó h pasos adelante, donde cada salida (predicción) tendrá diferentes pesos para cada conexión hacia las neuronas. Alternativamente, la predicción a un paso adelante pudiera ser “retroalimentada” remplazando así un valor de retardo (lag) de alguna variable de entrada. Entonces, la misma arquitectura puede ser utilizada para construir la predicción del segundo paso adelante y así sucesivamente. Esta última opción es la que generalmente se utiliza [23].

Máquinas de Soporte Vectorial

En los últimos años las SVM se han convertido en un tema de investigación de gran interés, dando lugar a una amplia gama de aplicaciones reportadas en la literatura entre ellas la predicción [27][28] y [29]. Sin embargo, en el campo de la predicción de ST, se encuentran pocos reportes de trabajos que hagan uso de las SVM, ya que debido a las características del algoritmo, no es posible obtener directamente una predicción de más de un punto de la serie a la vez, como es común en muchas otras técnicas. Es por ello, que en el trabajo [30] se presenta el empleo de una representación de ventana deslizante en el conjunto de entrenamiento del algoritmo SVM para realizar predicciones de múltiples puntos.

La representación de ventana deslizante para el conjunto de entrenamiento de la SVM, permite utilizar los últimos p valores de una ventana para predecir el siguiente valor de una ST. Esto es posible debido a que el corrimiento de la ventana a través de la serie permite a la SVM (mediante la función de kernel utilizada) estimar la dependencia existente entre cada uno de sus valores, de esta forma se obtiene la información necesaria para realizar la predicción del siguiente valor a partir de los valores anteriores de la serie. Entonces, para realizar una predicción de múltiples puntos es necesario crear un nuevo conjunto de entrenamiento cada vez que se realiza una predicción simple (de un solo punto). El nuevo conjunto se obtiene agregando al conjunto de entrenamiento original la siguiente ventana:

$$w' = (x_{i+p-2}, \dots, x_{i+p}) \quad (2.13)$$

donde x_{i+p} , es el valor correspondiente a la predicción anterior.

La representación de ventana con corrimiento, realizada en el conjunto de entrenamiento de la SVM, requiere que esta misma se retroalimente con los resultados previamente predichos, por lo anterior, el desempeño de la predicción se ve afectado debido al error acumulado conforme se avanza en los puntos predichos.

Es importante presentar un estado del arte de las técnicas que se utilizan actualmente para el aprendizaje de ST y algunas de sus aplicaciones como por ejemplo la predicción para poder ubicar nuestra investigación y así resaltar la contribución de la misma. En el siguiente capítulo se describe de manera más detallada la nueva técnica propuesta para aprendizaje del comportamiento dinámico de series de tiempo así como la su aplicación la predicción. También se mostrarán las diferencias que existen entre las dos técnicas explicadas en este capítulo, esto es, redes neuronales y máquinas de soporte vectorial con respecto a la nueva técnica propuesta.

Capítulo 3

Descripción de la técnica de aprendizaje basado en imagen

3.1. Introducción

En la literatura existen algunas técnicas para **aprender el comportamiento dinámico de series de tiempo**, desde técnicas lineales hasta técnicas no lineales [45]. Sin embargo, este proceso de aprendizaje está basado en obtener la información de los datos de la ST a través de ventanas en el tiempo, esto es, por medio de un corrimiento de ventanas para el caso de la técnica de soporte de máquinas vectoriales o a través de representar la información de los datos de la ST en nuevos vectores los cuales contienen los datos correspondientes a ciertos valores con un determinado retardo como es el caso de las redes neuronales o en algunos otros casos, los datos de la ST se mapean a otro espacio por ejemplo al *espacio de fase* [11]. En otras palabras, se busca la mejor representación de los datos para que las técnicas de aprendizaje funcionen mejor.

En este trabajo, se propone una nueva técnica de aprendizaje del comportamiento dinámico de ST. En esta técnica la representación de los datos es diferente a las técnicas anteriores, esto es, se divide la imagen o amplitud de la ST en n ventanas, en lugar de establecer ventanas sobre el eje del tiempo. Esta nueva forma de representación de los datos requiere de una nueva técnica de aprendizaje, la cual es ofrecida en este trabajo y llamaremos “Técnica de Aprendizaje Basado en Imagen” (TABI). La información que se obtiene de este proceso de aprendizaje es útil para aplicarlo a distintas problemáticas como por ejemplo: caracterización, clasificación, mapeo de series de tiempo a redes complejas, y para el complejo pero muy interesante problema de predicción.

Este capítulo está dividido principalmente en tres partes: la primera parte describe la propuesta de la nueva representación de los datos de entrada de las series de tiempo utilizando la información del eje imagen de la serie de tiempo; la segunda parte describe los pasos a seguir para la técnica TABI y finalmente, la última parte de este capítulo presenta una comparación de la metodología de esta nueva técnica respecto a las técnicas encontradas en la literatura, tanto en la representación de los datos de entrada como en los pasos necesarios para realizar el aprendizaje de series de tiempo de cada una de éstas.

3.2. Representación de los datos de entrada

Antes de continuar, es necesario determinar que en este trabajo el concepto de imagen se refiere al contradominio o codominio de la serie de tiempo (eje “y”), la cual puede ser vista como un tipo especial de función, esto es, para el matemático del siglo pasado el término “función” por lo general significaba una ecuación definida, como:

$$f(x) := x^2 + 3x - 5, \quad (3.1)$$

que asocia a cada número real x (dominio) otro número real $f(x)$ (imagen). A medida que se desarrolló la matemática, llegó a ser cada vez más claro que el requisito de que una función fuera una ecuación era indebidamente restrictivo y que entonces, sería útil una definición más general. La idea clave de esta actualización de la definición de función, es pensar en la gráfica de la función, es decir, en una colección de pares ordenados [12].

Definición 2 *Es posible analizar las series de tiempo como una función f cuyo dominio pertenece a \mathfrak{R} y su imagen o contradominio también pertenece a \mathfrak{R} , esto es:*

$$f : \mathfrak{R} \rightarrow \mathfrak{R} \quad (3.2)$$

donde f cumple con la siguiente propiedad:

$$\forall x \in \mathfrak{R} \exists! y \in \mathfrak{R} \mid f(x) = y. \quad (3.3)$$

donde x es el dominio de la función, que es el conjunto de valores para los cuales la función está definida, $f(x)$ es el conjunto imagen, también llamado contradominio o rango, y está formado por los valores que alcanza la función. Un ejemplo de esta definición de función se puede observar en la Figura 3.1 [13].

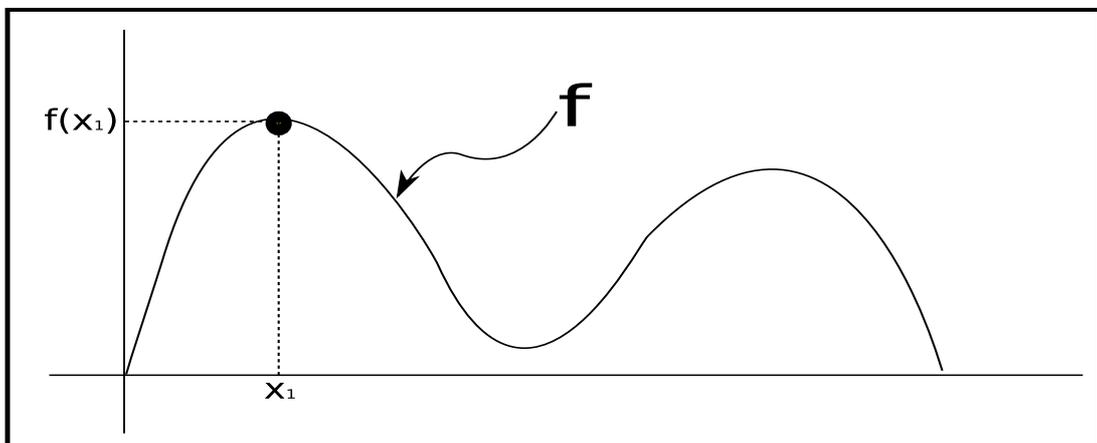


Figura 3.1: Ejemplo de la definición de función como una gráfica.

Es importante observar que una ST puede ser definida como una función y utilizar sus características matemáticas. Como hemos mencionado, dividiremos la amplitud de la ST en intervalos, esto es, vamos a dividir la imagen de la función f en un cierto número de intervalos, los cuales definiremos a continuación.

Definición 3 Si $y_1, y_2 \in \mathfrak{R}$, y $y_1 < y_2$ entonces el intervalo cerrado $[y_1, y_2]$ se define como el siguiente subconjunto en \mathfrak{R} :

$$[y_1, y_2] = \{y \mid y_1 \leq y \leq y_2\} \quad (3.4)$$

a los puntos y_1 y y_2 se les llama puntos terminales del intervalo cerrado $[y_1, y_2]$ y éstos están incluidos en él mismo.

Si ambos puntos terminales no están incluidos en el intervalo, entonces se le llama intervalo abierto:

$$1. (y_1, y_2) = \{y \mid y_1 < y < y_2\}. \quad (3.5)$$

Los conjuntos:

$$2. [y_1, y_2) = \{y \mid y_1 \leq y < y_2\} \quad (3.6)$$

y

$$3. (y_1, y_2] = \{y \mid y_1 < y \leq y_2\}, \quad (3.7)$$

son llamados intervalos semiabiertos (o semicerrados) determinados por los puntos terminales y_1 y y_2 .

Al definir un intervalo, es necesario definir sus dos puntos terminales ó cotas: la inferior y la superior de la siguiente manera:

Definición 4 La cota inferior $ci(y_1)$ de un intervalo $[y_1, y_2]$, también denotado como \underline{y} , es definido de la siguiente manera:

$$\underline{y} = ci([y_1, y_2]) := \sup\{a \in \mathfrak{R} \cup \{-\infty, \infty\} \mid \forall y \in [y_1, y_2], a \leq y\} \quad (3.8)$$

Definición 5 Por otro lado la cota superior $cs(y_2)$ de algún intervalo $[y_1, y_2]$, también denotado por \bar{y} está definido de la siguiente manera:

$$\bar{y} = cs([y_1, y_2]) := \inf\{b \in \mathfrak{R} \cup \{-\infty, \infty\} \mid \forall y \in [y_1, y_2], y \leq b\} \quad (3.9)$$

Entonces, \underline{y} es el número más grande a la izquierda del intervalo $[y_1, y_2]$ y \bar{y} indica el número más chico a la derecha del intervalo $[y_1, y_2]$ [13]. Por ejemplo, supongamos el intervalo cerrado $[-3, 7]$ entonces $\underline{y} = -3$ y $\bar{y} = 7$.

Una vez que fueron definidas las cotas superior (\underline{y}) e inferior (\bar{y}) entonces es posible definir la anchura de un intervalo no vacío de la siguiente manera:

$$a[y_1, y_2] \triangleq \bar{y} - \underline{y} \quad (3.10)$$

Es posible notar que el tipo de intervalo que empleamos en este trabajo, se utiliza para encapsular puntos que pertenecen a la imagen de la serie de tiempo, pero además necesitamos determinar el tiempo de estos puntos encapsulados, es decir, es necesario encerrar los puntos a analizar en dos intervalos: el dominio y la imagen de la serie de tiempo. Lo anterior, se realiza mediante el vector de intervalos, el cual nos ayuda en nuestros propósitos y definiremos a continuación.

3.2.1. Vector de intervalos

Un vector de intervalos es un subconjunto de \mathfrak{R}^n , el cual puede ser definido como un producto Cartesiano de n intervalos cerrados y lo denotaremos de la siguiente manera: $[Box]$. Esto se debe a que cuando no existe ambigüedad sobre dicho vector de intervalos, se le puede llamar simplemente “caja” (Box) [13] y se expresa matemáticamente de la siguiente manera:

$$[Box] = [\underline{x}_1, \overline{x}_1] \times [\underline{y}_1, \overline{y}_1] \times [\underline{y}_2, \overline{y}_2] \times \dots \times [\underline{y}_n, \overline{y}_n] \quad (3.11)$$

El *iésimo* componente de la caja $[Box_i]$ es la proyección de $[Box]$ sobre el *iésimo* eje. El conjunto vacío de \mathfrak{R}^n debiera ser escrito como; $0 \times \dots \times 0$ porque todos los componentes del intervalo son vacíos. Expresiones tales como;

$$[Box] = 0 \times [0, 1], \quad (3.12)$$

están prohibidas, debido a que, $[0, 1]$ no es una proyección de $[Box]$ sobre el segundo eje. Esto garantiza que la notación de cajas sea única [13]. El conjunto de todas las cajas n -dimensionales será denotada por $I\mathfrak{R}^n$. Cajas no vacías son paralelepipedos n -dimensionales alineados con sus respectivos ejes. La Figura 3.2 ilustra el ejemplo de una caja donde $n = 2$, es decir, tiene dos ejes, con $[Box] = [\underline{x}_1, \overline{x}_1] \times [\underline{y}_1, \overline{y}_1]$.

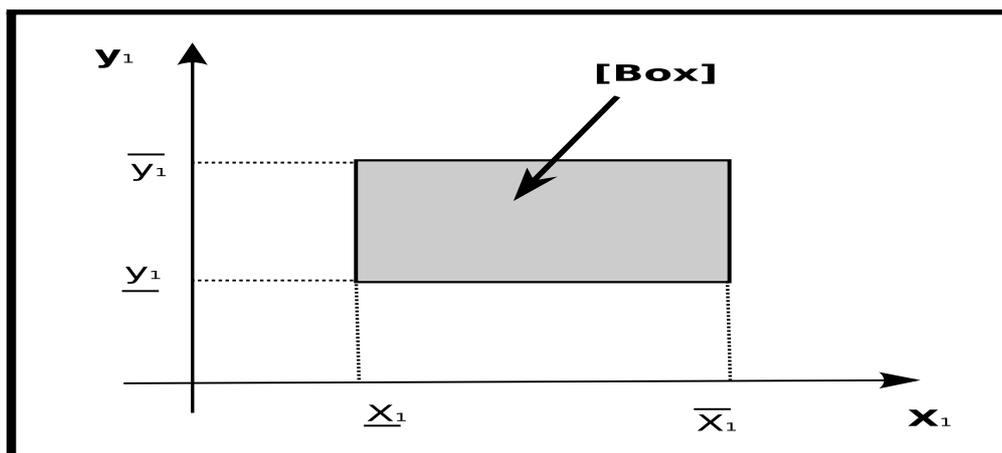


Figura 3.2: Ejemplo de una caja $[Box] = [\underline{x}_1, \overline{x}_1] \times [\underline{y}_1, \overline{y}_1]$ de $I\mathfrak{R}^2$

Las cajas o vector de intervalos que se van a emplear en este trabajo serán de la siguiente manera:

$$[Box] = [x_1, x_2] \times [y_{1i}, y_{2i}] \quad \text{para } i = 1, \dots, n. \quad (3.13)$$

La expresión anterior significa que el eje de las x , es decir, el dominio de la función f no cambia, pero lo que si cambia son los intervalos de la imagen o bien el eje de las y . Un ejemplo de lo anterior se observa en la Figura 3.3. A partir de las cajas definidas en (3.13) podemos hacer la siguiente definición:

Definición 6 Sea $f : \mathfrak{R} \rightarrow \mathfrak{R}$, es posible definir la siguiente función:

$$\text{área}_f : [Box] \rightarrow \mathbb{N} \cup \{0\} \cup \{\infty\} \quad (3.14)$$

esto es, la nueva función área_f va de los intervalos de la imagen $[\underline{y}_1, \overline{y}_1]$ de la función f hacia los números \mathbb{N} incluyendo el cero y los infinitos. Entonces, la función área se puede definir de la siguiente manera:

$$\text{área}_f [Box] = \text{card}([\underline{y}_1, \overline{y}_1] \cap \{f(x) \mid x \in [\underline{x}_1, \overline{x}_1]\}) \quad (3.15)$$

En otras palabras, la función área_f indica el número de puntos que se encuentran dentro del área de cada una de las cajas (vector de intervalos). Un ejemplo más claro se puede observar en la Figura 3.3, en donde el número de puntos dentro del $\text{área}_f [Box] = 86$ puntos.

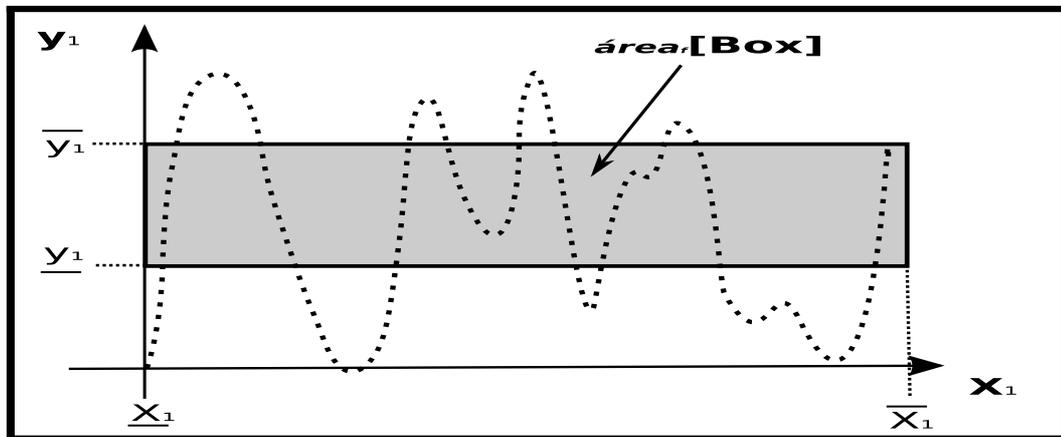


Figura 3.3: Ejemplo de como se encapsulan los puntos dentro de una caja y cómo a partir de ésta se obtiene el $\text{área}_f [Box]$ correspondiente. Además se observa que para esta figura el $\text{área}_f [Box] = 86$ puntos.

Las series de tiempo que se trabajarán en esta investigación son discretas como la que se observa en la Figura 3.4.

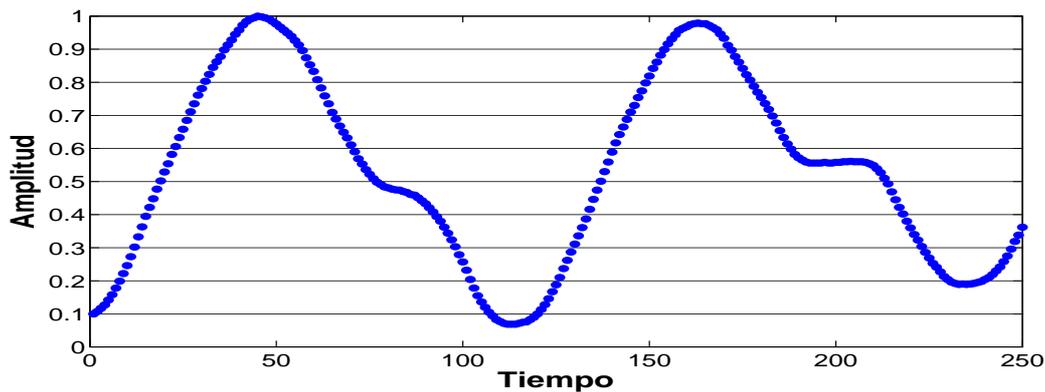


Figura 3.4: Secuencia de puntos que conforman una ST. Se observa que la imagen de la ST está dividida en 10 *cajas* de igual anchura, esto es, $a[y_1, y_2] = 0.1$. Es debido a esto, que las cajas para esta ST serían las siguientes $[0, 250] \times [0, 0, 1] \times [0, 1, 0, 2] \times [0, 2, 0, 3] \times \dots \times [0, 9, 1]$ y como se observa los intervalos que cambian pertenecen a la imagen de la ST.

Las definiciones de intervalos que mencionamos en esta subsección van a ser útiles al momento de dividir la imagen de la ST en un cierto número de cajas [*Box*], las cuales nos proporcionarán más información de ciertos momentos de la dinámica de la ST en un determinado intervalo de la imagen. De esta manera, se aprenderá de manera local todo el comportamiento de la dinámica de la ST como se describirá a continuación.

3.2.2. Método de submetas aplicado a series de tiempo

En la cultura popular, **divide y vencerás** hace referencia a un refrán que implica resolver un problema difícil, dividiéndolo en partes más simples tantas veces como sea necesario, hasta que la resolución de las partes se torna obvia. Este método está basado en la resolución recursiva de un problema dividiéndolo en dos o más subproblemas de igual tipo o similar. La solución del problema principal se construye con todas las soluciones encontradas. Entonces, como se puede notar, este método tiene el inconveniente de no encontrar una solución para alguno de los subproblemas [14].

Es por eso que en la literatura existe una técnica que utiliza la filosofía de **divide y vencerás** (pero sin su inconveniente) y ésta es conocida como *submetas* (subgoals, en inglés), la cual fue publicada por Newel en 1960 [5], este principio define un sentido fundamental de resolver subproblemas que están ligados serialmente, mientras estos mismos subproblemas aparentemente se resuelven de manera independiente. En otras palabras, por una parte, todos los subproblemas tienen un objetivo en particular, pero por otro lado, todos los subproblemas siguen un objetivo global. La diferencia de esta técnica con la de **divide y vencerás** es que la técnica de submetas divide un problema que tiene solución secuencial y en algunas ocasiones no necesita la solución de todos los subproblemas para alcanzar el objetivo global.

Después, en 1963 Donald Michie publicó varios artículos [3, 4] en el que utilizaba este principio de submetas para generar una máquina que de manera mecánica aprendiera a jugar el juego del gato (tic-tac-toe, en inglés) y éste argumentaba que era más fácil dividir el tablero del gato (problema difícil) en varios subtableros (problemas fáciles) donde cada uno de estos

tableros contenía la jugadas frecuentes para dicho tablero, es decir, cada uno de éstos contenían *información local* y por otro lado también se contaba con toda la información del sistema llamada *información global* que contenía la información del tablero principal. Como es bien sabido, la mayoría de las veces los juegos se resuelven de manera secuencial, esto es, hasta no saber la posición del oponente no es posible continuar jugando.

La idea principal de submetas secuenciales se puede utilizar en este trabajo de la siguiente manera, dividimos una ST en intervalos con una cierta área en su amplitud llamadas *cajas*, donde cada *caja* encapsula un cierto número de puntos, los cuales contienen información local sobre un cierto segmento de su amplitud, es decir, de su imagen. Lo anterior es posible hacerlo debido a que las series de tiempo por definición son secuenciales como se observa en la Figura 3.5.

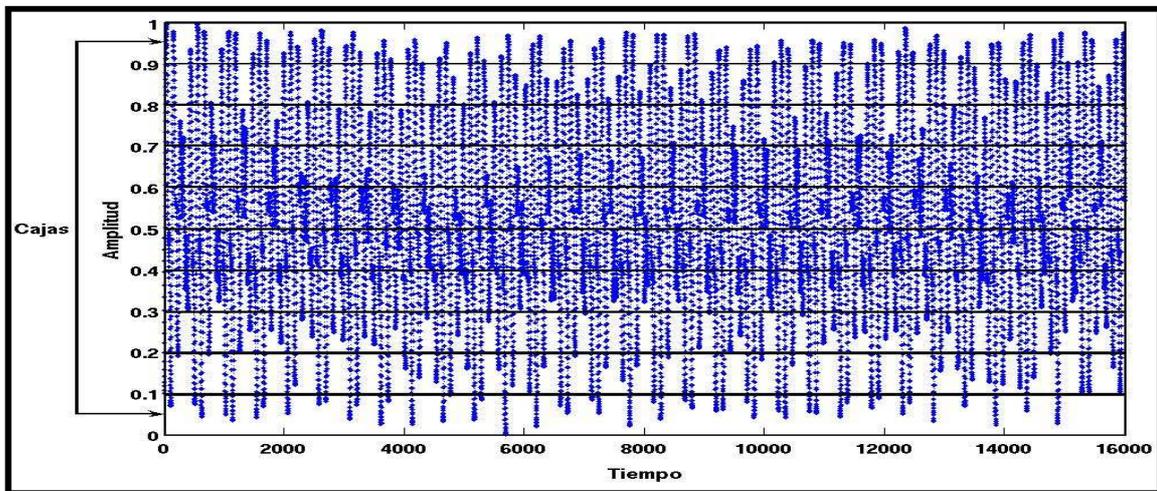


Figura 3.5: Ejemplo de como se dividió la ST con respecto a sus amplitudes. Como se puede observar en este análisis la información con la que trabajamos es la que se encuentra dentro de cada *caja* la cual representa la información únicamente de la amplitud.

La nueva representación de los datos de entrada de las series de tiempo propuesta en este trabajo consta de los siguientes pasos:

1. Evaluar si la ST es estacionaria.
2. Normalizar las series de tiempo.
3. Encontrar el número de cajas adecuadas para recolectar la información apropiada obtenida de la amplitud de la ST de manera automática, de tal manera que todas las cajas manejen la misma cantidad de información.

Prueba de estacionaridad

Una serie de tiempo se dice que es estrictamente *estacionaria* si sus propiedades no se ven afectadas por algún cambio en su tiempo de origen [22]; esto es, si la distribución de probabilidad de alguna ST asociada con m observaciones $x(t_1), x(t_2), x(t_3), \dots, x(t_m)$, generadas en

cualquier conjunto de tiempo $t_1, t_2, t_3, \dots, t_m$, es la misma que la asociada con m observaciones $x(t_{1+k}), x(t_{2+k}), x(t_{3+k}), \dots, x(t_{m+k})$, generada en los tiempos $t_{1+k}, t_{2+k}, t_{3+k}, \dots, t_{m+k}$, entonces la ST es estacionaria. Por lo tanto, para que una ST discreta sea estrictamente estacionaria, la distribución de probabilidad de cualquier conjunto de observaciones no debe de ser afectada por cualquier desplazamiento hacia adelante o hacia atrás del eje del tiempo por alguna constante k , ver sección 2.1.4.

Para determinar si una ST es estacionaria se evalúan los siguientes pasos:

1. Dada una ST definida por una secuencia de N puntos, se define una ventana de m puntos de la ST donde $m \in N$ tal que, $m < N$.
2. Se definen las estadísticas media y varianza para probar la estacionaridad de series de tiempo (ver sección 2.1.4).
3. Cuando se han cumplido con los pasos anteriores, entonces se procede a aplicar la media y la varianza a cada una de las ventadas definidas para la ST y esta ventana se desplaza hacia adelante o hacia atrás a través del eje del tiempo.
4. Finalmente, se grafican los resultados de la media y la varianza, obtenida de cada ventana. Si estos valores muestran una distribución uniforme, entonces significa que la ST es estacionaria, pero por el contrario, si los resultados no muestran una distribución uniforme, entonces la ST es no estacionaria.

Una vez definida la prueba de estacionalidad se procede a realizar una normalización de la amplitud de la ST de la siguiente manera.

Normalización de la serie de tiempo

El siguiente paso para llevar a cabo la técnica TABI pudiera parecer sencillo, pero no es así, pues es uno de los pasos primordiales para comparar el rendimiento de esta técnica de aprendizaje con las encontradas en la literatura, el paso al que nos referimos es la *normalización*.

En los primeros experimentos los puntos de la ST fueron normalizados dentro del intervalo $[0, 1]$ y como se observa únicamente se trabajaba con puntos positivos. Esta normalización se llevó a cabo de la siguiente manera:

$$normalizar1(i) = \frac{x(t_i) - \text{mín}(x)}{\text{máx}(x) - \text{mín}(x)} \forall i = 1, \dots, n \quad (3.16)$$

donde $x(t_i)$ corresponde a cada uno de los puntos de la ST original, $\text{mín}(x)$ es el valor mínimo de la imagen de la función f o ST y $\text{máx}(x)$ representa el punto máximo de la imagen de la función f . El resultado de la ecuación (3.16) será una ST dentro del intervalo $[0, 1]$.

Sin embargo, en la literatura se encontró que técnicas como las redes neuronales y las máquinas de soporte vectorial, también normalizan las series de tiempo pero éstas dentro del intervalo $[-1, 1]$. Entonces para poder hacer una comparación del rendimiento de esta técnica TABI con las ya mencionadas fue necesario normalizar las series de tiempo al intervalo $[-1, 1]$ de la siguiente manera:

$$normalizar(i) = \frac{2 * (x(i) - \text{mín}(x))}{\text{máx}(x) - \text{mín}(x)} - 1 \quad \forall i = 1, \dots, n \quad (3.17)$$

Como se observa se obtiene una ST normalizada dentro del nuevo intervalo $[-1, 1]$. Es importante mencionar que durante el proceso de normalización no se pierde información, es decir, el comportamiento dinámico de la serie de tiempo al realizar esta normalización no cambia. Un ejemplo se puede observar en la Figura 3.6, cuando se realiza la normalización en la ST “Lorenz” y el comportamiento dinámico de esta ST no se ve afectado.

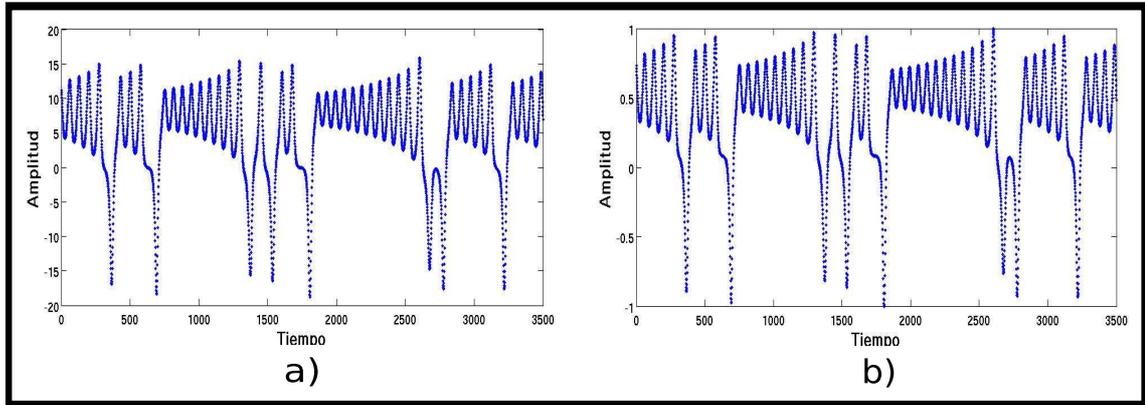


Figura 3.6: Ejemplo de la normalización de una ST. En a) se observa la ST “Lorenz” original, la cual se observa que los valores de su imagen están dentro del intervalo $[-20, 20]$ y en b) se observa la serie de tiempo normalizada dentro del intervalo $[-1, 1]$. Es claro observar como la dinámica de la serie es preservada en a) y b).

División de la imagen de la función f en “Cajas”

El objetivo de emplear este principio de submetas en series de tiempo es encontrar nueva información a través de técnicas nuevas. Con esta nueva información es posible: predecir el comportamiento de las series de tiempo y también caracterizar las series de tiempo. Esta división de la imagen de la ST en *cajas* se realizó de acuerdo a los siguientes tres tipos:

Primer tipo de división: misma anchura

La primera división de la imagen de la ST, la cual llamamos “misma anchura”, es como su nombre lo indica respecto a la anchura de los intervalos definidos en la ecuación (3.10), esto es, se divide la imagen de la ST en 10 *cajas* en donde cada uno de estas *cajas* $[y_1, y_2]$ contenía la misma anchura calculada de la siguiente manera:

$$a(y_1, y_2) = \frac{y_2 - y_1}{N} \quad (3.18)$$

donde y_2 y y_1 son las cotas de la imagen de la función f y N indica el número total de cajas en las que la imagen de la ST se va a dividir. Por ejemplo, la imagen de la ST al principio fue normalizada dentro del intervalo $[0, 1]$ por lo tanto $y_2 - y_1 = 1$ y donde N corresponde al número total de *cajas* en que fue dividida la imagen de la ST, en nuestro caso, $N = 10$ por lo tanto la anchura de cada *caja* fue de $a(y_1, y_2) = 0.1$. Por otro lado, cuando la ST se normaliza dentro del intervalo $[-1, 1]$ el valor de $y_2 - y_1 = 2$ y si conservamos el mismo número de cajas

que en el intervalo de normalización anterior, es decir, $N = 10$, la anchura estaría definida como $a[y_2, y_1] = 0.2$.

Pero como se puede observar, al dividir las amplitudes de la ST en 10 *cajas* con $a[y_1, y_2] = 0.1$ (ver Figura 3.7a)), la distribución de los puntos de la ST dentro de cada *caja* formaban la mayoría de las veces una distribución Gaussiana. Debido a lo anterior, siempre se tendrá que las *cajas* que se encuentran en el centro de la imagen de las ST tienen casi el doble de información que las *cajas* que están en las orillas, lo que hacía poco eficiente el proceso de aprendizaje pues los puntos no estaban distribuidos de manera uniforme entre las *cajas* y esto genera que algunas cajas tengan más información que aprender que las demás. En el trabajo [15] se realizaron las primeras pruebas de esta técnica mediante la división de la imagen de la series de tiempo en 10 *cajas* con una misma anchura y donde los resultados muestran que aún con este inconveniente se tiene una buena precisión, ahí se pueden encontrar los resultados de los primeros experimentos. Un ejemplo de la distribución Gaussiana que se encontró de dividir la imagen de la función f en *cajas* con $a[y_1, y_2] = 0.1$ se observa en la Figura 3.7. Además, otra de las desventajas de esta división, es que aun cuando se estuviera analizando alguna ST de comportamiento dinámico diferente la división de la imagen sería la misma, esto es, la división de la imagen de series de tiempo es constante.

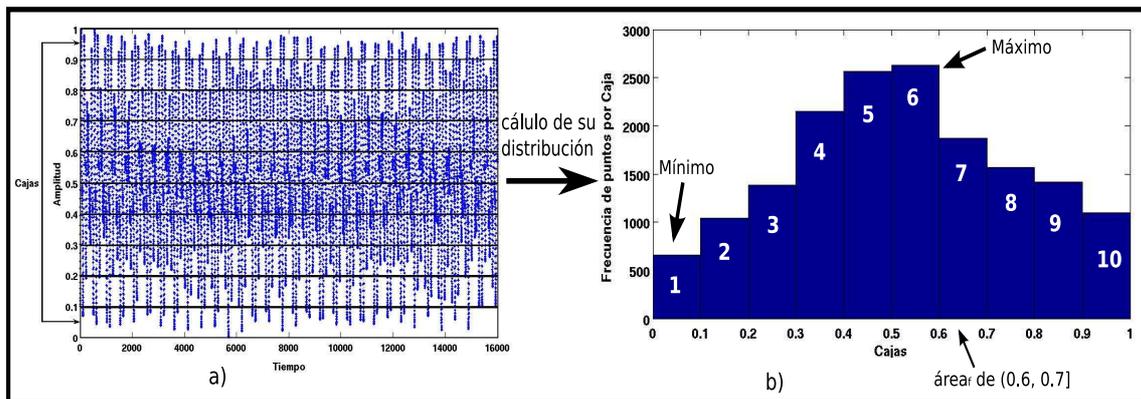


Figura 3.7: a) ST dividida en 10 *cajas* con $a[y_1, y_2] = 0.1$ y b) Distribución Gaussiana de los puntos de la ST dentro de cada *caja*. Como se observa dentro de las *cajas* 5 y 6 se tiene la mayor frecuencia de puntos y en las *cajas* 1, 2 y 10 se observa la menor frecuencia de puntos.

Segundo tipo de división: por frecuencias

Debido al problema de distribución gaussiana de los puntos dentro de las cajas al dividir la imagen de la ST con la misma anchura, se propone una nueva forma de hacer la división de *cajas* y esta nueva división la llamamos “división por frecuencias” y se realiza de la siguiente manera: tomando en cuenta la ecuación (3.15) donde se definió el concepto de área_f podemos definir las cotas inferior y superior de cada *caja* de manera dinámica, y como podemos observar la diferencia entre este tipo de división con respecto a la anterior es que ahora tomaremos en cuenta el número de puntos dentro de cada *caja*, (mediante el área_f) y no de acuerdo a su anchura. Con lo anterior se controla que todas las cajas tengan igual número de información (puntos) (ver Figura 3.7). Por ejemplo, si $\text{área}_f = 100$ puntos significa que cada *caja* tendrá la

misma cantidad de puntos haciendo que la distribución de los puntos sea uniforme y todas las *cajas* tengan igual número de puntos. Otra de las ventajas de hacer la división de la imagen de la serie de tiempo a través del número de puntos es, que este tipo de división es variable, esto es, el número de *cajas* y el área para una ST dada será diferente respecto a otra ST con el mismo o diferente comportamiento dinámico, pues depende por completo de los puntos de la ST.

Antes de definir las nuevas cotas de cada *caja*, primero es importante determinar cómo calcular el número adecuado de puntos dentro de cada *caja*. Para lo cual, se puede utilizar la división de cajas con misma anchura a (y_1, y_2] = 0.1 definida en el primer tipo de división y una vez definida esta división, es posible calcular el número de puntos contenidos en cada *caja*, con lo cual se puede obtener la frecuencia de puntos dentro de cada *caja*, donde la frecuencia se puede expresar de la siguiente manera:

$$Frec(i) = area_f (y_{1i}, y_{2i}] \text{ for } i = 1, 2, \dots, N, \quad (3.19)$$

donde $Frec(i)$ es un vector que contiene las frecuencias acumuladas de los puntos que pertenecen a una cierta *caja* (y_1, y_2], las cuales se calcularon siguiendo la ecuación (3.18) para $N = 10$ que corresponde al número total de *cajas*. Obteniendo así una distribución muy parecida a la que se calculó en la Figura 3.7. De acuerdo a esta distribución de puntos por cada *caja* se procede a calcular el número de puntos adecuado para las nuevas *cajas*, esto mediante el nuevo cálculo de la función $área_f$ como:

$$área_f = \frac{\text{máx}(Frec(i)) + \text{mín}(Frec(i))}{2} \quad (3.20)$$

donde $\text{máx}(Frec(i))$ se refiere a la *caja* con mayor frecuencia de puntos y $\text{mín}(Frec(i))$ se refiere a la *caja* con menor frecuencia de puntos de la ST. La ecuación (3.20) es el promedio entre la frecuencia máxima y la frecuencia mínima de puntos entre las cajas. Por ejemplo, supongamos que se obtiene una distribución muy parecida a la de la Figura 3.7, en donde la *caja* 6 tiene la frecuencia máxima igual a 2,628 puntos y la *caja* 1 tiene la frecuencia mínima igual a 656 puntos, entonces $área_f = 1,642$ puntos dentro de cada *caja*. Por lo que se definirán las cotas de cada *caja* de tal forma que cada una contenga aproximadamente 1,642 puntos.

Tercer tipo de división: por porcentaje de información total

Como es posible observar la división por frecuencias depende de la distribución gaussiana de los puntos dentro de cada *caja* al realizar la división por misma anchura, pero en algunas ocasiones esta distribución gaussiana de los puntos no se generaba al realizar la división por misma anchura y como consecuencia al realizar la división por frecuencias, el número de puntos que deberían permanecer dentro de las *cajas* era muy grande, generando un bajo rendimiento en la técnica de aprendizaje. Es por eso, que propusimos una tercera opción que no dependiera del resultado de alguna de las divisiones anteriores y a este tipo de división lo llamamos “porcentaje de información total”. El cual como su nombre lo indica, determina el número de puntos dentro de cada *caja* mediante un cierto porcentaje del número total de puntos de la serie de tiempo de la siguiente manera:

$$área_f = size(x) * 0.10 \quad (3.21)$$

donde $size(x)$ corresponde al número total de puntos de la ST, supongamos que estamos analizando una ST de 16,000 puntos como la que se muestra en la Figura 3.7a), entonces $size(x) = 16,000$ puntos y el multiplicador 0.10 calcula el 10 por ciento del total de puntos. En otras palabras, la expresión (3.21) significa que el número de puntos dentro de cada caja ($área_f$) es igual al 10 por ciento del total de puntos de la ST, continuando con el ejemplo anterior $área_f = 1,600$ puntos dentro de cada *caja*. Esta medida se utiliza cuando las frecuencias de las *caja* definidas con una misma anchura no muestra una distribución gaussiana entre las *cajas*. Es posible observar que el multiplicador puede ser variable, por ejemplo, puede ser igual a 0.05 que corresponde al 5% del total de puntos para definir el $área_f$ de una *caja*.

Como es posible observar, las cotas de las cajas del segundo y tercer tipo de división de la imagen son variables, pues dependen del número de puntos dentro de cada *caja* y no como en la división por misma anchura en el que dicho número de *cajas* las cotas son constantes. Es por eso que el siguiente paso se realiza únicamente para cuando se implemente el tipo de división dos y tres.

Cálculo de las cotas superior e inferior de las cajas. Una vez que se definió el número adecuado de puntos para cada *caja*, se procede a calcular las cotas superior e inferior de las nuevas *cajas* de acuerdo al diagrama de flujo que se observa en la Figura 3.8.

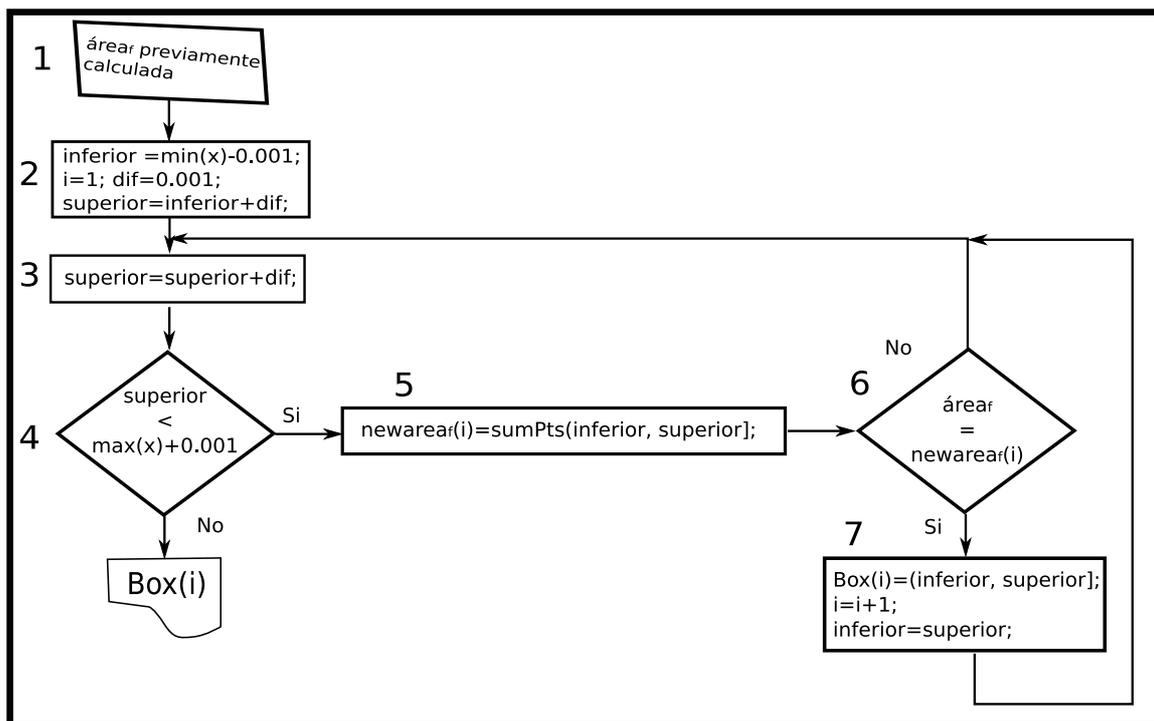


Figura 3.8: Diagrama de flujo utilizado para el cálculo de las cotas superior e inferior de las nuevas *cajas* con distribución uniforme de puntos.

Como se observa el diagrama de flujo de la Figura 3.8 tiene como parámetro de entrada el número de puntos que va a contener cada *caja*, recordemos que este valor se calculó efectuando el cálculo de la ecuación (3.20) ó la ecuación (3.21). La mejor forma de explicar la lógica del

código es mediante un ejemplo, por lo que usando el ejemplo del tipo de división dos, en donde el número de puntos para cada *caja* fue de 1,642 puntos, el primer paso sería definir el número de puntos de cada *caja* en este ejemplo igual a 1,642. El siguiente paso es inicializar el valor de la cota inferior, la cual va a ser igual a $(0 - 0.001)$ si la ST está normalizada dentro del intervalo $[0, 1]$ o igual a $(-1 - 0.001)$ si la ST está normalizada en el intervalo $[-1, 1]$. En el paso 3, calculamos el valor de la cota superior y al principio lo inicializamos con el valor de la cota inferior más un valor de diferencia, esto es, $superior = 0.002$ si la ST está normalizado entre $[0, 1]$, pero si ésta está normalizada entre $[-1, 1]$, $superior = -1.000$, después, en el paso 4 se pregunta si la cota superior que se definió ya excedió al número máximo de la amplitud de la ST, esto es, si la cota superior es mayor que 1, si no es así, entonces, en el paso 3 se calcula el número de puntos dentro de este nuevo intervalo ($newárea_f$) y solo si este número de puntos es igual al que se estableció al principio supongamos igual a 1,642 puntos (paso 6), entonces, en el paso 7 se definen las cotas inferior y superior de la primera *caja*, entonces el valor de la cota superior ahora es la cota inferior de la siguiente *caja*, pero sí el número de puntos que se encuentra en $newárea_f$ no tiene el número de puntos definidos para cada *caja*, entonces, se va incrementando el valor de la cota superior y se evalúan el número de puntos contenidos en la *caja*, este paso se reitera hasta obtener el número adecuado de puntos. Este proceso se repite hasta que se hayan definido todas las nuevas *cajas*.

En algunas ocasiones, se observó que ciertas *cajas* no cumplían con el número de puntos definidos para cada *caja* (con el $área_f$ definida en la ecuación (3.20) ó en la ecuación (3.21)), los puntos dentro de alguna determinada *caja* superaban el $área_f$ establecida, por tal motivo no era posible establecer las cotas inferior y superior de cada una de las *cajas*. Para resolver este problema se tomó la siguiente medida:

$$área_f máxima = área_f + error \quad (3.22)$$

$$área_f mínima = área_f - error, \quad (3.23)$$

donde $error$ es el número de puntos de tolerancia para que se definan las cotas de una cierta *caja* $(y_1, y_2]$. Con esta nueva medida se modifica el paso número 6 del diagrama de flujo de la Figura 3.8 de la siguiente manera:

$$newárea_f(i) \leq área_f máxima \ \& \ newárea_f(i) \geq área_f mínima. \quad (3.24)$$

Con este nuevo principio de dividir la imagen de la función f a partir de su adecuada $área_f$ de puntos dentro de cada *caja* fue posible cambiar la distribución Gaussiana de las *cajas* en una distribución uniforme, lo que significa que todas las *cajas* contienen casi el mismo número de puntos, es decir, la misma cantidad de información. Lo anterior se puede observar en la siguiente Figura 3.9.

Lo que podemos resumir de todo este paso es que el tamaño de cada *caja* se puede realizar de manera constante y variable con los procedimientos que se explicaron anteriormente. Para determinar el número de *cajas* de manera constante solo es necesario determinar la anchura de éstas, la cual será igual para todas las *cajas*, por otro lado si empleamos el tipo de división variable, el número y tamaño de las *cajas* dependerá de la distribución de la información sobre el eje imagen y con esta metodología de generar las *cajas* de manera variable se uniformiza la cantidad de información dentro de cada *caja* al momento de realizar la generación de las *cajas* y con ésto el proceso de aprendizaje se realizará de manera más óptima.

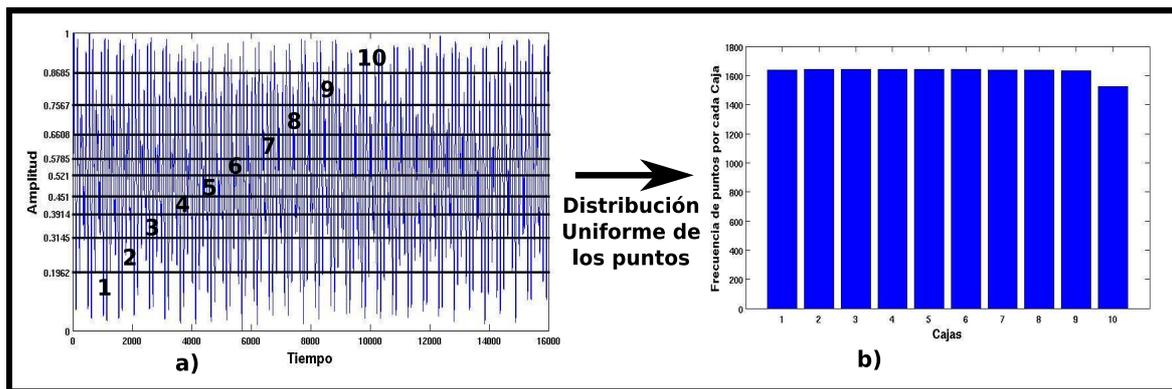


Figura 3.9: Ejemplo de la división de la imagen de f de acuerdo al nuevo principio definido por área_f . a) Se observan las nuevas *cajas* enumeradas del 1 al 10, en donde las centrales están más estrechas que las de los costados. b) Se observa la distribución que se obtiene de las nuevas divisiones, la cual es uniforme.

3.3. Técnica de aprendizaje de ST

Para empezar con el proceso de aprendizaje, primero es necesario definir el concepto de “*aprendizaje*” que se va a estar manejando durante este trabajo. Para fines prácticos tomaremos tres definiciones que encontramos en la literatura, el primero es la definición que se encuentra en la real academia española [18], en donde *aprendizaje* lo definen de la siguiente manera:

Acción y efecto de aprender.

y el concepto de aprender lo definen de la siguiente manera:

Adquirir el conocimiento de algo por medio del estudio o de la experiencia.

Con la definición anterior no queda claro qué significa “*aprendizaje*”, por lo que consultamos una segunda definición propuesta por Herbert Simon [19] la cual dice:

Aprender es cualquier cambio en un sistema el cual permite ejecutar mejor la misma tarea o alguna tarea del mismo tipo la siguiente vez que se ejecute.

Pero algunos científicos criticaron esta definición, pues argumentaban que no cualquier cambio que mejora el rendimiento es *aprendizaje*. Otros investigadores [9] afirmaban que los humanos (sistemas) pueden procesar información la cual les permite realizar alguna tarea sin siquiera conocer dicha tarea. Por ejemplo, supongamos que estamos caminando en el centro de la ciudad de Chetumal y pasamos enfrente de una librería que no sabíamos que estaba ubicada ahí y aunque en ese momento no necesitamos la información de que existe una librería ubicada en el centro, aún así, almacenamos la información que recién aprendimos. Ahora supongamos que días después alguien nos pregunta si conocemos alguna librería en el centro de la ciudad de Chetumal, entonces podremos hacer uso de la información que previamente almacenamos y podremos contestar la pregunta. Es decir, se almacenó información que parecía poco útil y que en el futuro resultó ser muy útil. Sin embargo, Michalski [21] opinaba diferente y propuso la siguiente definición:

Aprender es la construcción y modificación de representaciones de lo que se ha experimentado.

Entonces, tomado en cuenta las tres definiciones anteriores de *aprendizaje* se hace una unión de éstas para formar la definición de *aprendizaje* que aquí, manejaremos y es la siguiente:

Definición 7 *Aprendizaje es adquirir y acumular el conocimiento de alguna tarea a través de la experiencia, la cual permitirá una mejor ejecución de dicha tarea la siguiente vez que se ejecute.*

Tomando en cuenta la definición anterior de aprendizaje podemos empezar a definir los pasos que se llevan a cabo en esta técnica TABI para *aprender* el comportamiento dinámico de una ST a través de la información de su imagen.

Una vez que la imagen de la función f fue normalizada dentro del intervalo $[0, 1]$ ó $[-1, 1]$ y posteriormente dicha imagen fue dividida en n cajas, utilizando alguna de las tres maneras de dividir vistas en la subsección anterior, se procede a efectuar el paso de aprendizaje, en donde la experiencia acumulada puede ser vista como una matriz en la que se va almacenando la información de puntos que caen dentro de una cierta *caja*.

El algoritmo de aprendizaje que se propone en este trabajo es probabilístico, debido a que su resultado se basa en la solución de algunas decisiones al azar, de tal forma que en promedio, obtiene una buena solución al problema planteado para cualquier distribución de los datos de entrada [10]. Es decir, a partir de los mismos datos de entrada se pueden obtener distintas soluciones y en algunos casos soluciones erróneas.

La técnica de aprendizaje que plateamos se puede resumir principalmente en lo siguiente:

1. Generar a partir de números aleatorios acotados una secuencia que siga el comportamiento dinámico de la ST original. En este paso se almacena la información que va a servir para que cada caja aprenda el comportamiento de ciertos puntos.
2. Calcular el error de aprendizaje entre la serie original y la serie calculada.

Entonces, el primer paso del aprendizaje es representar a la ST como el siguiente vector:

$$x(t_1), x(t_2), x(t_3), \dots, x(t_i) \quad (3.25)$$

donde t representa el tiempo y cumple con la condición $t_1 < t_2 < t_3 < \dots < t_i$ y donde i es el número total de puntos de la ST. Para que se pueda observar mejor el funcionamiento de la técnica que se propone en este trabajo, vamos a hacer un acercamiento de los primeros 150 puntos de la ST "Qperiodic2" [31], con el fin de ejemplificar paso a paso este proceso, en la Figura 3.10.

Así entonces, se toma un punto de la ST en el instante t_i a partir de $i = 1$ y se pregunta a qué *caja* pertenece dicho punto, mediante la siguiente validación definida de la siguiente manera:

Definición 8 *Para determinar si algún punto de la ST pertenece a alguna determinada caja, se realiza la siguiente condición: si $x(i) > ci(\text{Box}(j))$ y $x(i) \leq cs(\text{Box}(j))$ entonces $x(i) \in \text{Box}(j)$, $\forall i = 1, \dots, N$ y $j = 1, \dots, m$, donde i corresponde al número total de puntos de la ST y m corresponde al número total de cajas.*

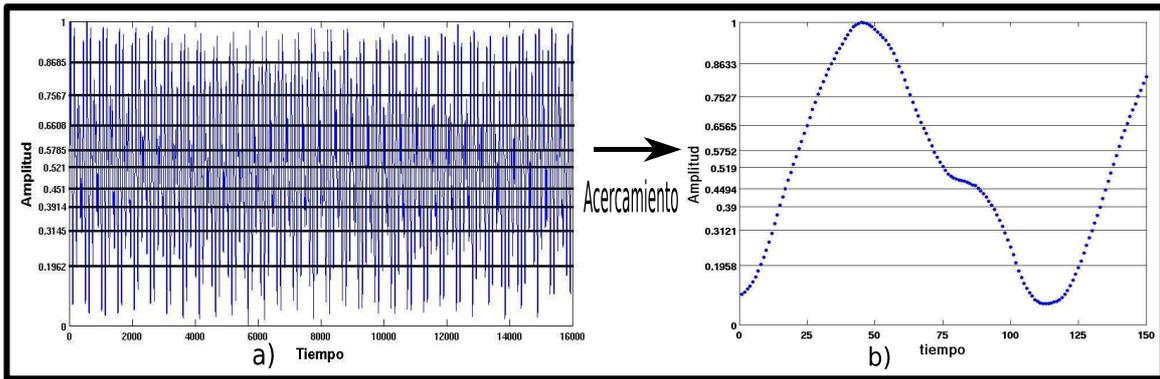


Figura 3.10: Acercamiento de los primeros 150 puntos de la ST original, la cual como se observa en a) ésta consta de más de 16,000 puntos y en b) se observa este acercamiento de la ST “Qperiodic2”.

La definición anterior significa que si el punto $x(i)$ de la ST cumple con la validación definida en (8), entonces dicho punto se encuentra dentro de la *caja Box* (j). Por ejemplo, supongamos que deseamos analizar el punto en el instante $i = 1$, y si observamos la Figura 3.11a) obtenemos que el valor del punto $x(1) = 0.12$, entonces para determinar a que caja pertenece este punto se efectúa la condición de la definición 8 y como se observa en la Figura 3.11b) el punto, $x(1)$ pertenece a la *caja* número 1, siendo el intervalo de esta *caja* igual a $(0.00, 0.20]$.

El siguiente paso es determinar la tendencia de los puntos, esto es, si éstos están “incrementando” o “decrementando”. En este trabajo se consideró diferente la información de la ST cuando está subiendo que cuando está bajando. Para diferenciar el sentido de los puntos de la ST utilizamos el manejo de una bandera llamada *band*, la cual es igual a 1 cuando los puntos están subiendo, pero cuando los puntos están bajando la bandera se cambia a 0. Al principio la bandera se inicializa a 1, pues se observó que en la gran mayoría de las series de tiempo el primer punto (t_1) con respecto al segundo (t_2) su sentido era hacia arriba, esto es, los puntos empiezan subiendo.

Definición 9 Si los puntos de la ST están subiendo ($band = 1$) y el siguiente punto es menor que el punto anterior entonces, la serie empieza a bajar y en ese momento se cambia la bandera a $band = 0$, esto es, los puntos de la ST cumplieron con lo siguiente: si $x(i) < x(i - 1)$ entonces $band = 0$.

Definición 10 Pero si los puntos de la ST están bajando ($band = 0$) y el siguiente punto es mayor al punto anterior, entonces se dice que los puntos empiezan a subir y por lo tanto se cambia la bandera a $band = 1$, es decir, los puntos de la ST cumplieron con la siguiente expresión: si $x(i) > x(i - 1)$ entonces $band = 1$.

Si continuamos con el ejemplo de la Figura 3.11, observamos que el punto $x(1)$ tiene la bandera inicial igual a $band = 1$. Pero cuando $i = 9$ y $x(9) = 0.82$ entonces hay un cambio en el sentido de los puntos, pues éstos han llegado al límite de la última *caja* ($caja=5$), lo que

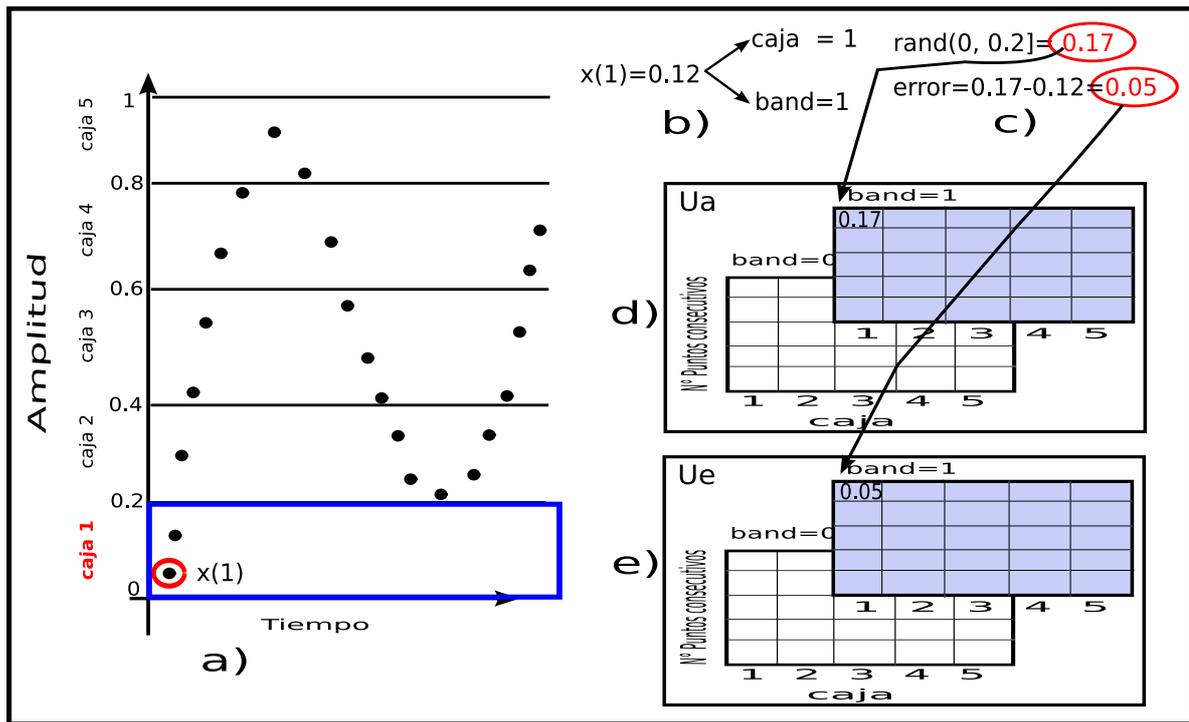


Figura 3.11: Proceso de aprendizaje. En a) se observa cómo se van obteniendo los puntos de la ST original para posteriormente en b) extraer la información de la caja a la que pertenece dicho punto y el sentido de éste. En c) se observa el procedimiento aleatorio para aprender el comportamiento dinámico de la ST, así como el cálculo del error entre el valor generado menos el original, después en d) se observa como se almacena el valor generado en su matriz correspondiente y finalmente en e) se observa como se almacena el error obtenido.

significa que los siguientes puntos tienen un descenso y por lo tanto, el valor de su bandera cambia a 0.

Antes de continuar describiendo este proceso de *aprendizaje* primero es necesario definir la estructura de las matrices en donde se van a almacenar los valores aprendidos de la ST. Estas matrices cumplen con la siguiente definición:

Definición 11 Sea A la matriz definida por $A := (a_{j,k,l}) \in \mathbb{R}^{m \times n \times o} \quad \forall 1 \leq j \leq m, 1 \leq k \leq n$ y $1 \leq l \leq o$, donde m representa el número de puntos consecutivos que se encuentran en una determinada caja sin cambiar el sentido de su bandera, n se refiere a las cajas en las que puede caer algún punto de la ST y o representa la tercera dimensión que es equivalente a la información que se almacena para cada una de las posibles banderas, que puede ser 0 (baja) ó 1 (sube) (ver Figura 3.12).

Aquí manejaremos dos matrices, una llamada U_a en donde se van a almacenar los datos que se van a ir generando y otra llamada U_e , en donde se van a ir almacenando sus respectivos errores. Un ejemplo de la estructura de esta matriz se observa en la Figura 3.12, donde podemos observar que la matriz A es equivalente a U_a , pues lo que almacena son los valores calculados de la ST original.

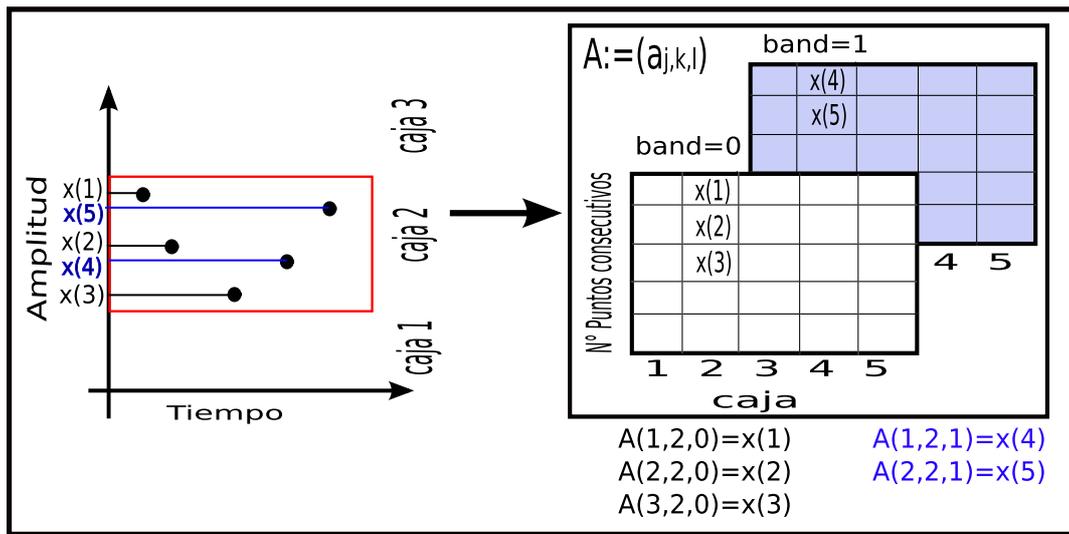


Figura 3.12: Ejemplificación de las matrices utilizadas para almacenar los valores generados con la técnica de aprendizaje. A la izquierda se observan los puntos calculados de la ST, donde los primeros tres puntos tienen una tendencia descendente y los otros ascendente, a la derecha se muestra el almacenamiento de los valores calculados dentro de la matriz A , finalmente en la parte inferior derecha se observa cómo obtener los datos almacenados en esta matriz.

Entonces, continuando con el ejemplo de la Figura 3.11, para el punto $x(1)$, contamos con la siguiente información: $j = 1$, pues es el primer punto que entra a la *caja* $k = 1$ y el sentido de los puntos es ascendente, por lo tanto, $band = 1$, pues los puntos están subiendo. Tomando en cuenta la información anterior sabemos que es la primera vez que algún punto entra a esta *caja* y por lo tanto no se cuenta con información previa, es decir, las dos matrices U_a y U_e están en ceros, por lo tanto se genera un valor aleatorio acotado por los límites de la *caja* ($Box(k)$) de la siguiente manera:

$$punto(i) = rand(Box(k)), \quad (3.26)$$

es decir, se genera un valor aleatorio entre el intervalo $[0.0, 0.2]$ que es son las cotas de la *caja* 1, donde se encuentra el punto actual (ver Figura 3.11c)). Este primer punto se realiza de manera aleatoria debido a que no tiene un conocimiento previo de lo que tiene que aprender, el único conocimiento previo es que no debe exceder los límites establecidos por cada *caja*, es por eso que se acota la generación aleatoria del primer punto y de los posteriores.

Esto es, para $i = 1$ donde $x(1) = 0.12$, observamos que este punto es la primera vez que visita la caja $k = 1$, y el sentido de este punto es ascendente ($i = 1$), por lo tanto, se genera un valor aleatorio dentro del intervalo de la caja, el cual en este caso es igual a $(0.0, 0.2]$ y este nuevo valor ($punto(1) = 0.17$) lo comparamos con el punto original ($x(1) = 0.12$), calculamos el error que se produjo ($error(1) = 0.5$) (ver Figura 3.11c)) y finalmente almacenamos los valores calculados en sus matrices correspondientes como se observa en la Figura 3.11d) y e).

Posteriormente, para $i = 2$ donde $x(2) = 0.19$ observamos en la Figura 3.13a) que este punto sigue perteneciendo a la caja 1 y observamos que el sentido de los puntos tampoco ha cambiado, el procedimiento para calcular el siguiente valor, es muy parecido al procedimiento anterior, con la diferencia de que este nuevo punto ($punto(2)$) debe cumplir que sea mayor al punto anterior ($punto(1)$), esto con la finalidad de conservar el sentido ascendente entre estos puntos y para lograr lo anterior ahora se escogerá un valor aleatorio entre el último valor que se generó ($punto(1) = 0.17$) y se conserva la cota superior del intervalo de la caja 1, esto es, el nuevo valor se escoge de manera aleatorio entre el intervalo $(0.17, 0.2]$ y observamos que $punto(2) = 0.18$, el cual procederemos a calcular su error de aprendizaje que sería igual a -0.01 y finalmente, almacenamos estos resultados en sus respectivas matrices como se observa en la Figura 3.13d) y e). Esto es, se puede observar que este procedimiento se realizará para calcular el siguiente valor mientras que éste pertenezca a la misma caja y el sentido de los puntos no cambie y únicamente se modifica la cota inferior por el valor que se obtuvo en el punto anterior ($punto(n - 1)$) y se conserva el valor de la cota superior de la caja a la que pertenecen dichos puntos, podemos notar que cuando el sentido de los puntos es descendente ($band = 0$), lo que permanece del intervalo de decisión es la cota inferior y lo que estará variando será la cota superior. Cuando el siguiente punto entra a la caja $k = 2$ y éste sigue conservando su sentido ($band = 1$), supongamos el punto en el instante $i = 3$ (ver Figura 3.13a)), calculamos este punto de manera aleatoria entre el intervalo $(0.2, 0.4]$ y es igual a $punto(3) = 0.39$, éste lo procederemos a comparar con el original ($x(3) = 0.30$) y obtenemos el error ($error(8) = 0.09$) y posteriormente lo almacenamos en su matriz correspondiente.

Este procedimiento se realizará en este ejemplo hasta el punto $x(8)$ pues como se puede observar en la Figura 3.13b) sigue conservando las características de mismo sentido que los puntos anteriores y observamos que el valor que se genera es igual a $rand(0.8, 1] = 0.98$ y el error es igual a -0.08 .

Si observamos el siguiente puntos, esto es, $x(9)$ notamos que el sentido de los puntos ahora es descendente y procederemos de la siguiente manera, escogemos una valor aleatorio entre el intervalo $(0.8, 1]$ y el valor que se genere ahora se almacenará en la matriz blanca U_a de la Figura 3.13d) y el error correspondiente se almacena en la matriz de color blanco U_e de la Figura 3.13e), que como se puede observar dichas matrices ahora almacenan la información de los puntos con sentido descendente.

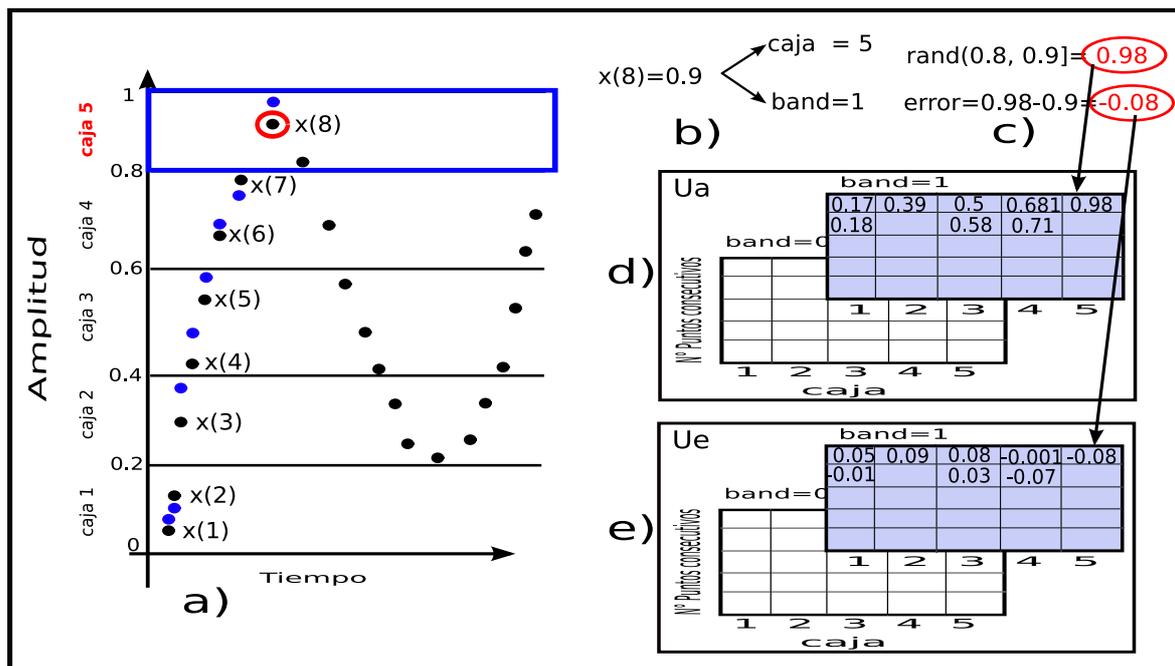


Figura 3.13: Proceso de aprendizaje de los primeros 8 puntos. En a) se observa como se van obteniendo los puntos de la ST original para posteriormente en b) extraer la información de la caja a la que pertenece dicho punto y el sentido de éste. En c) se observa el procedimiento aleatorio para aprender el comportamiento dinámico de la ST, así como el cálculo del error entre el valor generado menos el original, después en d) se observa como se almacena el valor generado en su matriz correspondiente y finalmente en e) se observa como se almacena el error obtenido.

Una vez que se genera un nuevo punto se procede a calcular el error entre el punto calculado y el punto original de la siguiente manera:

Definición 12 Si $x(i)$ corresponde al punto original de la serie de tiempo y punto (i) se refiere al punto que se genera aleatoriamente calculado en (3.26) en el instante i , entonces el error se define de la siguiente manera: $error(i) = punto(i) - x(i)$, (ver Figura 3.13c). Este error (i) nos va a indicar la precisión de nuestra estimación del valor deseado.

Como se observa de la definición anterior, el valor de $error(i)$ puede ser tanto negativo como positivo, esto es, cuando el valor de $error(i)$ es positivo ($error(i) > 0$) significa que el $punto(i)$ que se calculó es mayor que el valor original $x(i)$, es decir, $punto(i) > x(i)$.

Por otro lado, cuando $error(i)$ es negativo ($error(i) < 0$) significa que el $punto(i)$ que se calculó es menor que el valor original $x(i)$, es decir, $punto(i) < x(i)$. En otras palabras, este $error(i)$ nos va a ayudar a ajustar los valores generados para cada $caja$ cuando ésta sea visitada¹ posteriormente.

Una vez que se calcula el punto aprendido y su error correspondiente, entonces, el valor generado se almacena en la matriz Ua y el error correspondiente se almacena en la matriz Ue

¹Consideramos una visita cuando un punto de la serie de tiempo ingresa más de una vez a una $caja$.

como se observa en la Figura 3.14.

Matriz de valores generados		Matriz de errores	
Ua(:, :, 1) =		Ue(:, :, 1) =	
0.0980	0.2569	-0.0011	0.0579
0.1190	0.2722	0.0124	0.0502
0.1208	0	0.0038	0
0.1470	0	0.0200	0
0.1687	0	0.0267	0
0.1949	0	0.0369	0
0.1953	0	0.0173	0

Figura 3.14: Ejemplo de cómo se van generando las matrices de datos Ua y la de errores Ue . En este caso, debido a que únicamente se han calculado los primeros 9 puntos, la primera columna de ambas matrices muestra los resultados obtenidos para la caja 1 y la segunda columna muestra los resultados de los primeros dos puntos dentro de la caja 2.

Sin embargo, ¿qué sucede si un punto $x(i)$ de la ST ingresa a una *caja* que ya fue previamente visitada (entrenada), por ejemplo, supongamos que en el instante $i = 126$ los puntos vuelven a visitar la *caja* 2 en el instante i , con sentido ascendente ($band = 1$). Entonces, lo que se hace es lo siguiente: se obtiene el valor que se generó de la última visita de la matriz Ua ($punto(últimaVisita(i)) = Ua(i, 2, 1)$) y también se obtiene el valor del error que se produjo con el punto generado en esta última visita en el instante i de la matriz Ue ($error(últimaVisita(i)) = Ue(i, 2, 1)$) y éste lo comparamos con la siguiente condición: $error(últimaVisita(i)) > 0.009 \mid error(últimaVisita(i)) < -0.009$, nótese que usamos esta comparación debido a que es importante no solo conocer el error de predicción si no también la posición del *punto* (i) relativa al punto $x(i)$, dicha suposición se obtiene con el signo del error como se explica a continuación.

Sí el *punto*($últimaVisita(i)$) tiene un $error(últimaVisita(i)) > 0.009$, significa que el *punto* calculado en la última visita en el mismo instante fue muy grande con respecto al original, por lo que en esta visita es necesario generar un *punto* más chico y esto se hace acotando el intervalo sobre el cual se va a escoger de manera aleatorio este nuevo valor de la siguiente manera:

$$punto(i) = \text{rand}(newci, newcs) \quad (3.27)$$

donde $newci = punto(i - 1)$ es la nueva cota inferior, correspondiente al último punto calculado en la visita actual y $newcs = punto(últimaVisita(i))$ es la nueva cota superior y éste corresponde al punto que se calculó en la última visita en el instante i . De esta manera se define y acota nuevamente el intervalo $(newci, newcs]$ de donde va a ser recalculado el nuevo *punto* (i).

Por ejemplo, supongamos ahora los puntos de la ST de la Figura 3.15, donde se observa el punto $x(8)$, el cual es el primer punto que visita por primera vez la caja 2 (ver Figura 3.15a)),

recordemos que del entrenamiento de esta ST se generan dos matrices Ua y Ue , las cuales muestran una estructura parecida a la que se observa en la Figura 3.15b), de donde obtenemos la matriz correspondiente al sentido de los puntos que en este caso, si observamos nuevamente esta figura es ascendente para los puntos que se encuentran encerrados en un rectángulo, los cuales pertenecen a la caja 2, posteriormente, generamos los puntos de manera aleatoria (ver Figura 3.15c)) y los almacenamos en la matriz $Ua(:, :, 1)$ y los errores los almacenamos en la matriz $Ue(:, :, 1)$ (ver Figura 3.15d)) y hasta ese momento no se tenía información previa en dicha caja. Como comentamos anteriormente los puntos vuelven a visitar la caja 2 cuando están en el instante $i = 126$ como se observa en la Figura 3.15e) y aquí es donde decimos que empieza el aprendizaje, como mencionamos anteriormente, la manera de saber si en la visita anterior se calculó un punto cercano al original es mediante el error, el cual obtenemos de la siguiente manera $error(\acute{u}ltimaVisita) = Ue(1, 2, 1)$ y si observamos nuevamente la Figura 3.15d) podemos notar que éste es igual a 0.057, lo que significa que el punto que se calculó en la última visita fue mayor que el original ($error(\acute{u}ltimaVisita(i) > 0.009)$), lo que significa que el nuevo valor para este punto debe de ser menor y para eso necesitamos extraer de la matriz Ua el valor que se calculó en esta última visita ($i = 8$), el cual es igual a $Ua(1, 2, 1) = 0.256$ y esto nos va a servir para modificar el intervalo sobre el cual de manera aleatoria se estimará el siguiente punto y siguiendo con la ecuación 3.27, este nuevo punto se escogerá entre el intervalo $(0.195, 0.256]$, podemos observar que por ser el primer punto de la caja dos, la cota inferior de este nuevo intervalo de decisión fue igual a la cota inferior de la caja 2 y lo que cambió fue la cota superior, la cual observamos es igual al valor calculado en la visita anterior ($i = 8$) (ver Figura 3.15f)) y el nuevo valor lo almacenamos en $Ua(1, 2, 1) = 0.219$ y como podemos observar en la Figura 3.15g), se sobrescribe este valor sobre el que se calculó en la visita anterior y calculamos el valor de error de esta nueva visita, en donde si el valor $x(126) = 0.210$ y el valor calculado es igual a $punto(126) = 0.219$, el error entre estos es igual a 0.009, el cual se almacena en la matriz $Ue(1, 2, 1) = 0.009$.

Ahora analizemos el caso del instante $i = 127$, donde observamos que el último valor calculado ($Ua(2, 2, 1) = 0.272$) y obtuvo un error igual a 0.050, que es muy parecido al instante anterior, pero la diferencia ahora es que las cotas superior e inferior del nuevo intervalo de decisión van a cambiar de la siguiente manera: la cota inferior va a ser igual al valor calculado en el instante $i = 126$ y la cota superior va a ser igual al valor que se calculó en la visita anterior pero del segundo punto (en el instante $i = 9$) y de acuerdo a la ecuación 3.27 tenemos el nuevo intervalo de decisión $(0.219, 0.272]$ del cual se estimará el nuevo punto para el instante $i = 127$, el cual se almacenará en $Ua(2, 2, 1)$ y el error de este nuevo punto se almacenará en la matriz $Ue(2, 2, 1)$, si observamos por última vez la Figura 3.15g) podemos notar que los errores son menores en la segunda visita con respecto de la primera.

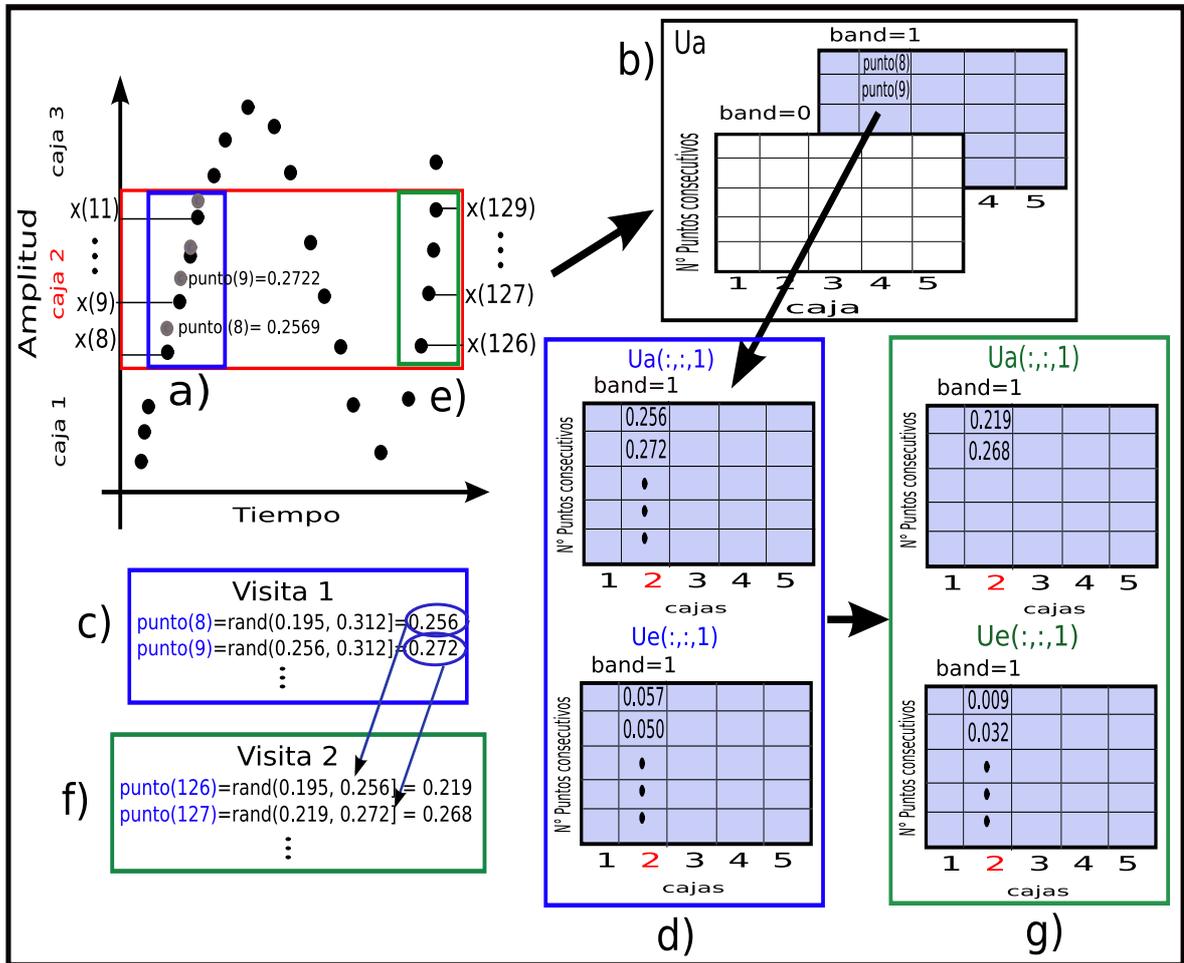


Figura 3.15: Esta figura muestra paso a paso cómo se lleva a cabo el proceso de aprendizaje, el inciso a) nos muestra un acercamiento de los primeros puntos de la ST a analizar, además se observa en un rectángulo los puntos que pertenecen a la caja 2 durante la primera visita cuando tienen el sentido ascendente (puntos negros), b) nos muestra la estructura de las matrices que utilizamos para almacenar la información, c) muestra el cálculo de los primeros puntos estimados, d) muestra cómo se almacenan los resultados en su respectiva matriz, los cuales se observan en a) como puntos grises, en e) se observa la segunda visita que hacen los puntos sobre la misma caja, posteriormente en f) se ajustan los nuevos intervalos de decisión y los resultados finalmente se almacenan en g).

Del ejemplo anterior podemos observar que este nuevo intervalo está diseñado para que el siguiente número que se genere de manera aleatoria sea más chico que el *punto* (*últimaVisita* (*i*)) calculado en la visita anterior y por lo tanto se van ajustando los puntos generados con respecto a la ST que se desea aprender. La información que se almacena del ejemplo anterior se observa en la Figura 3.16.

Matriz de valores generados Visita 1		Matriz de valores generados Visita 2									
Ua(:, :, 1) =		Ua(:, :, 1) =									
caja 1	caja 2	caja 1	caja 2	caja 3	caja 4	caja 5	caja 6	caja 7	caja 8	caja 9	caja 10
0.0980	0.2569	0.0222	0.2193	0.3652	0.4394	0.4792	0.5297	0.6307	0.6710	0.8471	0.9778
0.1190	0.2722	0.0480	0.2681	0.3746	0.4444	0.4914	0.5385	0.6385	0.7280	0.8567	0.9946
0.1208	0	0.1213	0.2977	0	0.4479	0	0	0.6482	0.7374	0.8600	0.9976
0.1470	0	0.1671	0.3098	0	0	0	0	0	0.7506	0.8630	0.9992
0.1687	0	0.1774	0.3098	0	0	0	0	0	0	0.8632	1.0006
0.1949	0	0.1877	0	0	0	0	0	0	0	0.8633	1.0020
0.1953	0	0.1913	0	0	0	0	0	0	0	0	1.0030
		0.1915	0	0	0	0	0	0	0	0	1.0031
		0.1916	0	0	0	0	0	0	0	0	1.0033
		0.1929	0	0	0	0	0	0	0	0	1.0033
		0.1930	0	0	0	0	0	0	0	0	1.0033
		0	0	0	0	0	0	0	0	0	0

Figura 3.16: Ejemplo de cómo se reajustan los valores calculados con la técnica propuesta dentro de la matriz Ua . En la Matriz de la izquierda observamos dentro del recuadro rojo el valor del punto ($i = 8$) calculado durante la primera visita, mientras que en la Matriz de la derecha se observa el recuadro rojo, el que ahora pertenece al primer valor generado durante la segunda visita ($i = 126$) y se observa que éste obtuvo un valor más pequeño que la visita anterior.

Si por otro lado, se cumple con la condición $error(últimaVisita(i)) < -0.009$ entonces, el punto que se calculó en la última visita fue más chico que el punto original en esa visita ($punto(últimaVisita(i))$) y esto significa que la siguiente vez que se vuelva a visitar dicha *caja* el valor generado para el *punto* (i) debe de ser más grande. Tomando como base la ecuación (3.27) pero modificando el significado de las cotas, donde ahora $newci$ es igual a $punto(últimaVisita(i))$, esto es, el punto que se calculó en la última visita dentro de dicha *caja* y cambiando también la cota superior $newcs$ por $punto(últimaVisita(i + 1))$, es decir, el valor de la última visita en el instante $i + 1$ pues queremos ampliar esta cota para poder escoger un número más grande que el anterior. Con lo anterior, se cumple el objetivo de generar un valor más grande que el que se calculó en la visita anterior $punto(últimaVisita(i))$. Este algoritmo puede funcionar mejor mientras más visitada sea una *caja* pues va a ser capaz de corregir su precisión y aprender de la experiencia previa.

Todo el proceso de aprendizaje se resume en el diagrama de flujo de la Figura 3.17. Como podemos observar de la Figura 3.17, en el paso 1 se observa cómo se recorren los puntos de la ST, en el paso 2 se observa cómo se acumulan los puntos de la ST en un puntero. La validación del paso 3 corresponde a la condición de la definición 8, que recordemos es en donde se valida el punto $x(i)$ de la ST, para saber a qué *caja* pertenece y si cumple, entonces, se realiza el paso 4 en donde se pregunta si se está entrando por primera vez a una determinada *caja* y si cumple con este paso, entonces el siguiente (paso 5) calcula el valor del punto (i) de manera aleatoria dentro de los límites de la *caja*, así como su error correspondiente. Sin embargo,

cuando alguna determinada caja es visitada posteriormente se cumple con el paso 6, que es en donde ajustamos el valor de la visita anterior. En el paso 7, se valida si el error generado por el punto aleatorio satisface las condiciones de frontera de error que definimos previamente y si esto no se cumple, es decir, cumple con la condición del paso 7, entonces, se procede a recalcular las cotas del intervalo de donde se generará el nuevo valor i de acuerdo a la expresión (3.27) con lo cual se vuelve a generar el valor de i de manera aleatoria así como su error, como se observa en el paso 8, si por el contrario no cumpliera con la condición del paso 7, esto significa que el valor que se calculó fue muy acertado respecto del original y se conserva ese punto y lo que cambia ahora es el error entre ese punto y el original (paso 9).

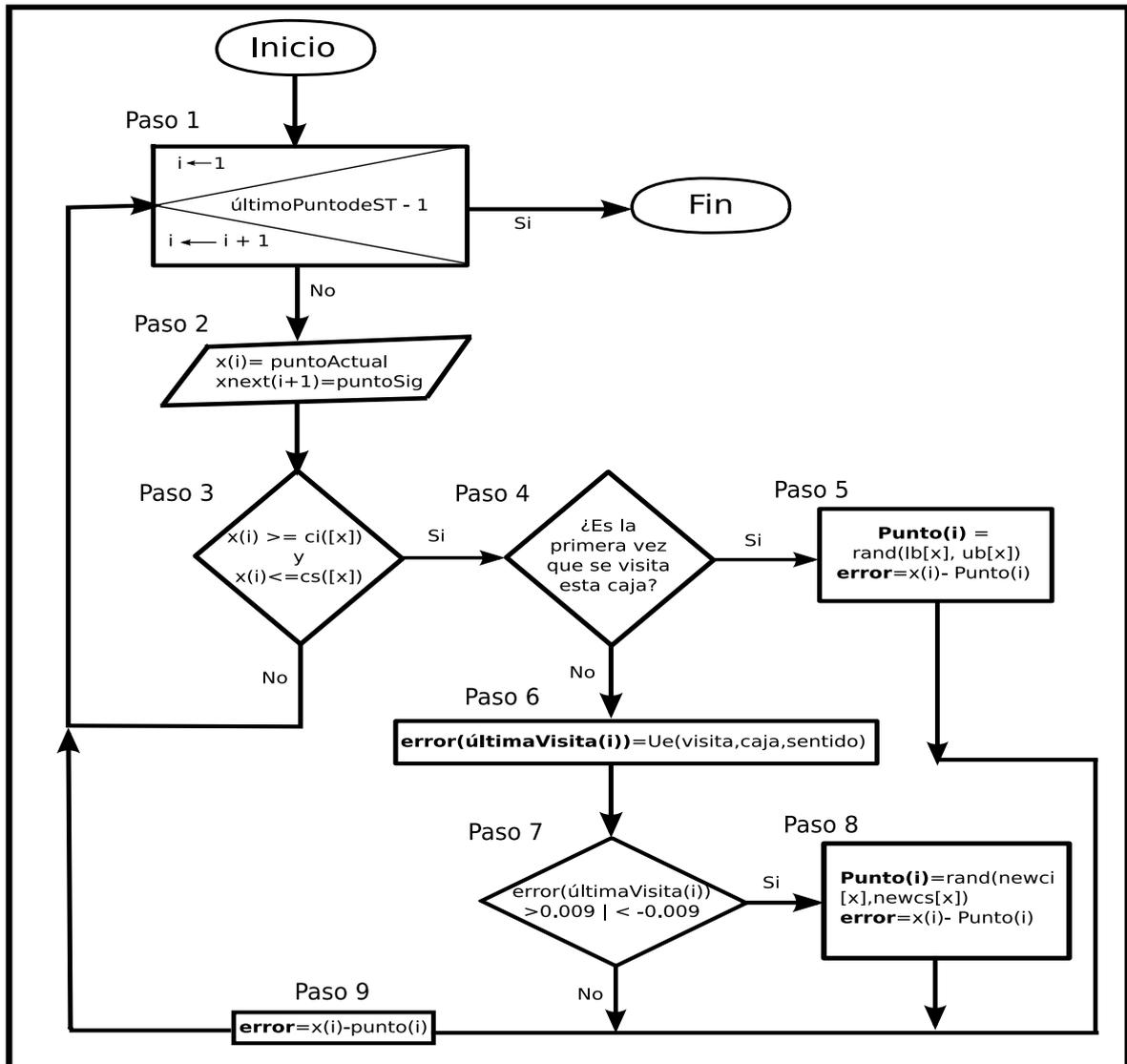


Figura 3.17: Diagrama de flujo del proceso de aprendizaje del comportamiento dinámico de series de tiempo.

Lo que se obtiene al final de todo el entrenamiento de la ST que se observó en la Figura 3.10a) son las matrices Ua y Ue que se observan en las Figuras 3.18 y 3.19.

a) Matriz de valores generados										
Ua(:, :, 0) =										
caja 1	caja 2	caja 3	caja 4	caja 5	caja 6	caja 7	caja 8	caja 9	caja 10	
0.1876	0.3016	0.3833	0.4431	0.5137	0.5646	0.6475	0.7218	0.8630	0.9628	
0.1766	0.2921	0.3822	0.4279	0.4991	0.5598	0.6346	0.7184	0.8444	0.9574	
0.1677	0.2842	0.3792	0.4235	0.4906	0.5535	0.6271	0.7065	0.8376	0.9444	
0.1416	0.2599	0.3745	0.4211	0.4803	0.5391	0.6266	0.6929	0.8282	0.9336	
0.1365	0.2585	0.3697	0.4178	0.4801	0.5385	0.5987	0.6902	0.8016	0.9304	
0.1255	0.2581	0.3502	0.4157	0.4799	0.5363	0.5881	0.6848	0.7920	0.9252	
0.1218	0.2486	0.3476	0.4139	0.4779	0.5354	0.5863	0.6841	0.7914	0.9207	
0.1181	0.2456	0.3445	0.4041	0.4748	0.5336	0.5846	0.6828	0.7903	0.9036	
0.1035	0.2453	0.3427	0.4024	0.4731	0.5317	0.5753	0.6729	0.7684	0.8891	
0.0856	0.2402	0.3272	0.4023	0.4624	0.5287	0.5753	0.6700	0.7636	0.8890	
0.0819	0.2341	0.3261	0.4001	0.4622	0.5231	0	0.6582	0.7610	0.8776	
0.0585	0.2335	0.3261	0.3994	0.4607	0	0	0.6569	0.7574	0.8719	
0.0448	0.2305	0.3244	0.3986	0.4595	0	0	0.6565	0	0.8650	
0.0313	0.2225	0.3233	0.3942	0.4569	0	0	0	0	0.8633	
0.0094	0.2026	0.3164	0.3903	0.4524	0	0	0	0	0	
0.0066	0.1987	0	0.3900	0.4520	0	0	0	0	0	
0	0	0	0	0.4500	0	0	0	0	0	
0	0	0	0	0.4495	0	0	0	0	0	

b) Matriz de Errores										
Ue(:, :, 0) =										
caja 1	caja 2	caja 3	caja 4	caja 5	caja 6	caja 7	caja 8	caja 9	caja 10	
-0.0044	-0.0054	-0.0047	-0.0029	0.0097	-0.0054	0.0025	-0.0132	0.0070	-0.0082	
-0.0004	0.0051	0.0052	-0.0151	0.0101	0.0118	0.0066	0.0004	-0.0116	-0.0076	
0.0027	0.0192	0.0162	-0.0115	0.0186	0.0265	0.0151	0.0065	-0.0124	-0.0126	
-0.0134	0.0149	0.0265	-0.0129	0.0173	0.0101	0.0396	0.0099	-0.0158	-0.0164	
-0.0105	0.0315	0.0397	-0.0142	0.0271	0.0175	-0.0133	0.0232	-0.0334	-0.0076	
-0.0135	0.0511	-0.0108	-0.0143	0.0169	-0.0077	-0.0139	-0.0152	-0.0310	-0.0028	
-0.0112	-0.0114	-0.0084	-0.0131	0.0279	-0.0086	-0.0017	-0.0069	-0.0196	0.0047	
-0.0099	-0.0094	-0.0105	-0.0179	0.0088	-0.0084	0.0056	-0.0022	0.0003	-0.0024	
-0.0175	-0.0087	-0.0063	-0.0166	0.0111	-0.0023	-0.0007	-0.0021	-0.0046	-0.0009	
-0.0344	-0.0078	-0.0178	-0.0137	0.0034	0.0097	-0.0037	0.0050	0.0096	0.0130	
-0.0381	-0.0129	-0.0159	-0.0109	0.0082	0.0041	0	-0.0288	-0.0190	-0.0024	
0.0273	-0.0135	-0.0119	-0.0106	-0.0103	0	0	-0.0191	-0.0066	-0.0041	
0.0172	-0.0145	-0.0116	-0.0114	-0.0045	0	0	-0.0065	0	-0.0080	
0.0049	-0.0245	-0.0117	-0.0108	0.0009	0	0	0	0	-0.0067	
-0.0126	-0.0324	-0.0146	-0.0097	-0.0256	0	0	0	0	0	
-0.0154	0.0007	0	-0.0060	-0.0180	0	0	0	0	0	
0	0	0	0	-0.0130	0	0	0	0	0	
0	0	0	0	-0.0055	0	0	0	0	0	

Figura 3.18: Resultado del aprendizaje de la ST cuando está decrentando, esto es, cuando $band = 0$. En a) se observa la matriz $Ua(:, :, 0)$ que se refiere a los puntos que fueron aprendidos y en b) se observa la matriz de errores $Ue(:, :, 0)$.

a) Matriz de valores generados									
Ua(:, :, 1) =									
caja 1	caja 2	caja 3	caja 4	caja 5	caja 6	caja 7	caja 8	caja 9	caja 10
0.0817	0.2332	0.3241	0.3941	0.4607	0.5295	0.5875	0.6650	0.7592	0.8709
0.0842	0.2431	0.3286	0.4090	0.4826	0.5373	0.5954	0.6813	0.7747	0.8802
0.0878	0.2496	0.3354	0.4172	0.4836	0.5471	0.6112	0.6997	0.7903	0.8896
0.0990	0.2594	0.3646	0.4183	0.4919	0.5477	0.6143	0.7067	0.8074	0.9109
0.1034	0.2718	0.3684	0.4266	0.4965	0.5480	0.6165	0.7070	0.8226	0.9248
0.1152	0.2877	0.3694	0.4270	0.4983	0.5523	0.6169	0.7078	0.8421	0.9346
0.1254	0.2896	0.3772	0.4301	0.5013	0.5558	0.6268	0.7142	0.8442	0.9402
0.1408	0.2907	0.3839	0.4320	0.5086	0.5649	0.6275	0.7186	0.8472	0.9494
0.1424	0.2914	0.3852	0.4327	0.5103	0.5669	0.6275	0.7198	0.8487	0.9530
0.1544	0.2949	0.3898	0.4346	0.5181	0.5698	0.6304	0.7224	0.8487	0.9531
0.1659	0.2967	0	0.4425	0.5184	0.5751	0.6353	0.7288	0.8494	0.9655
0.1781	0.3069	0	0.4474	0.5188	0	0.6400	0.7357	0.8521	0.9658
0.1866	0.3114	0	0.4491	0.5189	0	0.6488	0.7479	0.8550	0.9815
0.1907	0	0	0	0	0	0.6491	0.7507	0.8620	0.9993
0	0	0	0	0	0	0.6564	0	0	0
b) Matriz de Errores									
Ue(:, :, 2) =									
caja 1	caja 2	caja 3	caja 4	caja 5	caja 6	caja 7	caja 8	caja 9	caja 10
-0.0403	0.0242	0.0101	0.0031	-0.0123	0.0015	0.0085	-0.0060	-0.0008	-0.0081
-0.0418	0.0161	-0.0124	-0.0090	-0.0184	-0.0157	-0.0086	-0.0137	-0.0063	-0.0108
-0.0442	0.0026	-0.0296	-0.0298	-0.0084	-0.0189	-0.0158	-0.0153	-0.0137	-0.0174
-0.0400	-0.0086	0.0056	-0.0247	-0.0161	-0.0083	-0.0357	-0.0333	-0.0186	-0.0081
-0.0456	-0.0182	0.0014	0.0086	0.0085	-0.0220	-0.0395	-0.0280	-0.0264	-0.0062
-0.0438	0.0007	-0.0066	-0.0020	0.0023	-0.0067	-0.0161	-0.0442	-0.0209	-0.0074
-0.0506	-0.0114	-0.0118	-0.0109	0.0003	-0.0052	-0.0142	-0.0138	0.0152	-0.0138
-0.0502	-0.0143	-0.0011	0.0020	0.0006	0.0039	-0.0205	-0.0134	0.0122	-0.0106
-0.0446	-0.0056	0.0072	-0.0053	-0.0007	0.0059	-0.0285	-0.0142	0.0107	-0.0130
-0.0186	-0.0171	-0.0002	-0.0104	0.0031	0.0088	-0.0066	0.0084	0.0027	-0.0179
-0.0251	0.0237	0	-0.0025	0.0004	0.0031	-0.0107	0.0148	0.0014	-0.0075
0.0071	0.0179	0	0.0184	0.0008	0	-0.0150	0.0217	-0.0019	-0.0092
-0.0004	0.0044	0	0.0091	0.0009	0	-0.0012	0.0329	-0.0030	0.0215
-0.0043	0	0	0	0	0	-0.0009	0.0247	0.0020	0.0193

Figura 3.19: Resultado del aprendizaje de la ST cuando su bandera es igual a 1, es decir está incrementando. En a) se observa la matriz $Ua(:, :, 1)$ que almacena los valores generados y en b) se observa la matriz $Ue(:, :, 1)$ la cual almacena sus respectivos errores.

3.3.1. Cálculo del error de aprendizaje

El aprendizaje generado de la serie de tiempo, se almacenó en la matriz Ua para cuando sube ó cuando baja. Pero, con esta representación matricial no es posible observar de manera visual si en realidad la técnica está aprendiendo el comportamiento dinámico de la ST. Es por ello, que los valores que se generan a lo largo de toda la ST, se concatenan dentro del vector U y con esta representación de vector es posible observar cómo poco a poco se aprende el comportamiento de alguna ST. En otras palabras, cuando graficamos el vector U , éste puede ser visto como una ST que aprende el comportamiento de una determinada función. El vector U obtenido del aprendizaje de la ST “Seno” se graficó en la Figura 3.20.

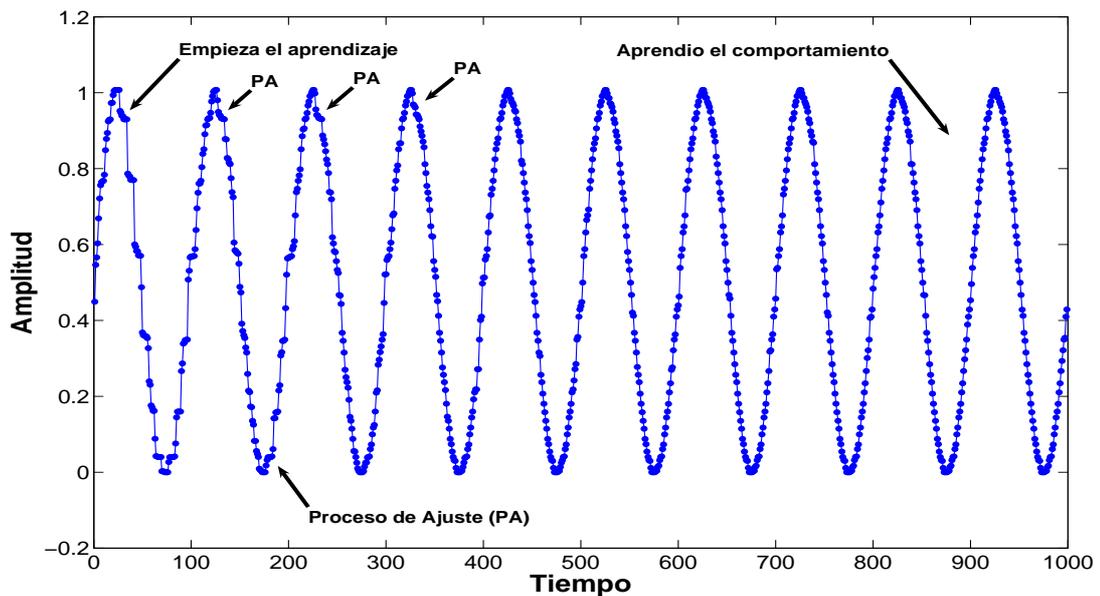


Figura 3.20: Ejemplo del proceso de aprendizaje del comportamiento de la serie de tiempo “Seno”.

El error que mide la diferencia entre la ST que generamos con la “Técnica de Aprendizaje Basado en Imagen” respecto a la ST original lo llamamos *error de aprendizaje* y lo calculamos con la métrica conocida como raíz cuadrada del error cuadrático medio (RMSE, por sus siglas en inglés), la cual se calcula de la siguiente manera:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_o - x_a)^2}{n}} \quad (3.28)$$

donde x_o se refiere a los puntos de la ST original, x_a corresponde a los puntos que van aprendiendo el comportamiento de la serie de tiempo original y n significa el número total de ambas secuencias de puntos (vector de puntos originales y el vector generado U).

Este error de aprendizaje se obtiene de manera global para todo el aprendizaje de la ST,

por ejemplo, el error de la serie “Seno” que se generó en la Figura 3.20 fue de 0.000701. El cual indica que la ST que generamos aprendió el comportamiento de la serie de tiempo original con una precisión muy buena.

3.4. Comparación con otras técnicas de aprendizaje

Como hemos comentado, uno de los objetivos de la tesis es encontrar una nueva manera de representación de los datos para el aprendizaje del comportamiento dinámico de series de tiempo y esto nos llevó a proponer una nueva técnica de aprendizaje.

Para ver la diferencia entre la representación de los datos y de las técnicas que se encuentran en la literatura, haremos un breve resumen de las técnicas de redes neuronales, máquinas de soporte vectorial y de la técnica que proponemos en esta tesis. Es importante aclarar que la comparación que a continuación realizaremos es únicamente respecto a la metodología y no a nivel de resultados, debido a que en la literatura no se han reportado resultados para el proceso de aprendizaje, pero sí de predicción.

Redes neuronales. Como se comentó en la sección 2.2.1 los puntos que representan el vector de entrada de las redes neuronales se obtiene de la siguiente manera:

1. Primero, se define el número de puntos que conformarán el vector de entrada.
2. Se define el retardo adecuado entre estos puntos.
3. Posteriormente, se obtiene el vector de entrada de los puntos de la ST correspondiente a los puntos con algún retardo previamente establecido.

Una vez que se obtiene este vector de entrada, entonces los siguientes pasos serán:

1. Escoger una arquitectura adecuada para cada ST, esto es, definir el número óptimo de capas ocultas y neuronas adentro de cada capa.
2. La interconexión entre los nodos (feed-back, nodos completamente conectados, entre otros).
3. Definir el tipo de función de transferencia en capas interconectadas y de salida.
4. Determinar el número de nodos de salida.
5. Definir el tipo de entrenamiento o algoritmo de entrenamiento.

Supongamos que ya se definió todo lo anterior, entonces tendríamos un esquema de aprendizaje más o menos parecido al de la Figura 3.21.

Como se puede observar antes de realizar el proceso de aprendizaje primero hay que tomar decisiones que pudieran afectar gravemente este proceso, es decir, si no se escoge el número óptimo de capas ocultas o el número de neuronas dentro de cada capa, cambiaría por completo el resultado del proceso de aprendizaje.

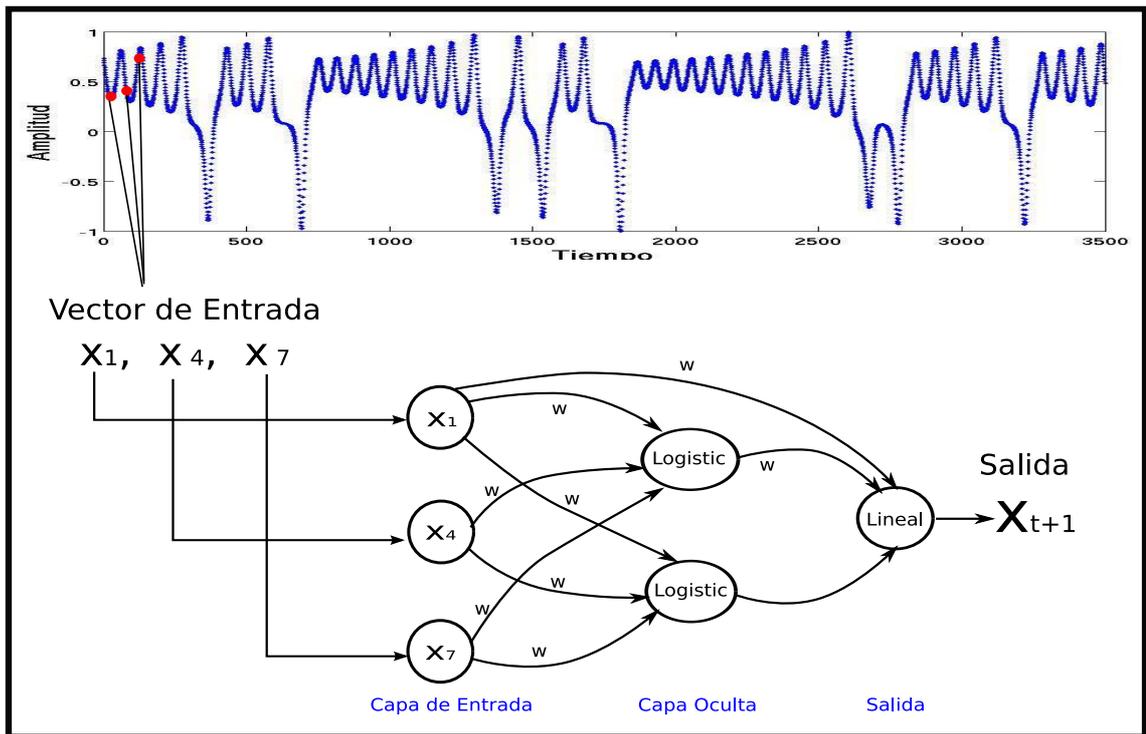


Figura 3.21: Esquema de cómo funcionan las redes neuronales y de cómo se trabajan los puntos de la serie de tiempo durante su proceso de aprendizaje. Además, se observa el vector de entrada correspondiente a dicha red neuronal.

Máquinas de soporte vectorial. Una técnica que funciona de manera diferente a las redes neuronales es la técnica de máquinas de soporte vectorial (SVM, Support Vector Machine), cuyo vector de entrada está representado por una ventana en el tiempo que se va deslizando a través de un retardo entonces lo que se necesita definir para que esta técnica funcione adecuadamente es lo siguiente:

- Definir el número consecutivo de puntos que entran en una ventana $w(x_i, \dots, x_{i+p-1})$.
- Escoger una función kernel adecuada para nuestros datos. Existen funciones kernel lineales y no lineales.

Un diagrama que explica cómo funcionan las máquinas de soporte vectorial y su representación de los datos se puede observar en la Figura 3.22.

Es posible observar, que la técnica de soporte vectorial tiene menos decisiones que tomar antes de empezar el proceso de aprendizaje que las redes neuronales, pero aún así una mala elección en la definición de la función “kernel” y ésta afectaría el rendimiento de esta técnica.

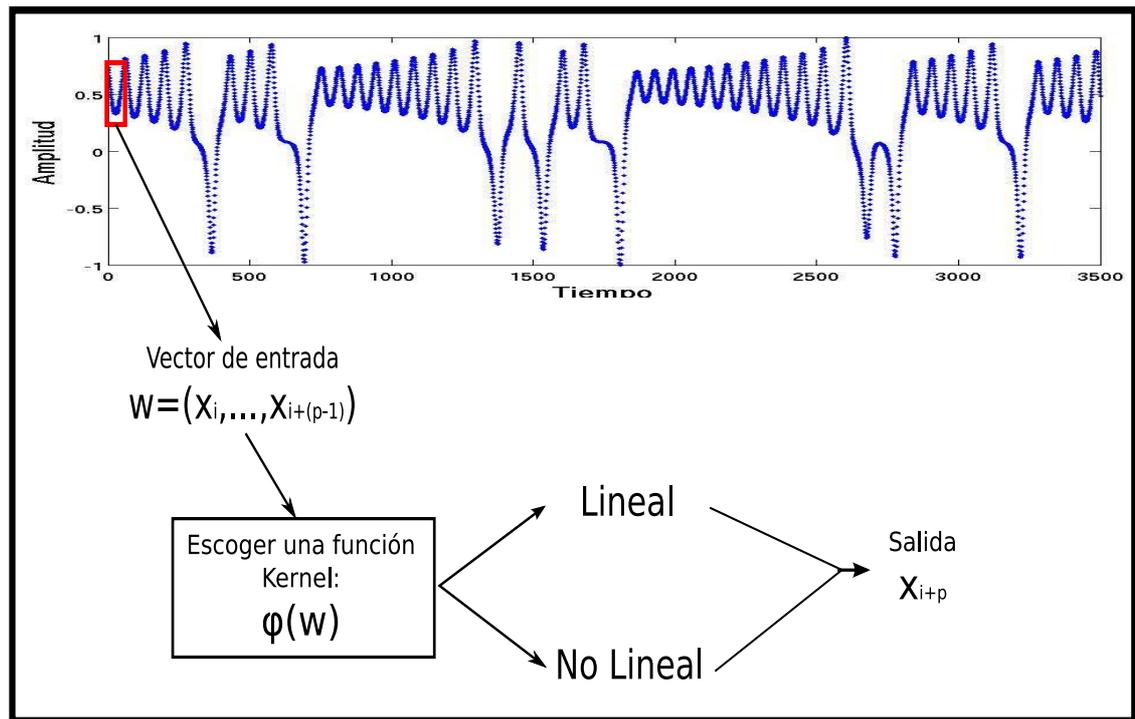


Figura 3.22: Esquema explicativo de cómo funcionan las máquinas de soporte vectorial y cómo se obtiene el vector de entrada de los datos de la serie de tiempo para su aprendizaje.

Técnica de aprendizaje TABI. Una vez, que hemos recordado cómo se realiza el proceso de aprendizaje de las técnicas: redes neuronales y máquinas de soporte vectorial, procederemos a compararlas con la técnica que estamos proponiendo. Lo primero que debemos notar es que la representación de los datos para aprender los puntos de la ST es nueva con respecto a las encontradas en la literatura, lo cual representa una contribución importante para la literatura.

Primero, la representación de los datos a entrenar de la ST, es a través de ventanas en la imagen de ésta y lo que se debe de decidir es lo siguiente:

- talvez la única posible decisión es el número óptimo de cajas, que está dado por el número de puntos y éste es único para cada ST, como se explicó en la sección 3.3. Además, que se ofrece un método de definir el número de cajas óptimas de manera automática.

El esquema de TABI y la representación de los datos de la ST para su aprendizaje se observa en la Figura 3.23.

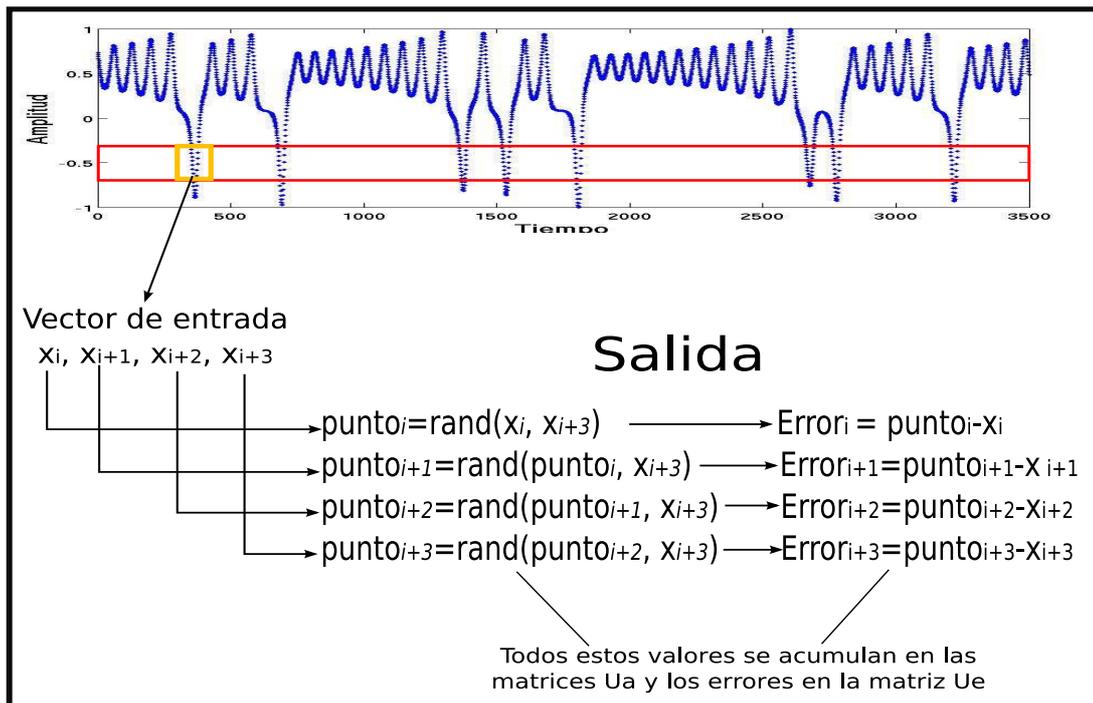


Figura 3.23: Esquema de TABI y también el vector de entrada de los puntos de la serie de tiempo en ventanas de su imagen para su aprendizaje.

Como observamos con la técnica que proponemos no hay que tomar decisiones, lo que la hace menos compleja que la técnica de redes neuronales y que la técnica de máquinas de soporte vectorial. Lamentablemente en la literatura no hay experimentos del aprendizaje de las técnicas de redes neuronales ni de máquinas de soporte vectorial, pero sí hay de la predicción de series de tiempo con las técnicas antes mencionadas, es por ello, que el siguiente capítulo estará dedicado a la predicción de series de tiempo. De esta manera podemos comparar el rendimiento de cada una de estas técnicas con la que proponemos en este trabajo.

Capítulo 4

Aplicaciones de la técnica

Como mencionamos en capítulos anteriores, primero definimos una nueva manera de representar los datos de entrada de la técnica de aprendizaje de ST a analizar. Esta representación es diferente a las reportadas en la literatura. Usando esta nueva representación de datos, generamos una nueva técnica de aprendizaje del comportamiento dinámico de ST estacionarias. Esta nueva técnica se basa en el principio conocido como *submetas* (subgoals) [5], el cual dice que es más fácil dividir un problema difícil en varios problemas fáciles. Esto es, por una parte, todos los subproblemas tienen un objetivo en particular (aprender la dinámica de los puntos dentro de alguna caja) pero por otro lado, todos los subproblemas siguen un objetivo global (aprender toda la dinámica de alguna ST). Como vimos en la sección 3.2.2, la técnica que se propone para aprender el comportamiento dinámico de series de tiempo, se basa en la información de su imagen y ésta encuentra nueva información que las técnicas clásicas de aprendizaje y que algunas de las técnicas modernas.

En este capítulo, emplearemos un mecanismo de indexación para recuperar la información obtenida durante el proceso de aprendizaje, también emplearemos técnicas probabilísticas con la finalidad de tomar decisiones sobre los futuros puntos de la ST. Estas decisiones se realizarán de acuerdo a la información aprendida y de esta manera generar una primera aproximación hacia el problema de predicción.

4.1. Predicción de series de tiempo

La predicción ha sido una de las aplicaciones más utilizadas en el área del aprendizaje, especialmente en el análisis de series de tiempo debido a las múltiples aplicaciones que esto conlleva en áreas tales como: economía, control, medicina, entre otras. La predicción de ST la podemos definir de la siguiente manera:

Definición 13 *La predicción de una ST $x(t_1), x(t_2), \dots, x(t_n)$, se define como el proceso de estimar un valor $x(t_{n+1})$ o varios valores futuros $x(t_{n+h})$.*

Recordemos que el tipo de predicción que se va a manejar en este trabajo es la *predicción iterada*, la cual, predice un punto adelante, dicho punto predicho, posteriormente será utilizado para predecir el siguiente punto y así sucesivamente hasta tener h predicciones.

4.1.1. Información almacenada para la predicción

Durante el proceso de aprendizaje de la técnica TABI se almacena información de manera individual para cada *caja* dentro de las matrices Ua y Ue y estas matrices contienen la información de los puntos de la ST que se va aprendiendo durante el proceso de entrenamiento. También durante este proceso, se observa que los puntos de la ST que pertenecen a una determinada *caja*, son los siguientes:

$$punto(t_1), punto(t_2), punto(t_3), \dots, punto(t_l) \quad (4.1)$$

donde l es el número de puntos que pertenecen a una determinada *caja*, y a éstos l puntos se les llamará *longitud* si cumplen con las siguientes condiciones (ver Figura 4.1):

1. Todos los puntos deben pertenecer a una determinada *caja*.
2. Dichos puntos deben tener el mismo sentido (ascendente ó descendente).
3. Dependiendo del sentido de estos puntos se deberá cumplir al menos con alguna de las siguientes condiciones:
 - a) sí estos puntos tienen un sentido ascendente ($band = 1$) se deberá cumplir con la siguiente condición:

$$punto(t_1) < punto(t_2) < punto(t_3) < \dots < punto(t_l) \quad (4.2)$$

- b) por otro lado, si estos puntos tienen un sentido descendente ($band = 0$) tendrán que cumplir con la siguiente condición:

$$punto(t_1) > punto(t_2) > punto(t_3) > \dots > punto(t_l) \quad (4.3)$$

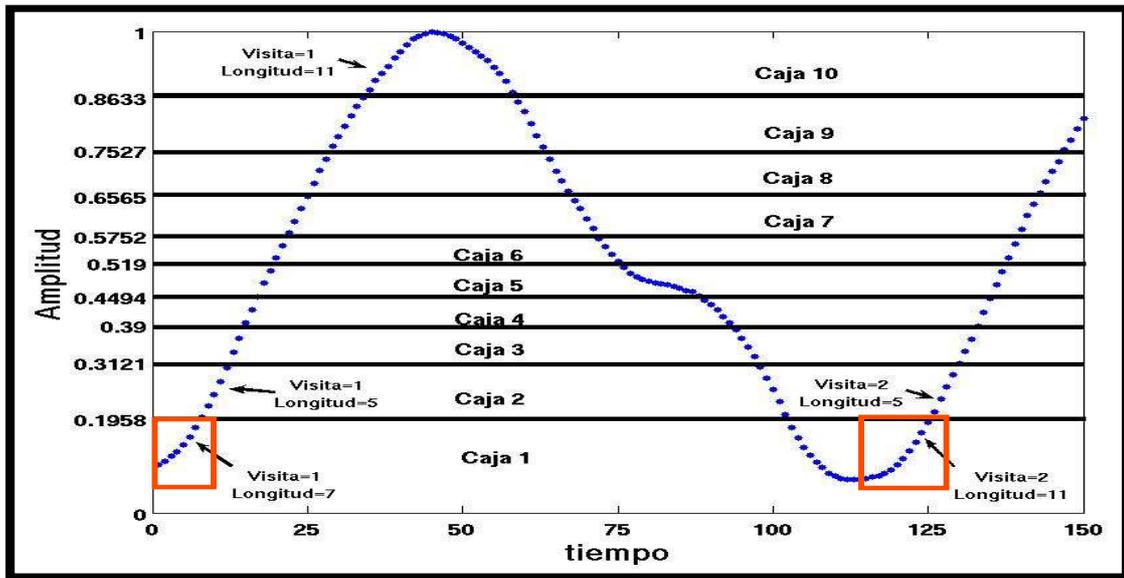


Figura 4.1: Ejemplo de cómo se obtiene la longitud de los puntos (recuadro naranja) de la serie de tiempo durante el proceso de aprendizaje.

Por ejemplo, supongamos que se desea obtener la *longitud* de los puntos que visitaron por primera vez la *caja 1* de la Figura 4.1, entonces, se observa que la *longitud* de los puntos que visita esta *caja* es igual a 7 (la visita de éstos puntos a esta *caja*=1 empieza en el instante $t = 1$ y terminan en el instante $t = 7$). También se observa que en el instante $t = 8$ los puntos están visitando por primera vez la *caja 2* y termina en el instante $t = 12$ teniendo una *longitud* = 5 y así sucesivamente. De tal manera que podemos tener información de la *longitud* de puntos que permanecieron en una determinada *caja* por un cierto tiempo durante alguna visita.

La información de estas *longitudes* por cada *caja* se almacena en una matriz llamada *longitud* la cual cumple con la siguiente definición:

Definición 14 Sea *longitud* una matriz definida como $longitud := (longitud_{i,j,k}) \quad \forall 1 \leq i \leq m, 1 \leq j \leq n \text{ y } 1 \leq k \leq o$, donde **i** representa el número de la visita, **j** representa la caja en la que los puntos realizaron la visita y **k** representa el sentido de los puntos cuando ingresaron a alguna caja durante su visita.

De acuerdo a la definición anterior, la posición $longitud(1, 1, 1)$ significa que es la primera visita ($i = 1$) que realizan los puntos dentro de la *caja 1* ($j = 1$), cuando el sentido de los puntos de la ST es ascendente, esto es, $band = 1$ ($k = 1$) y dentro de la posición $longitud(1, 1, 1)$ se acumula la longitud de puntos que cumplieron con las condiciones anteriores que para el ejemplo $longitud(1, 1, 1) = 7$. Lo anterior se observa mejor en la Figura 4.2.

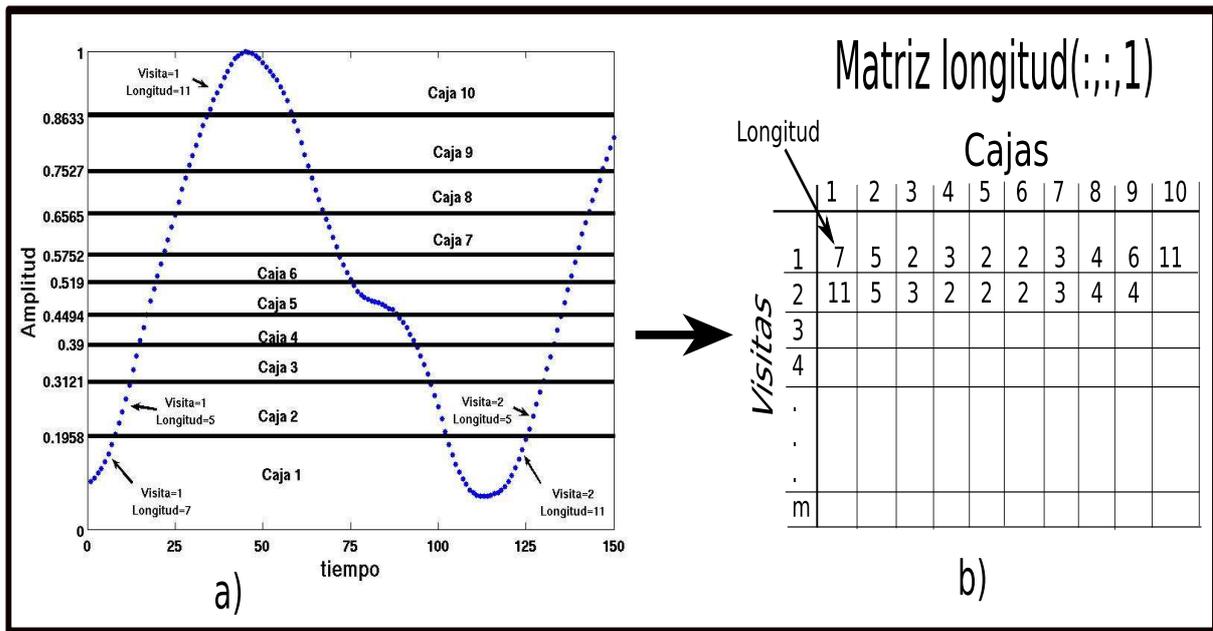


Figura 4.2: Esta figura muestra cómo se genera la matriz *longitud*. En a) se observa la ST dividida en 10 *cajas*, además se observan las visitas y las longitudes durante cada visita y en b) se observa el llenado de la matriz *longitud* cuando los puntos de la ST están subiendo, esto es, con *band* = 1.

Además, de la longitud de los puntos, durante el proceso de aprendizaje, también se almacena información sobre los saltos que se hacen de una *caja* hacia otra *caja*, por ejemplo, si se observa nuevamente la Figura 4.1, en el instante $t = 7$ el punto pertenece a la *caja* 1 y el siguiente punto en el instante $t = 8$ ahora pertenece a la *caja* 2, lo que significa que hay un salto de la *caja* 1 a la *caja* 2 y esta información se va almacenando en una matriz llamada “saltos”.

Esta matriz de “saltos” cumple con la definición 14 pero, la diferencia radica en que el valor que se almacena dentro de la matriz es el número de la *caja* a la que salta. Por lo tanto, esto significa que si algún punto durante la visita i en el instante t el cual pertenece a la *caja* j mientras que los puntos tienen el sentido igual a k y si el siguiente punto, es decir, con $t + 1$, brinco a la *caja* $j + 1$, significa que $\text{saltos}(i, j, k) = j + 1$.

Por ejemplo, si observamos la Figura 4.3, es posible afirmar que $\text{saltos}(1, 1, 1) = 2$, lo que significa que los puntos que estaban en la *caja* 1 ($j = 1$), durante la primera visita ($i = 1$), y con un sentido ascendente ($k = 1$), brincaron a la siguiente *caja* 2. Un ejemplo de cómo se va almacenando esta información de los brincos dentro de la matriz de saltos también se observa en la Figura 4.3.

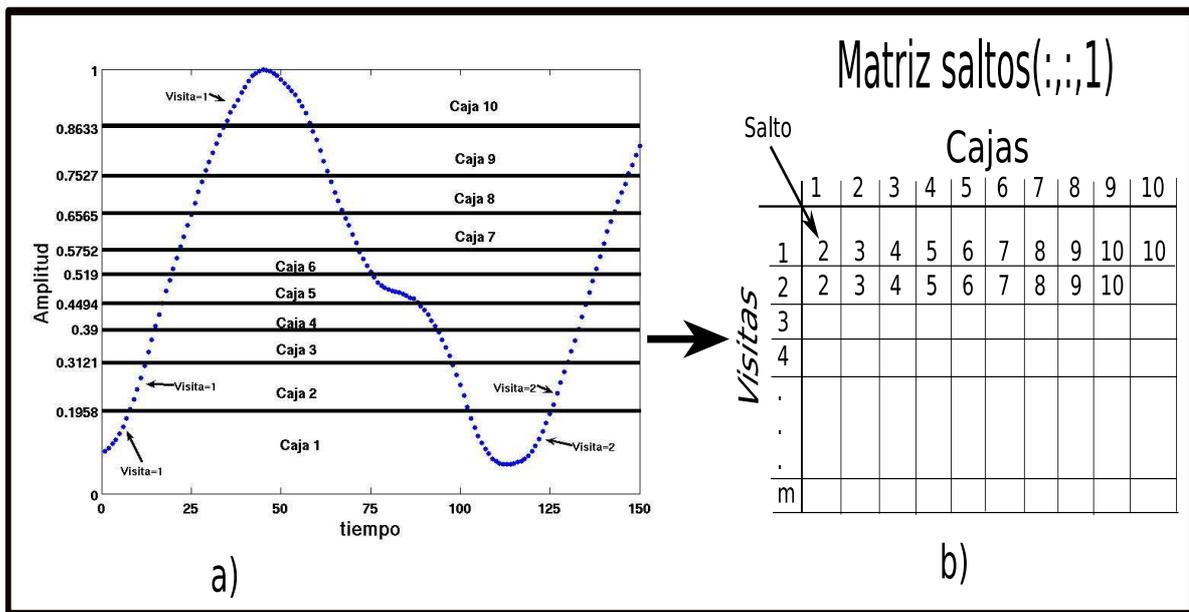


Figura 4.3: Se muestra cómo se genera la matriz *saltos*. En a) se observan los puntos de la ST y las *cajas* que éstos van visitando y en b) se observa como se llena la matriz *saltos* cuando los puntos de la ST tienen $band = 1$.

Con las matrices *longitud* y *saltos* obtenidas durante el proceso de aprendizaje, es posible observar que los puntos dentro de cada *caja* cuentan con la siguiente información:

- El tiempo en el cual ciertos puntos permanecen dentro de una cierta *caja* (*longitud*).
- También se almacena información de la siguiente *caja* a la cual brincan los puntos después de haber visitado la *caja* actual (*saltos*).

Si se analiza la información obtenida de la matriz *longitud* (ver Figura 4.2) y la matriz *saltos* (ver Figura 4.3) se puede observar que es posible unir las matrices *longitud* y *saltos* en una sola matriz que contenga la información de ambas y aumentar esta información. Su definición sería la siguiente:

Definición 15 Sea *longSaltos* una matriz definida como $longSaltos := (longSaltos_{l,j,k,i}) \forall 1 \leq l \leq m, 1 \leq j \leq n, 1 \leq k \leq o, 1 \leq i \leq p$ donde m representa la longitud de los puntos por cada visita, n representa la caja a la que va a saltar después de estar en una determinada caja (*salto*), o se refiere a la caja actual y p corresponde al sentido de los puntos. El valor que se almacena en la matriz $longSaltos(l, j, k, i)$ es la frecuencia de los puntos que se obtienen de la unión de las matrices *longitud* y *saltos*, en otras palabras, $longSaltos(l, j, k, i)$ significa la frecuencia en que se repitieron ciertos eventos¹.

Un ejemplo que muestra como es la unión de las matrices *longitud* y *saltos* se observa en la Figura 4.4.

¹Un evento se refiere a que los puntos que se encontraban en la caja k brincaron a la caja j con una longitud de l mientras tenían el sentido i .

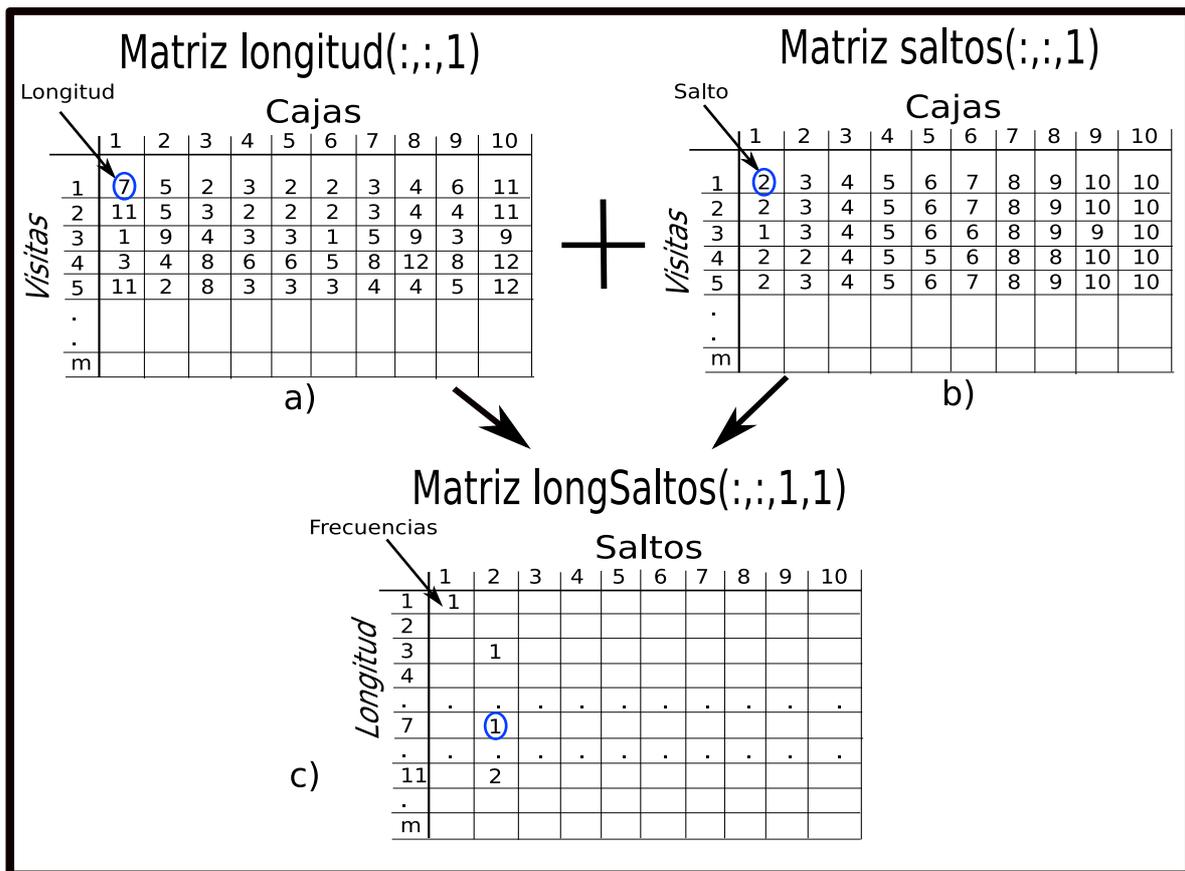


Figura 4.4: Fusión de las matrices *longitud* y *saltos*. En a) se observa la matriz *longitud*, en b) se observa la matriz *saltos* y en c) se observa la fusión de a) y b). Se observa que se encerró dentro de un círculo azul un ejemplo de cómo se construyó la nueva matriz *longSaltos* a partir de las matrices a y b, donde $longSaltos(7, 2, 1, 1) = 1$.

Por ejemplo, $longSaltos(7, 2, 1, 1) = 1$ significa que hasta ese momento del aprendizaje, una vez se repitió la siguiente situación:

- los puntos que tenían una longitud de 7 puntos ($l = 7$) brincaron hacia la *caja* 2 ($j = 2$) cuando anteriormente había visitado la *caja* 1 ($k = 1$) mientras que el sentido de estos puntos era ascendente ($i = 1$), ver Figura 4.4.

La información que se obtiene de la matriz *longSaltos* se puede considerar como la experiencia adquirida durante el proceso de aprendizaje, que para fines de predicción se le conoce como *entrenamiento*. Esta experiencia nos indica el número de veces que los puntos de la ST estuvieron en una determinada *caja* y luego después de un cierto tiempo estos puntos se cambiaron hacia otra *caja*. Además, también indica el sentido de los puntos cuando ocurrió esta transición. Un ejemplo de cómo sería la experiencia acumulada al final del aprendizaje de los puntos de la ST en la *caja* 5 mientras estos tenían el sentido ascendente se observa en la matriz $longSaltos(:, :, 5, 1)$, y éste sería muy similar a la que se observa en la Figura 4.5.

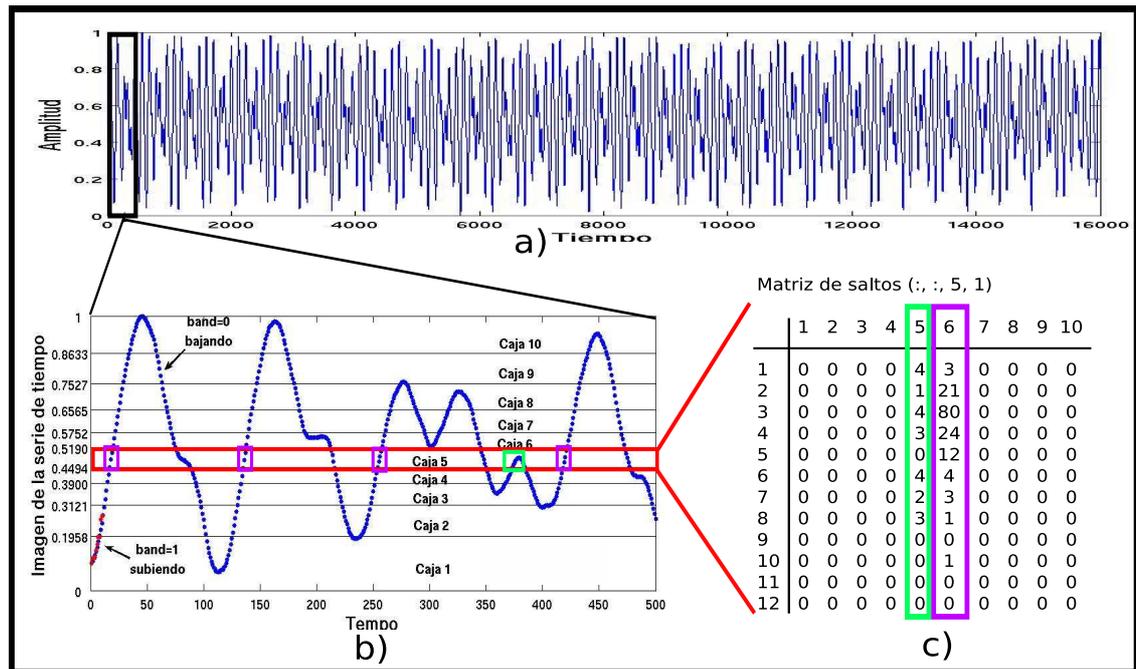


Figura 4.5: Obtención de la matriz *longSaltos* después de terminar el proceso de aprendizaje (entrenamiento). En a) se observa la serie de tiempo original, en b) se observan los primeros 500 puntos de la serie de tiempo, también se observan dos rectángulos uno de color morado el cual indica la situación de los puntos que estando dentro de la caja 5 brincan a la caja 6, y la caja verde indica la situación en donde apesar de estar en la misma caja 5 los siguientes puntos brincan a la misma caja 5 pero cambian su sentido, entonces, en c) se observa el resultado de la matriz *longSaltos* y se enfatizan las dos situaciones anteriores mencionadas con los colores respectivos.

Como se observa en la Figura 4.5, dentro de la matriz *longSaltos* existen algunas ocasiones en las que estando en la misma *caja* los puntos tienen dos posibles caminos para escoger en un futuro ya sea que éstos salten a la siguiente caja o bien los siguientes puntos permanezcan en la misma caja pero cambian el sentido de los puntos y es debido a lo anterior que el sistema que implementamos deben de tomar una decisión. Una primera aproximación para esta toma de decisiones es darle pesos a las frecuencias, teniendo así mayor peso aquellas decisiones que hayan sido las más repetidas durante el proceso de aprendizaje, es decir, las que tengan mayor frecuencia.

Sí tomamos en cuenta la definición de la matriz *longSaltos*, se tiene la siguiente información de los últimos puntos aprendidos durante el entrenamiento:

- Para cada caja se tiene información sobre las frecuencias de los brincos y longitudes que se repiten a lo largo del entrenamiento (información local). En otras palabras cada caja almacena las condiciones bajo las cuales cada una es visitada (su experiencia).
- Se cuenta con el último valor que se aprendió de la ST original *punto* (t_n).

- La última *caja* que este punto visitó.
- El sentido de los últimos puntos.
- Así como la última longitud de puntos durante esta última visita.

Esta información va a ser de vital importancia al momento de tomar una decisión para estimar el siguiente, esto es, la siguiente *caja* hacia la que van a brincar los futuros puntos y de esta manera realizar la predicción de los h puntos de la ST, pues indica de alguna manera los patrones que se estuvieron almacenando para cada *caja*. Este proceso de predicción y la toma de decisiones se verá a continuación.

4.1.2. Pasos para realizar la predicción

En nuestro caso, el proceso de predicción consiste en tomar una decisión sobre la siguiente *caja* a la que deben de brincar los puntos futuros de la serie de tiempo, así como estimar el número de puntos dentro de esta determinada *caja* y finalmente el sentido que tendrán dichos puntos estimados. Al juntar estos tres resultados se obtiene la predicción de h puntos de la ST. Este proceso se realiza tomando en cuenta la información anterior y la experiencia acumulada durante el proceso de aprendizaje.

Con la información anterior es posible determinar cuáles son los valores que se tienen que estimar para poder predecir los puntos de una ST. Tomando en cuenta el principio de submetas es posible estimar para cada *caja*, la siguiente longitud de puntos, así como también estimar la siguiente *caja* a la que estos puntos deben de saltar, esto es, cada *caja* de acuerdo a su experiencia acumulada puede determinar la siguiente longitud y la siguiente *caja* a la que deben de brincar estos puntos y con el resultado secuencial de cada *caja* es posible predecir la ST.

Al final del entrenamiento se cuenta con la información de la última *caja* visitada, la última longitud y el sentido que llevaban los últimos puntos de la ST. Entonces, el único valor que no tenemos, es la *caja* a la que van a brincar los siguientes puntos en un futuro, siendo así el valor que se debe de estimar primero.

Para estimar la siguiente *caja* a la que deben de brincar los futuros puntos se extrae información de la matriz *longSaltos*, es decir, con la experiencia acumulada obtenida durante el aprendizaje, se recupera la información de la última longitud de los puntos (l), la última *caja* que visitan (k) y el último sentido de dichos puntos (i), en otras palabras, se extrae la matriz *longSaltos* ($l, :, k, i$), pues el valor que no se sabe y se desea estimar es la siguiente *caja* a la que va a brincar estos puntos, esto es el valor de j . Supongamos, por ejemplo, que del resultado obtenido en el entrenamiento de los más de 16,000 puntos de la ST “Qperiodic2” se obtiene la siguiente información (ver Figura 4.5a)):

1. Última *caja* = 10
2. Última longitud = 9
3. Última bandera = 1

Con la información anterior, se obtiene la matriz *longSaltos* ($(:, :, 10, 1)$) que es la que se observa en la Figura 4.6. Recordemos que además, se tiene la información de la longitud de

los últimos puntos del proceso de aprendizaje, que se había dicho era de 9 puntos el cual si se observa la Figura 4.6 está encerrado en un rectángulo azul, este recuadro azul indica que durante el proceso de aprendizaje los puntos de la ST que entraban a la *caja* 10 mientras que cumplieran con las condiciones de que estos puntos de la ST tuvieran un sentido ascendente y tuvieran una longitud de 9 puntos consecutivos, entonces, la siguiente *caja* que visitarían sería la *caja* 10.

		matriz $longSaltos(:, :, 10, 1)$										
		Cajas										
Longitudes		1	2	3	4	5	6	7	8	9	10	
	1	0	0	0	0	0	0	0	0	0	1	5
	2	0	0	0	0	0	0	0	0	0	2	1
	3	0	0	0	0	0	0	0	0	0	0	2
	4	0	0	0	0	0	0	0	0	0	0	4
	5	0	0	0	0	0	0	0	0	0	0	2
	6	0	0	0	0	0	0	0	0	0	0	4
	7	0	0	0	0	0	0	0	0	0	0	6
	8	0	0	0	0	0	0	0	0	0	0	5
	9	0	0	0	0	0	0	0	0	0	0	16
	10	0	0	0	0	0	0	0	0	0	0	20
	11	0	0	0	0	0	0	0	0	0	0	16
	12	0	0	0	0	0	0	0	0	0	0	12
	13	0	0	0	0	0	0	0	0	0	0	3
	14	0	0	0	0	0	0	0	0	0	0	1

Figura 4.6: Resultado de buscar en la matriz $longSaltos$ la frecuencia de los puntos en la *caja* 10 y con el sentido de los puntos ascendente. Dentro de la matriz $longSaltos(:, :, 10, 1)$ se observan las frecuencias de los puntos con longitud igual a 9 puntos, la cual se encuentra encerrada dentro de un rectángulo azul y éste es igual a 16 cuando brinca a la siguiente *caja* 10.

El procedimiento para estimar la siguiente *caja* a la que deben de brincar los siguientes puntos es el siguiente:

1. Extraer la matriz $longSaltos(:, :, k, i)$, en donde, como se mencionó anteriormente se conoce la última *caja* visitada (k) y el último sentido de los puntos (i).
2. Extraer de esta matriz el vector correspondiente a la última longitud de los puntos (l) al cual llamaremos $extract$, en otras palabras $extract = longSaltos(l, :, k, i)$.

Así, una vez que se extrajo la matriz $longSaltos(:, :, k, i)$ se procede a extraer el vector $extract$ de esta matriz ejecutando el código de la Figura 4.7, en donde las variables de entrada son la última *caja* visitada ($lastBox$), la última longitud ($lastLong$), el último sentido de los puntos ($lastBand$) y la matriz $longSaltos$.

```

1  function extract=getVector(lastBox, lastLong, lastband, longSaltos)
2  [long, nextBox, actualBox, band]=size(longSalto);
3  for i=1:1:band
4      for k=1:1:actualBox
5          for l=1:1:long
6              if k==lastBox && l==lastLong && i==(lastband+1)
7                  extract=longSaltos(l,:,k,i);
8              end
9          end
10     end
11 end

```

Figura 4.7: Código para extraer el vector *extract* de la matriz *longSaltos* ($l, :, k, i$).

Como se observa en la Figura 4.7, en las líneas 3, 4 y 5 se recorre la matriz *longSaltos* hasta encontrar la fila de dicha matriz que cumpla con las condiciones de la última *caja*, longitud y bandera (línea 6) y si cumple, posteriormente se procede a extraer el vector *extract* en la línea 7.

Con la información que se extrajo de la matriz *longSaltos* (9, :, 10, 1) es posible saber hacia que *caja* o *cajas* brincó después de haber tenido ciertas condiciones en el pasado. Esto es, como comentamos anteriormente en la matriz *longSaltos* se van almacenando las rutas o caminos que fueron tomando los puntos de la ST durante el proceso de aprendizaje y esto se ve reflejado en la matriz *longSaltos*.

Continuando con el ejemplo anterior, el siguiente paso es estimar la siguiente *caja* a la que los futuros puntos deben de brincar de acuerdo a la información con la que se cuenta y si revisamos esta información, tenemos el siguiente vector (ver Figura 4.6):

$$extract = [0, 0, 0, 0, 0, 0, 0, 0, 0, 16], \quad (4.4)$$

lo que significa que durante el proceso de aprendizaje la *caja* 10 obtuvo una frecuencia igual a 16 cuando su bandera fue igual a 1 y teniendo una longitud de 9, lo que significa que los siguientes puntos brincaron a la *caja* 10. Además, hubo un cambio de bandera a 0. Este cambio de bandera se puede deber a tres cosas:

1. Que los puntos de la ST hayan alcanzado un valor mínimo ó un valor máximo dentro de la *caja* 1 o 10 respectivamente, esto es, si la ST está normalizada dentro del intervalo $[0, 1]$ el valor mínimo que pueden alcanzar los puntos es de 0 y un máximo de 1, pero si está normalizada dentro del intervalo $[-1, 1]$, sólo el valor mínimo cambia a -1 y el valor máximo se conserva.
2. O simplemente que el comportamiento de los puntos cuando visitaron una *caja* brincaron hacia esta misma *caja* y entonces, el sentido de los puntos se cambia.
3. Una tercera razón, es que el sentido de los puntos que pertenecen a una determinada *caja* (supongamos la *caja* 9) con un mismo sentido (supongamos ascendente) y además supongamos que los siguientes puntos cambien de manera brusca su sentido hacia alguna *caja* distinta de la que estaban los puntos anteriores (supongamos éstos brincaron ahora

a la caja 7). Lo anterior significa que los siguientes puntos brincaron hacia alguna caja anterior. Lo anterior también se aplica cuando el sentido de los puntos es descendente y cambian de manera brusca su sentido hacia alguna caja distinta de la última visitada con sentido contrario.

Un ejemplo de los cambios del sentido de los puntos durante el entrenamiento se observa en la Figura 4.8.

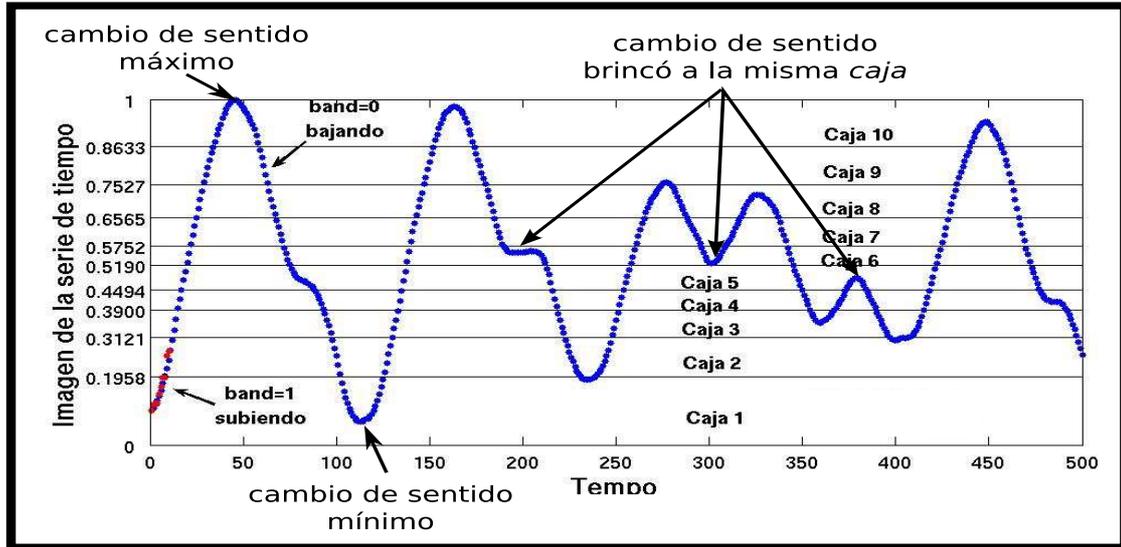


Figura 4.8: Se muestran tres posibles situaciones en las que el sentido de los puntos se cambia.

Entonces, continuando con el ejemplo anterior, el vector que se extrajo de la matriz *longSaltos* indica que los futuros puntos deben de brincar a la misma *caja* 10, y esto significa que cambiaron su sentido, es decir, ahora los puntos bajarán. En el ejemplo anterior la solución es trivial ya que no hay mucha información para tomar decisiones.

Pero, en el caso de que hayan diferentes frecuencias dentro del vector “*extract*”, se debe proceder de la siguiente forma. Supongamos que la última información de los puntos de la ST fue la siguiente:

1. Última *caja* = 10
2. Última longitud = 1
3. Última bandera = 1

Por lo tanto, en ese caso, si observamos la matriz *longSalto*(:, :, 10, 1) del ejemplo de la Figura 4.6, se extrae la fila correspondiente a la longitud de puntos igual a 1 correspondiente al siguiente vector,

$$extract = [0, 0, 0, 0, 0, 0, 0, 0, 1, 5], \quad (4.5)$$

en el cual se puede observar que, de acuerdo a las frecuencias de sus experiencias pasadas, existen dos posibilidades; que brinque a la *caja* 9 o bien que brinque a la misma *caja* 10 en

cualquiera de los dos casos el sentido de los puntos cambia debido a que, como se comentó anteriormente, en el primer caso se brincó a una caja menor y en el segundo caso significa que se ha alcanzado una *caja* máxima. Por lo tanto, existen dos posibles caminos y se tiene que tomar una decisión sobre qué camino se debe de tomar. Esto se puede determinar de la siguiente manera:

- Primero, las frecuencias nos van a indicar los pesos que tiene cada uno de los posibles caminos, por ejemplo la *caja* 10 tiene más peso que la *caja* 9.
- Segundo, los pesos pueden ser vistos como probabilidades de ocurrir, es decir, la *caja* 10 tiene más probabilidades de volverse a repetir en un futuro que la *caja* 9.
- Finalmente, de manera aleatoria, pero ponderada, se escoge cuál de las dos cajas es la que se visitará en el futuro.

Un ejemplo de cómo sucede lo anterior sería, cómo escoger al azar una pelota dentro de una bolsa en donde se tienen 5 pelotas azules y 1 pelota roja, donde, la probabilidad de escoger una pelota azul es mayor que de escoger una pelota roja. Por lo que es lógico pensar que en el futuro se va a repetir que escogamos una pelota azul, pero dentro de la bolsa existen pelotas de distintos colores, a pesar de que las azules tengan más peso que digamos la(s) blanca(s) también existe una posibilidad de escoger la opción que tiene menos peso, esto es, existe la posibilidad de escoger una pelota blanca.

En otras palabras, lo que necesitamos hacer es generar una tabla que contenga información de las futuras cajas junto con sus frecuencias y sus probabilidades de ocurrir y esto se genera con el código de la Figura 4.9, en la cual se observa la función “getTable”, cuyo argumento de entrada es el vector *extract*:

```

1  function table=getTable(extract)
2  contF=1;
3  tam=size(extract);
4  colF=tam(2);
5  for k=1:1:colF
6      if extract(1, k) > 0
7          table(contF,1)=k;
8          table(contF,2)=extract(1,k)
9          contF=contF+1;
10     end
11 end
12 sumFrec=sum(table(:,2));
13 sizeT=size(table);
14 f=sizeT(1);
15 for l=1:1:f
16     frec=table(l,2);
17     table(l,3) = frec/sumFrec;
18     if l>1
19         table(l,4)=table(l-1,4)+table(l,3);
20     else
21         table(l,4)=table(l,3);
22     end
23 end

```

Figura 4.9: Código para generar la distribución de probabilidades de cada futura *caja*.

En las líneas 3 y 4 del código anterior se almacena el tamaño del vector “extract” en donde el número de las columnas (ColF) del vector *extract* indica la(s) *caja(s)* en la(s) cuál(es) los

puntos del pasado brincaron. En la condición de la línea 6 se recorre todo el vector *extract* y sí existe un valor mayor que cero dentro de una determinada columna (*caja*) significa que ciertos puntos brincaron a esa *caja* y entonces, se acumula el valor de la columna (línea 7) en la matriz *table* (*ColF*, 1), el cual nos indica que hay una *caja* que tiene una cierta frecuencia de volver a ocurrir en un futuro y este valor de frecuencia también se almacena en la matriz *table* (*ColF*, 2) (línea 8). Debido a que es necesario calcular las probabilidades de las futuras *cajas* a las que se puede brincar, el siguiente paso (línea 12) es sumar todas las frecuencias que se acumularon en la segunda columna de la matriz *table*. Después, se procederá a calcular las probabilidades para cada fila de la matriz *table*, es decir, la probabilidad de ciertas *cajas*, esto se logra de la siguiente manera (línea 17):

$$prob = \frac{frec}{sumfrec}, \quad (4.6)$$

lo anterior se va acumulando en la tercera columna de la matriz *table* y finalmente existe una cuarta columna dentro de la matriz *table* la cual acumula la suma de las probabilidades generando ciertos intervalos. Un ejemplo de cómo se forma la matriz *table* a partir del vector que se extrajo de la matriz *longSalto* (1, :, 10, 1) se observa en la Figura 4.10.

siguiente Caja	Frecuencia	Probabilidad	Suma de Probabilidades
9	1	1/6=0.1667	0.1667
10	5	5/6=0.833	1
	suma=6		

Figura 4.10: Un ejemplo de cómo se genera la matriz *table*. Se observa que en la primera columna se almacena la(s) caja(s) a la(s) que posiblemente brinquen los puntos, en la columna dos se almacena la frecuencia de la(s) caja(s), en la columna tres se acumula la probabilidad de ocurrir de cada caja y finalmente en la cuarta columna se almacena la suma de probabilidades de la(s) caja(s) visitada(s) por los puntos durante el proceso de aprendizaje.

Finalmente, se escoge de manera aleatoria pero con ciertos pesos el valor de la siguiente *caja* a la cual los futuros puntos deben de brincar y esto se hace con el código que se observa en la Figura 4.11. Como se observa en esta figura, en la línea 1 los parámetros de entrada de la función “getBox” son la matriz *table* y la última bandera (sentido de los puntos), luego se genera un valor aleatorio entre 0 y 1 como se observa en la línea 2. Después es necesario comparar el valor que se calculó en la línea 2 con la última columna de la matriz *table* la cual si recordamos nos arroja la suma de probabilidades para cada caja, entonces en la línea 6 se busca la *caja* a la que pertenece el valor que se calculó de manera aleatoria y de acuerdo al resultado que obtengamos, será la futura *caja* hacia la que vamos a brincar. Por ejemplo, si el valor azar de la línea 2 es igual a 0.555, significa que está dentro de la probabilidad de ocurrir de la *caja* 10 y ésta sería la *caja* que se estimaría con este procedimiento y se almacenaría en *nextBox*.

```

1 function nextBox=getBox(table, lastband)
2 azar=rand();
3 tam=sizeof(table)
4 f=tam(1)
5 for m=2:1:f
6     if azar>table(m-1,4) && azar<=table(m,4)
7         nextBox=table(m,1);
8         if lastInterv==choseBox
9             if lastband==0
10                nextband=1;
11            else
12                nextband=0;
13            end
14        else
15            nextband=lastband;
16        end
17    end
18 end

```

Figura 4.11: Código para escoger de manera aleatoria ponderada el valor de la siguiente *caja* a la que los siguientes puntos van a brincar.

Cuando ya se estimó hacia qué *caja* se deben de mover los futuros puntos, el siguiente paso será estimar la siguiente longitud de los puntos que van a permanecer en esta futura *caja*. Esto es muy similar a la estimación anterior pero siguiendo estos pasos:

- De la matriz *longSalto* se extrae la información de la frecuencia de puntos con sentido descendente en donde además los puntos brincarón hacia la *caja* 10, esto es, extraemos la matriz correspondiente a *longSaltos* (:, 10, :, 0).
- Como se desea estimar la nueva longitud de los futuros puntos se suman las columnas de la matriz *longSaltos* (:, *j*, :, *i*), la cual contiene información sobre los puntos que estando dentro de la *caja* *j* con el sentido *i* tuvieron ciertas longitudes *l* en el pasado.
- Una vez que se tiene la suma de las columnas dentro de un vector llamado *suma*, también se procede a calcular una matriz *table*, en donde, en la primera columna se almacena el valor de las futuras longitudes, en la segunda columna se almacena el valor de la frecuencia de estas longitudes, hay una tercera columna que almacena la probabilidad de ocurrir de las longitudes y finalmente en la cuarta columna se almacena la suma de probabilidades de cada futura longitud.
- Finalmente y tomando en cuenta la matriz calculada *table* la cual tiene las probabilidades de ocurrir de cada longitud se escoge la siguiente longitud de puntos de manera aleatoria y ésta se almacena en *nextLong*.

Un ejemplo de la metodología que se siguió para decidir cuál era la siguiente longitud se observa en la Figura 4.12.

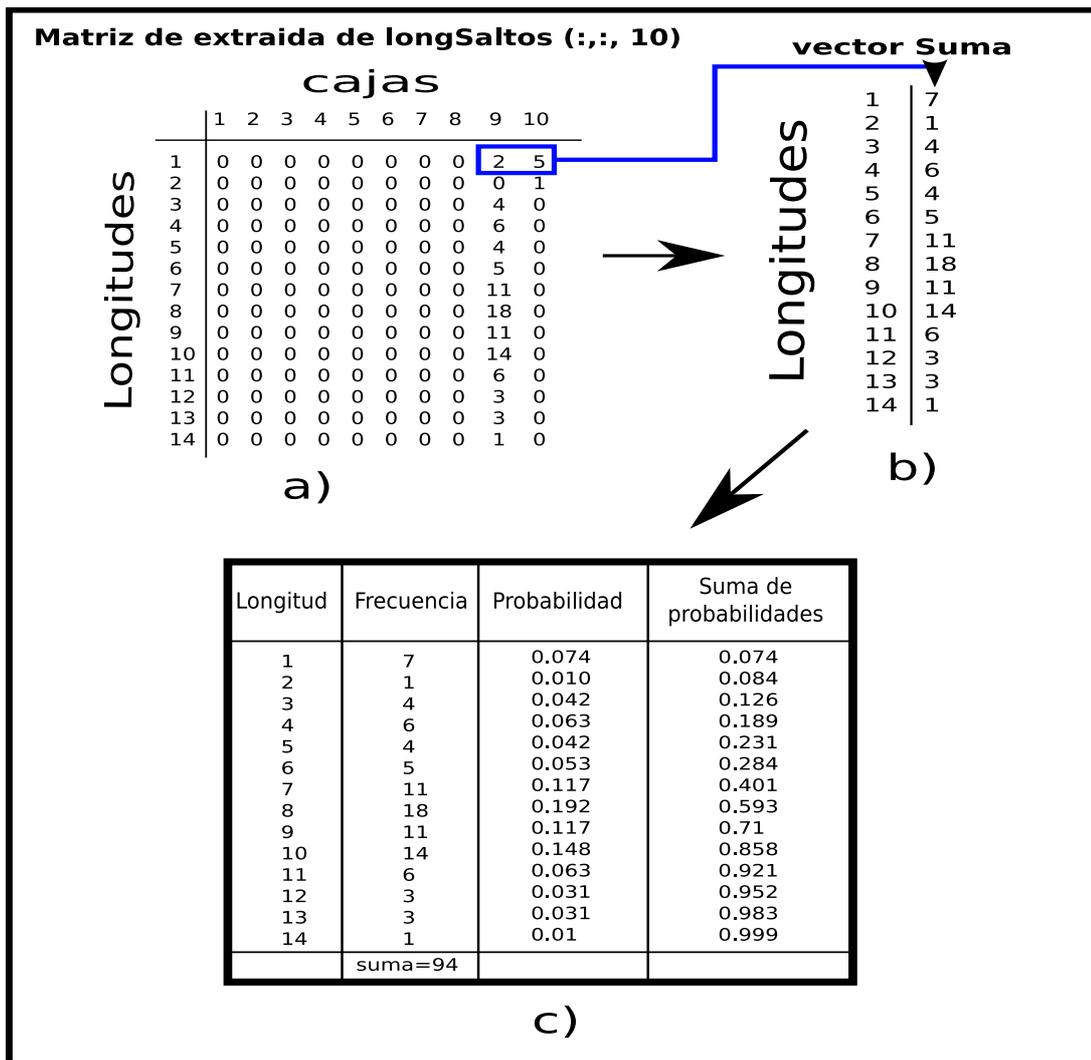


Figura 4.12: Pasos que se siguieron para elegir la longitud de los siguientes puntos. Se observa que en a) se tiene la matriz que se extrae de *longSaltos*, b) muestra cómo se genera el vector *suma* y c) muestra la tabla de probabilidades para cada longitud pasada.

Una vez que se estimó la siguiente *caja* j , bandera i y longitud l , es posible entonces predecir los siguientes h puntos de la ST. Estos h puntos se buscan dentro de la matriz Ua , que se llenó durante el proceso de aprendizaje, es decir, se busca $Ua(nextLong, nextBox, nextBand)$ y entonces, se predicen los h puntos.

En resumen, los pasos que se deben de seguir para el proceso de predicción son los siguientes:

1. Obtener la última información del aprendizaje:
 - a) la última *caja* que visitan los últimos puntos (k)
 - b) la última longitud de los puntos (l)
 - c) el último sentido de los puntos de la ST (i).

2. De acuerdo con lo anterior, la información que no se tiene es la siguiente *caja* a la que los futuros puntos brincarán, por lo que es necesario estimar esta siguiente *caja* (*nextBox*) y de ser necesario se actualiza el sentido de los puntos.

3. Cuando ya se estimó la siguiente *caja*, la información que se tiene es la siguiente:
 - a) la nueva *caja* estimada

 - b) el nuevo sentido de los puntos (si es que se modificó)

 - c) pero no se cuenta con la siguiente longitud de los futuros puntos. Esta longitud es la que se estima en este paso.

4. Hasta aquí se cuenta con la nueva *caja* estimada, el nuevo sentido de los puntos y la nueva longitud de éstos. En este paso se junta esta información y se busca en la matriz que se guardó del proceso de aprendizaje Ua (*nextLong*, *nextBox*, *nextBand*) lo que dará como resultado la predicción de h puntos de la ST.

5. Una vez que se predicen estos h puntos se regresa al paso 2 y se repite esta iteración hasta alcanzar el número total de puntos que se desean predecir. En la Figura 4.13 se observa un esquema explicativo de los pasos anteriores.

Cuando finalmente se estimaron los puntos futuros de la ST, entonces, decimos que estamos haciendo predicción de series de tiempo. Existen varias métricas que determinan el rendimiento de las técnicas de predicción de series de tiempo y en este trabajo se utilizó la métrica conocida como la raíz cuadrada del error cuadrático medio (RMSE, por sus siglas en ingles), el cual se calcula de la siguiente manera:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i^p - x_i^o)^2}{n}} \quad (4.7)$$

donde x_i^p representa el i -ésimo punto predicho con la técnica de predicción propuesta, x_i^o se refiere al i -ésimo punto de la ST original y n se refiere el número total de puntos predichos de la ST.

Esta métrica de error, RMSE, representa una medida típica del error de predicción [31], siendo particularmente sensible a las discrepancias grandes entre los valores original y predicho de la ST.

En el siguiente capítulo, aplicaremos esta técnica para la predicción de series de tiempo utilizadas en la literatura especializada y compararemos los resultados experimentales con los resultados obtenidos de las técnicas de redes neuronales y máquinas de soporte vectorial.

Capítulo 5

Análisis de resultados

En el capítulo anterior, empleamos un mecanismo de indexación para recuperar la información obtenida durante el proceso de aprendizaje, también empleamos técnicas probabilísticas con la finalidad de tomar decisiones sobre los futuros pasos de los puntos de la ST. Dichas decisiones se basan en la información aprendida y de esta manera acercarnos al problema de predicción. Ahora aplicaremos esta técnica a diferentes ST para validar su eficiencia.

La primera parte de este capítulo describe el conjunto de ST utilizadas en el análisis experimental que se realizó en este trabajo. Dicho análisis tiene el fin de probar la eficiencia de la nueva técnica de aprendizaje propuesta, este conjunto de ST es una muestra representativa utilizada en el análisis de series de tiempo y como vimos anteriormente durante el desarrollo de la técnica de aprendizaje se presentan tres tipos de dividir distintos para dividir la imagen de la ST y debido a lo anterior, en este capítulo se presentarán los resultados del aprendizaje para cada uno de estos tipos de división. Recordemos que la primera división se refiere a la división de la imagen de acuerdo a la anchura de las cajas, el segundo tipo de división emplea una división de la imagen dinámica a través del número de puntos empleando mecanismos frecuentistas y finalmente, el tipo tres se refiere a dividir la imagen de la ST de acuerdo a algún porcentaje del total de puntos de la ST, sin importar sus frecuencias.

En la segunda parte de este capítulo se evalúa la técnica de predicción que proponemos en este trabajo, probando su rendimiento. Dicha validación se lleva a cabo usando un conjunto de ST que por sus características dinámicas, son series de estructuras muy variadas, desde las más simples como las de tipo periódico, hasta las más complejas como las correspondientes a fenómenos naturales. Este conjunto de ST se utiliza para probar las técnicas de predicción reportadas en la literatura, por lo que, representan una muestra significativa debido a que este conjunto de ST llevan al límite a las técnicas de predicción.

Finalmente, en este capítulo también se muestra que, a partir de la información que se obtiene con la técnica de aprendizaje propuesta, obtuvimos mejores resultados en el problema de predicción de series de tiempo, comparado con los resultados obtenidos del análisis de estas mismas ST usando las técnicas de redes neuronales y máquinas de soporte vectorial.

5.1. Descripción del conjunto experimental

Para probar la eficiencia de las nuevas técnicas TABI y PBI propuestas en este trabajo se seleccionó un conjunto de ST con el siguiente criterio de selección:

- Las ST deben ser una muestra representativa utilizada en el análisis y evaluación de técnicas predicción de ST utilizadas en la literatura especializada [30][31].
- También, deben ser representativas de una clasificación básica de referencia, en este caso, en base a su comportamiento dinámico (periódico, cuasi periódico, caótico, complejo y estocástico, definidos en la sección 2.1.3).

A continuación, se describen las ST utilizadas en este trabajo, así como la descripción de su naturaleza y su comportamiento dinámico. También se puede observar en las Figuras (5.1 - 5.29) el comportamiento dinámico correspondiente a cada una de las ST analizadas en este trabajo. En estas figuras únicamente se graficaron los primeros 1000 puntos de las series de tiempo, esto con el fin de obtener una mejor visualización de su comportamiento dinámico:

Seno. Serie periódica generada para 10 ciclos (ver Figura 5.1), es decir, esta serie tiene 1,000 puntos y ésta se origina con la siguiente función:

$$f(x) = \text{sen}(x). \quad (5.1)$$

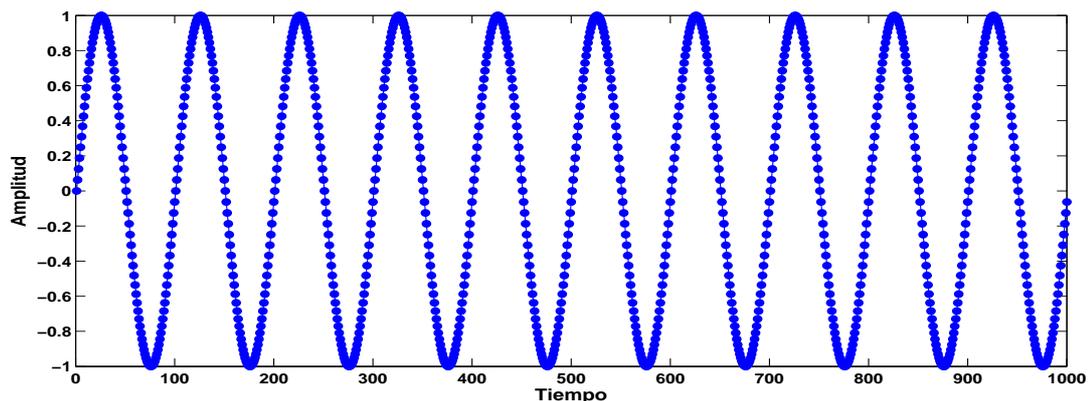


Figura 5.1: Serie de tiempo “Seno”.

Vanderpol. Serie periódica generada por una ecuación diferencial, la cual representa el modelo del paso de la carga eléctrica (denotado por y) a través de un circuito oscilador de un tubo de vacío [46]. Esta ST contiene un total de 2,000 puntos y en la Figura 5.2 se pueden observar los primeros mil puntos. La ecuación diferencial que representa al fenómeno es la siguiente:

$$\frac{d^2y}{dt^2} + (y^2 - n) \frac{dy}{dt} + w^2y = 0 \quad (5.2)$$

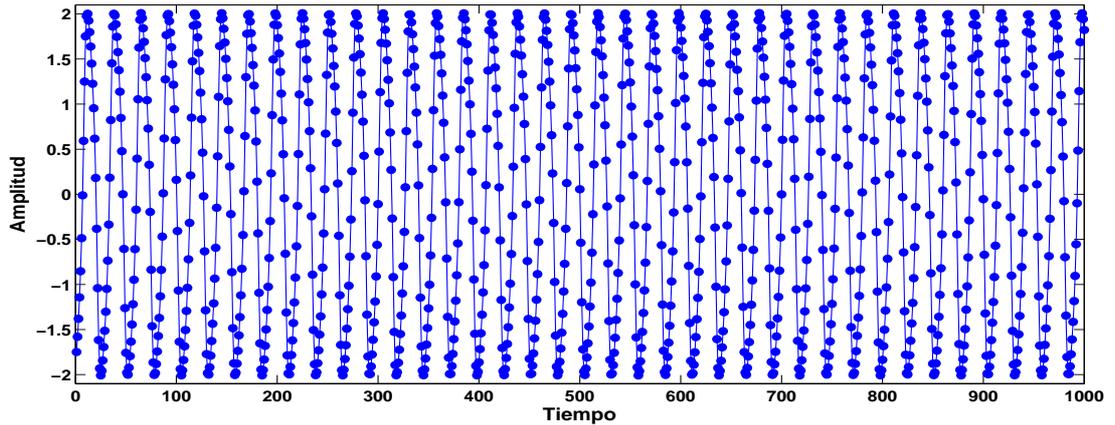


Figura 5.2: Serie de tiempo “Vanderpol”.

Qperiodic2. Serie cuasi periódica obtenida de la medición de una variable de velocidad (cm/s) en un experimento anular para reproducir un Flujo de Couette (Reología), las mediciones se obtuvieron de un muestreo de cada 0.1 *segundos* [31][30] y esto generó un total de 16,384, de los cuales únicamente se observan los primeros 1,000 puntos en la Figura 5.3.

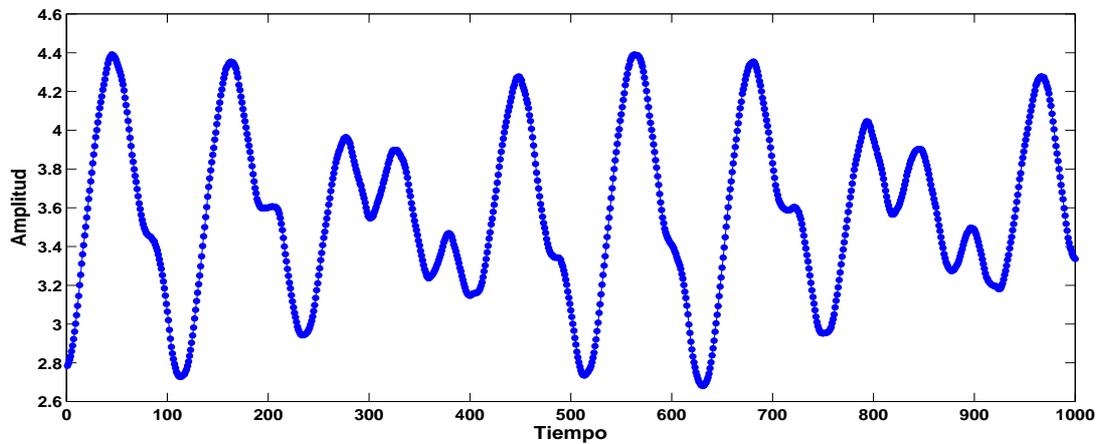


Figura 5.3: Serie de tiempo “Qperiodic2”.

Qperiodic3. Serie cuasi periódica obtenida de forma análoga a la serie Qperiodic2, con la diferencia de que ésta fue muestreada cada 0.4 *segundos*, y esto generó un total de 16,384, de los cuales únicamente se observan los primeros 1,000 puntos en la Figura 5.4.

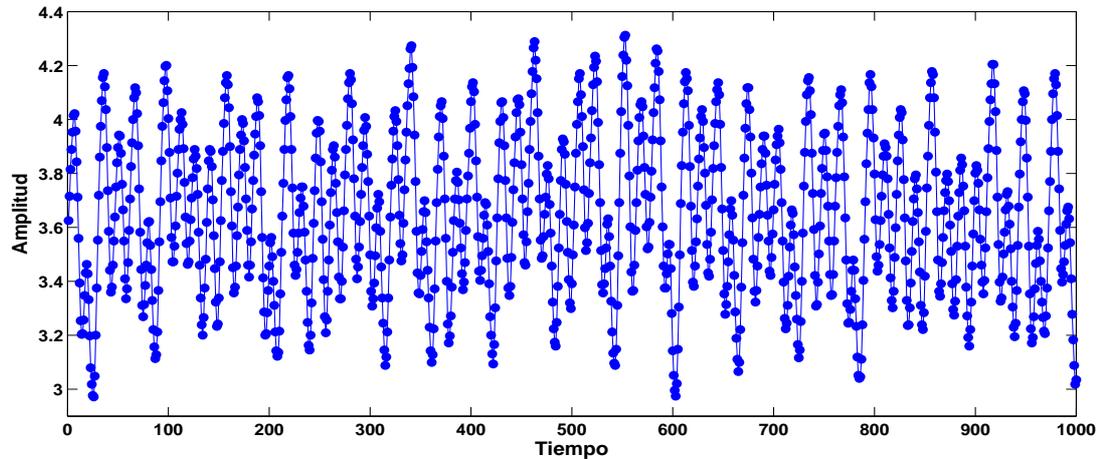


Figura 5.4: Serie de tiempo “Qperiodic3”.

Mackey-Glass. Serie caótica generada por una ecuación diferencial de retardo temporal: Modelo de formación de células sanguíneas blancas (linfocitos) [47][48]. La ecuación para obtener esta serie es de la forma:

$$\frac{dx}{dt} = -bx(t) + \frac{ax(t-\tau)}{1 + [x(t-\tau)]^{10}} \quad (5.3)$$

donde $a = 0.2$, $b = 0.1$ y $\tau = 3000$. Esta ST contiene un total de 2,000 puntos y únicamente se muestra la mitad de éstos en la Figura 5.5

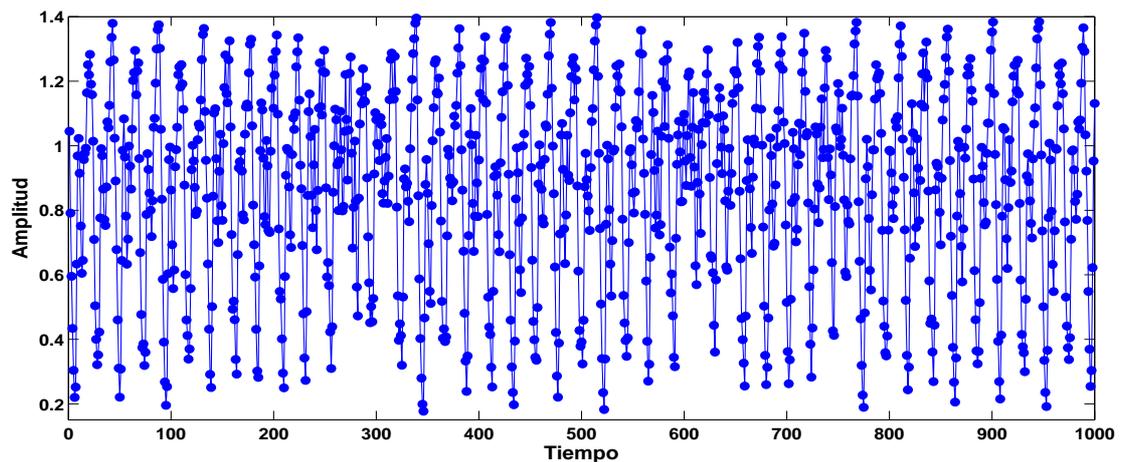


Figura 5.5: Serie de tiempo “Mackey & Glass”.

Logistic. Serie caótica generada por un mapa interpretado como un modelo ecológico de las variaciones anuales de poblaciones de insectos [46], esta ST cuenta con un total de 1,000 puntos como se observa en la Figura 5.6 y su expresión matemática es la siguiente:

$$x_{n+1} = rx_n(1 - x_n) \quad (5.4)$$

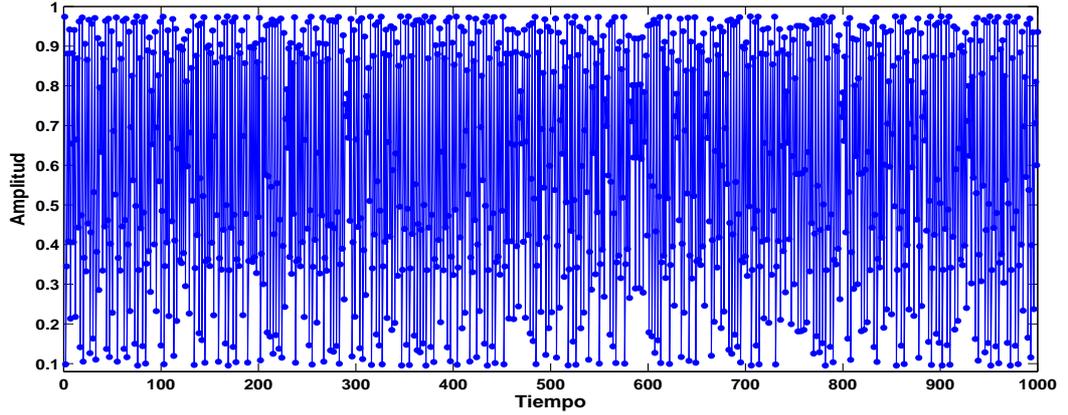


Figura 5.6: Serie de tiempo “Logistic”.

Lorenz. Serie caótica generada por un sistema de ecuaciones diferenciales y se refiere a un modelo de convección de fluidos (convección de Rayleigh-Benard), la cual se presenta en la atmósfera terrestre [46][49]. El sistema de ecuaciones que la representa es de la forma siguiente:

$$\begin{aligned} \frac{dx}{dt} &= -\tilde{\sigma}x + \tilde{\sigma}y \\ \frac{dy}{dt} &= -xz + \tilde{r}x - y \\ \frac{dz}{dt} &= xy - \tilde{b}z \end{aligned} \quad (5.5)$$

donde $\tilde{\sigma} = 10$, $\tilde{r} = 28$ y $\tilde{b} = 8/3$ son parámetros adimensionales, x es proporcional a la velocidad del flujo del fluido circulatorio, y caracteriza la diferencia de temperatura entre regiones de fluido ascendentes y descendentes, z caracteriza la distorsión del perfil de temperatura vertical con respecto de su variación de equilibrio, el número total de puntos que tiene esta ST es de 3,500 y únicamente se observan los primeros mil puntos en la Figura 5.7.

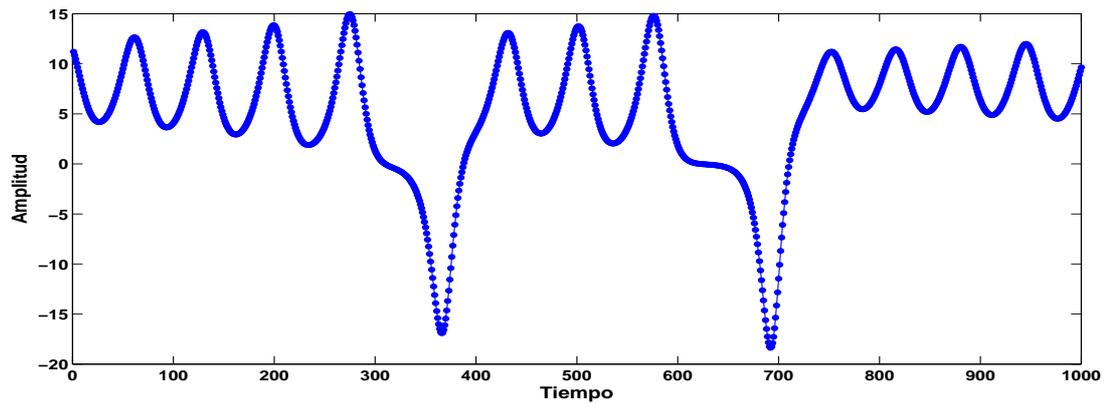


Figura 5.7: Serie de tiempo “Lorenz”.

Rossler. Serie caótica generada por un sistema de ecuaciones diferenciales y representa un modelo simplificado de Lorenz [47][50], esta ST contiene en total 5,000 puntos y sólo se observan los primeros mil en la Figura 5.8. Este sistema de ecuaciones es el siguiente:

$$\begin{aligned} \dot{x} &= -(y + z) \\ \dot{y} &= x + 0.2y \\ \dot{z} &= 0.4 + xz - 5.7z \end{aligned} \quad (5.6)$$

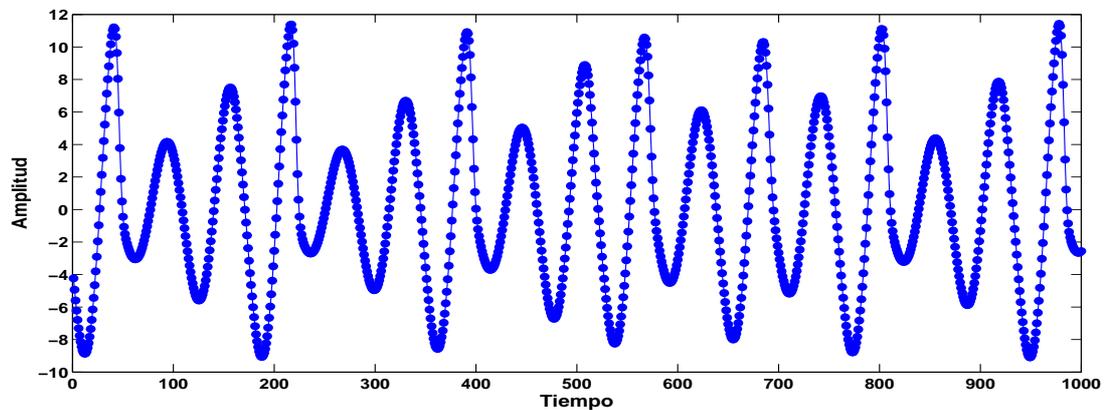


Figura 5.8: Serie de tiempo “Rossler”.

Ikeda. Serie caótica generada por un mapa construido en el plano complejo y representa un modelo de la dinámica de pulsos de luz que viajan a través de un medio no lineal [46]. La expresión matemática es de la forma:

$$z(n+1) = a + R * \exp\left(i(\phi) - \frac{p}{1 + |z^2(n)|}\right) \quad (5.7)$$

donde $z(n)$ representa al pulso que viaja a través de dicho medio. Los parámetros tienen los valores de $a = 1$, $R = 0.9$, $\phi = 0.4$ y $p = 6$. Para esta ST se cuenta con un total de 5,000 puntos, de los cuales se observan los primeros 1,000 puntos en la Figura 5.9.

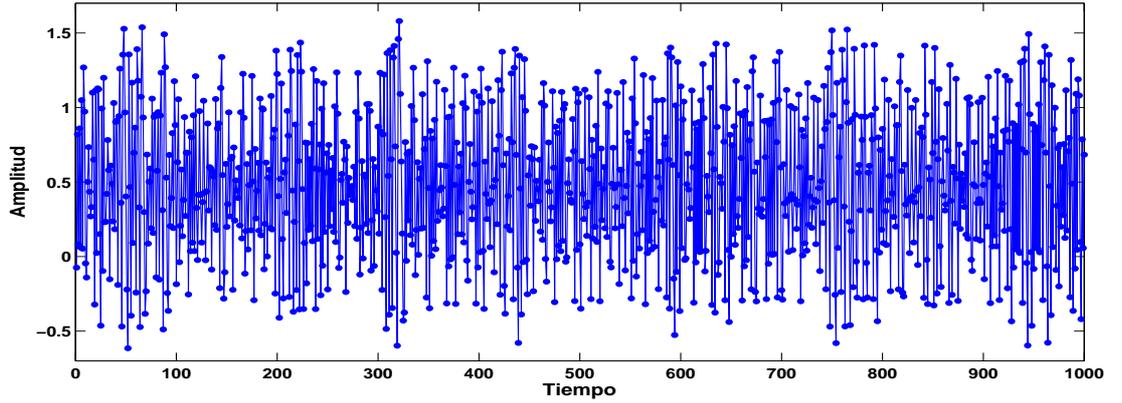


Figura 5.9: Serie de tiempo “Ikeda”.

Henon. Serie caótica generada a partir de un mapa, el cual representa un modelo simplificado del mapa de Poincaré para el modelo de Lorenz [46][51]. Para esta ST se cuenta con un total de 5,000 puntos, del cual únicamente se observa el primer 20% de los puntos en la Figura 5.10. Su expresión matemática es de la forma:

$$x(n+1) = 1 - a * x^2(n) + y(n) \quad y(n+1) = b * x(n) \quad (5.8)$$

$$a = 0.4$$

$$b = 0.3$$

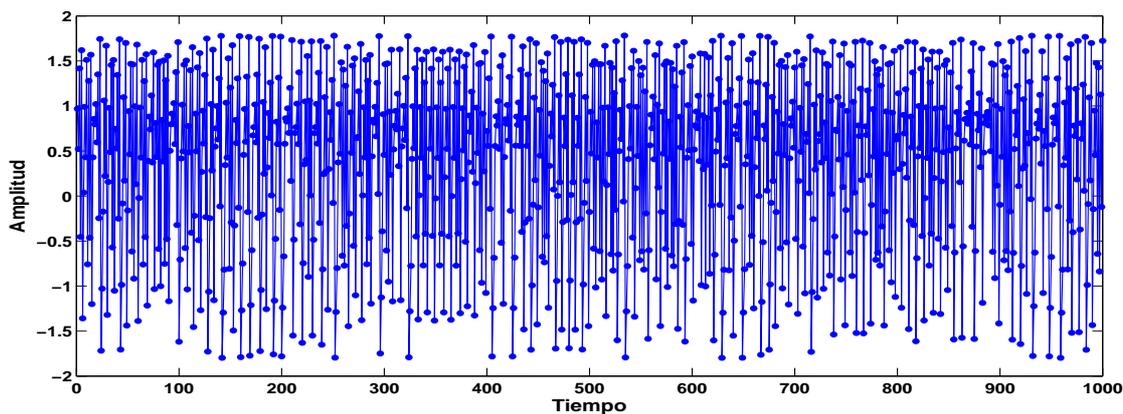


Figura 5.10: Serie de tiempo “Henon”.

Tent. Serie caótica generada por un mapa de tipo lineal por partes [46], para esta ST se cuenta con 10,000 puntos y en la Figura 5.11 se observan los primeros mil puntos de ésta. Su expresión matemática es de la forma,

$$x_{n+1} = 1 - 2 \left| x_n - \frac{1}{2} \right| \quad (5.9)$$

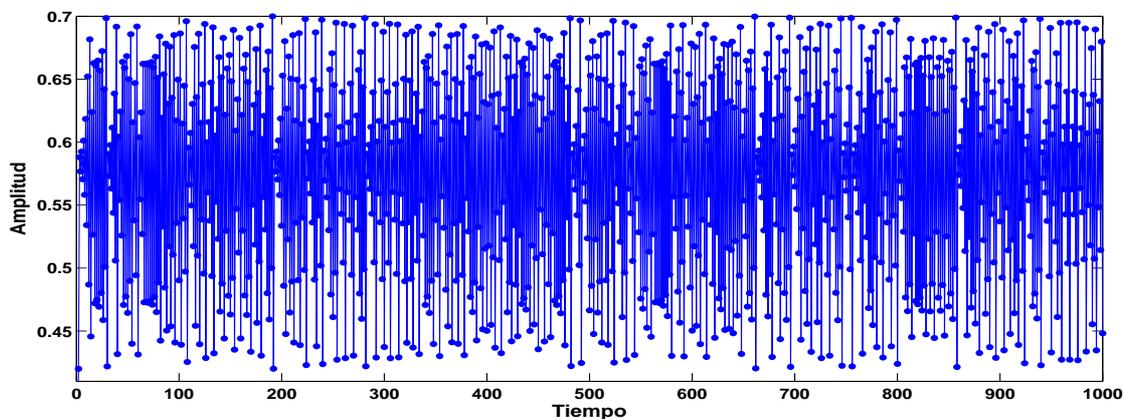


Figura 5.11: Serie de tiempo “Tent”.

A1. (Concurso Santa Fe). Serie compleja obtenida a partir de mediciones experimentales de la intensidad de un láser NH3 Infrarrojo Lejano [52], en la Figura 5.12 se observan los primeros mil puntos de un total de 25,000 puntos que contiene esta ST.

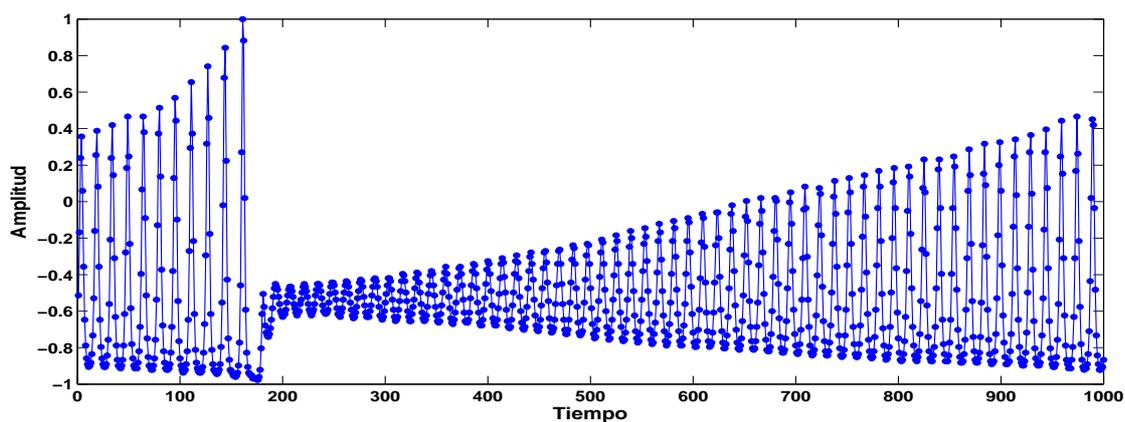


Figura 5.12: Serie de tiempo “A1” (Concurso Santa fe).

D1. (Concurso Santa Fe). Serie compleja generada por un modelo de la dinámica de una partícula amortiguada en un potencial de interacción [52], ver Figura 5.13. Esta ST contiene un total de 50,000 puntos.

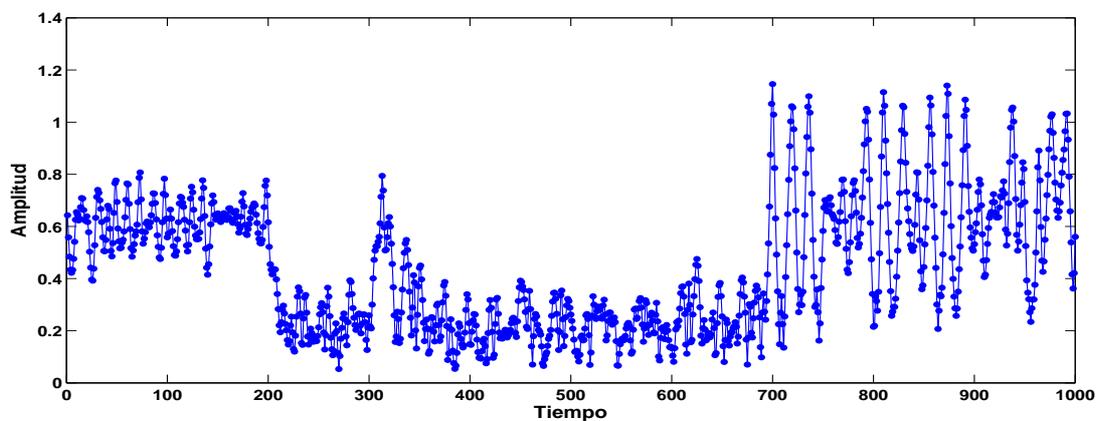


Figura 5.13: Serie de tiempo “D1” (Concurso Santa Fe).

Laser. (Concurso Santa Fe). Serie compleja obtenida a partir de mediciones experimentales de la intensidad de pulsos de láser NH3 Infrarrojo lejano con condiciones de la frecuencia ≥ 3 veces la frecuencia del sistema que genera la serie A1 [52], esta ST cuenta con un total de 5,000 puntos y en la Figura 5.14 solo se observan los primeros mil puntos.

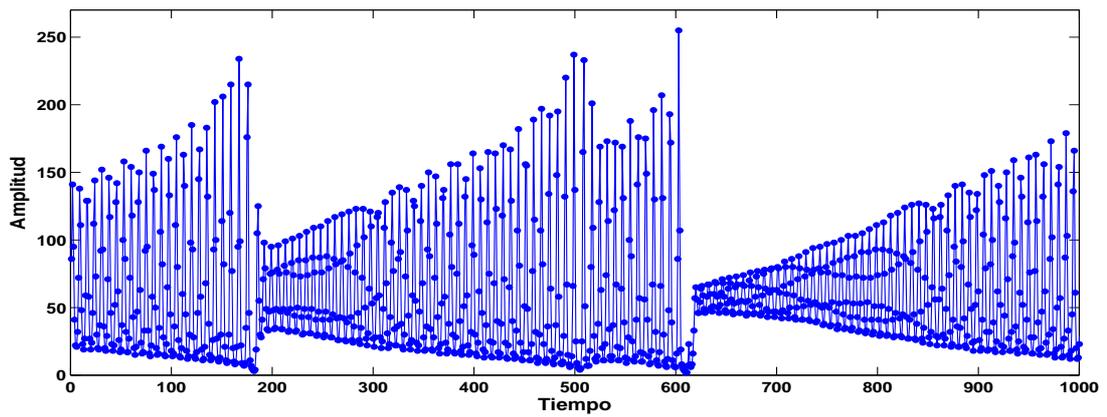


Figura 5.14: Serie de tiempo “Laser”.

El niño. Serie compleja obtenida a partir de la medición experimental de la dinámica de una variable del fenómeno climático el niño [30], y esta ST contiene un total de 2,000 puntos, en la Figura 5.15 se observan la mitad de éstos.

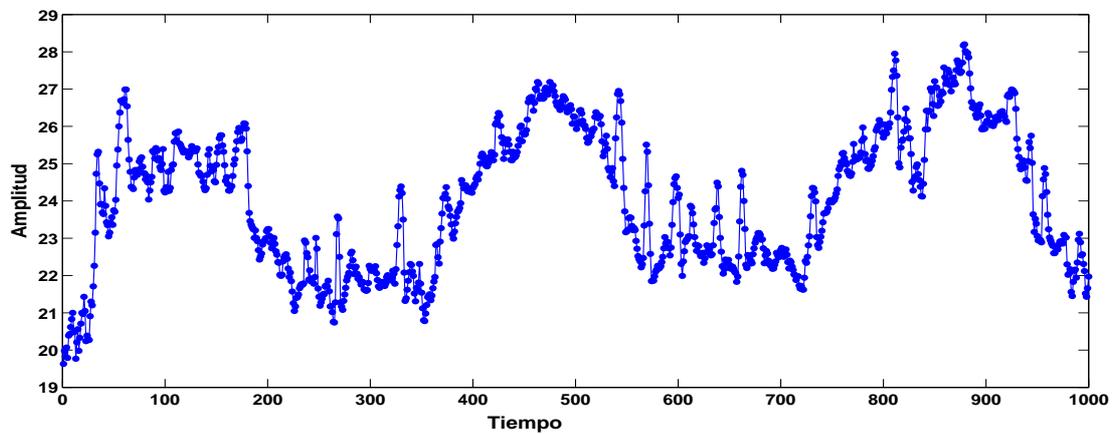


Figura 5.15: Serie de tiempo del “Niño”.

Lovaina. (Concurso Universidad de Lovaina). Serie compleja generada a partir de datos en formato ASCII [53], y esta ST contiene un total de 2,000 puntos y sólo se muestra la mitad de éstos en la Figura 5.16.

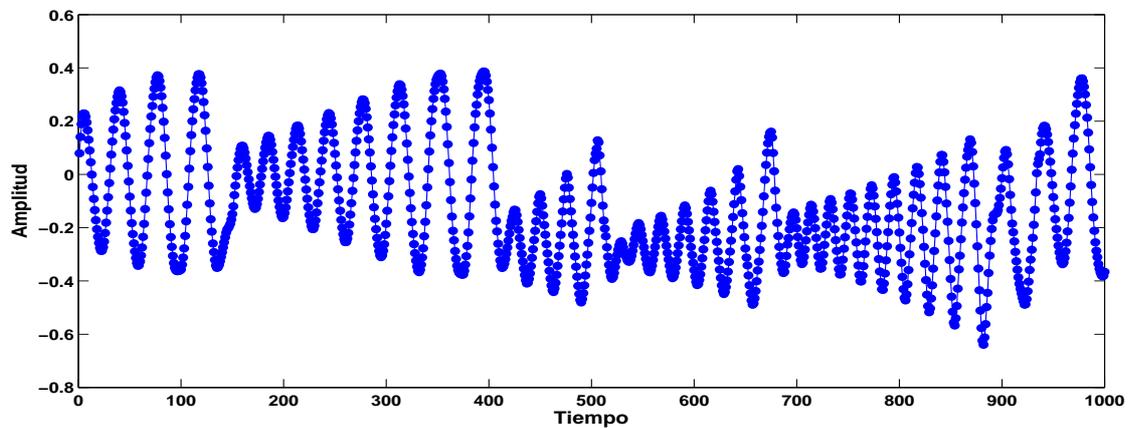


Figura 5.16: Serie de tiempo “Lovaina”.

S&P500. Serie compleja obtenida a partir del Índice Financiero de Standard & Poor para las 500 empresas más importantes de la bolsa de valores de Nueva York, y se cuenta con un total de 3,000 puntos para esta ST y en la Figura 5.17 sólo se observan los primeros mil puntos.

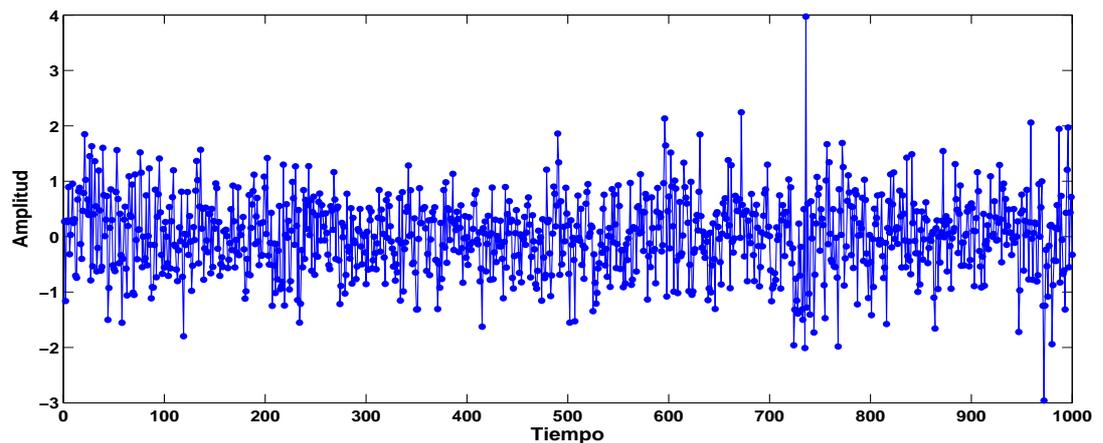


Figura 5.17: Serie de tiempo “S&P 500”.

Star. Serie compleja obtenida a partir de la medición de la intensidad luminosa de una estrella variable, y para esta ST se cuenta con 2,000 puntos como se observa en la Figura 5.18.

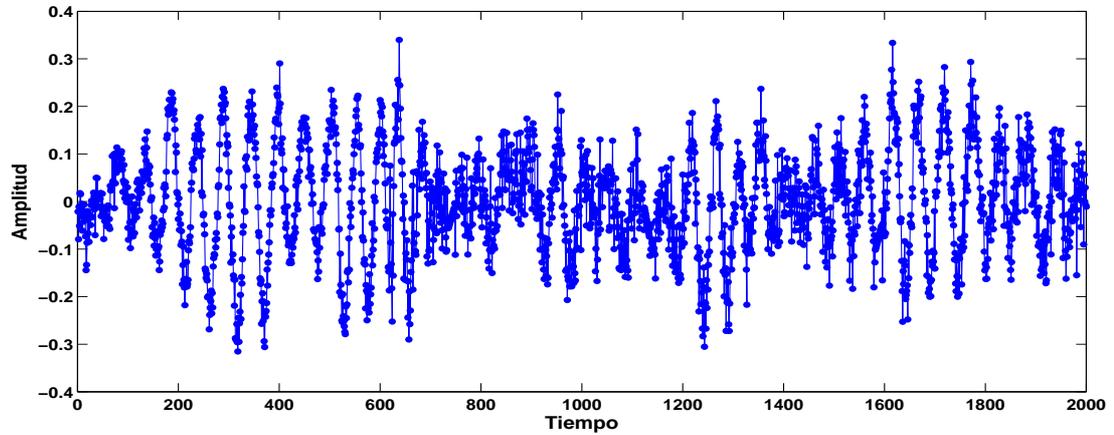


Figura 5.18: Serie de tiempo “Star”.

Brownian Motion. Serie estocástica generada a partir del modelado del movimiento browniano (proceso de ruido blanco integrado), esta serie contiene 1000 puntos como se observa en la Figura 5.19.

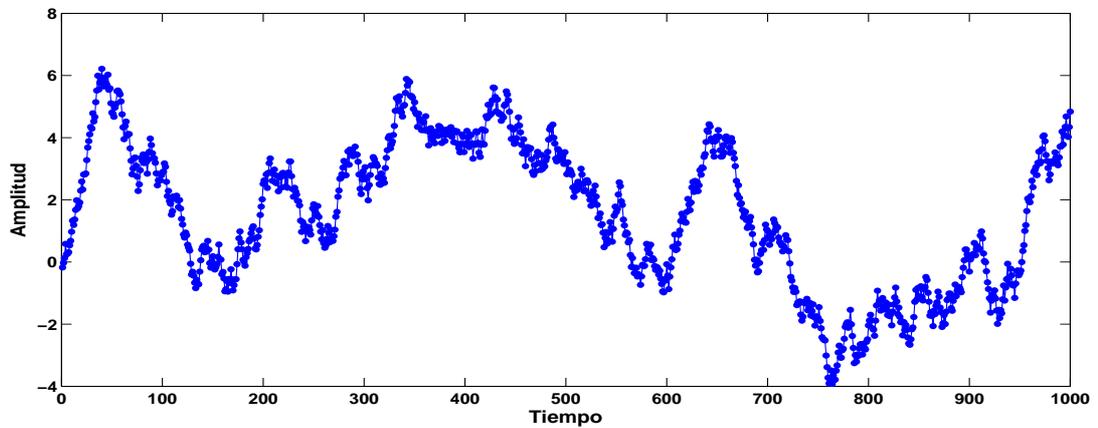


Figura 5.19: Serie de tiempo “Brown Motion”.

White Noise. Serie estocástica generada a partir del modelado de proceso de ruido blanco (ruido aleatorio uniforme), para esta ST se tiene un total de 1,000 puntos como se observa en la Figura 5.20.

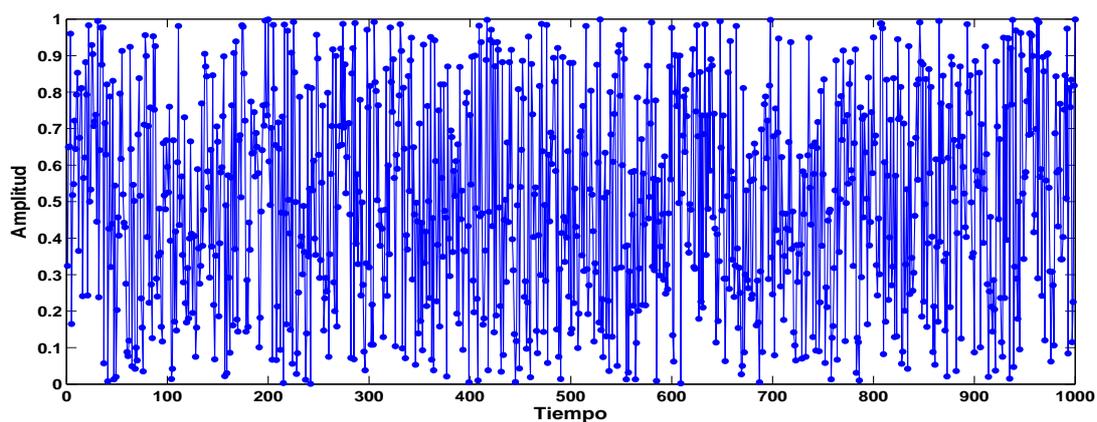


Figura 5.20: Serie de tiempo “White Noise” o también conocida como Ruido Blanco.

Precio Diario del Oro. Es una ST de tipo financiera y como su nombre lo indica, ésta monitorea diariamente el precio del Oro en dolares, a partir del primero de enero de 1985 hasta 31 de marzo de 1989 [31], por lo tanto esta ST contiene 1,074 puntos y únicamente se observan los primeros 1,000 puntos en la Figura 5.21.

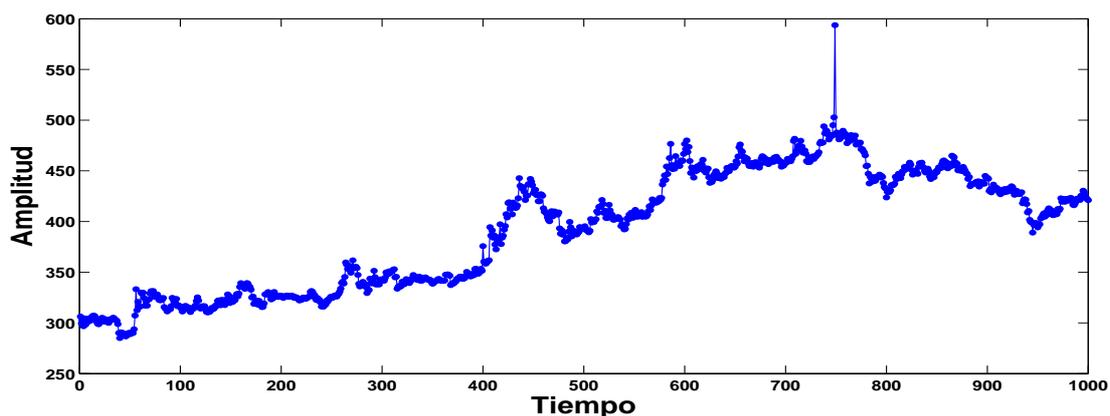


Figura 5.21: Serie de tiempo del Precio Diario del Oro.

Flujo Diario del Rio Jokulsa. Esta ST es de tipo hidrológica y tiene un comportamiento dinámico complejo. Esta serie de tiempo monitorea el flujo diario medio del rio Jokulsa Eystri. Los datos de esta ST fueron monitoreados desde el primero de enero de 1972 hasta el 31 de diciembre de 1974 [31], y por lo tanto esta ST cuenta con 1,096 puntos, pero en la Figura 5.22 se observan únicamente los primeros mil puntos.

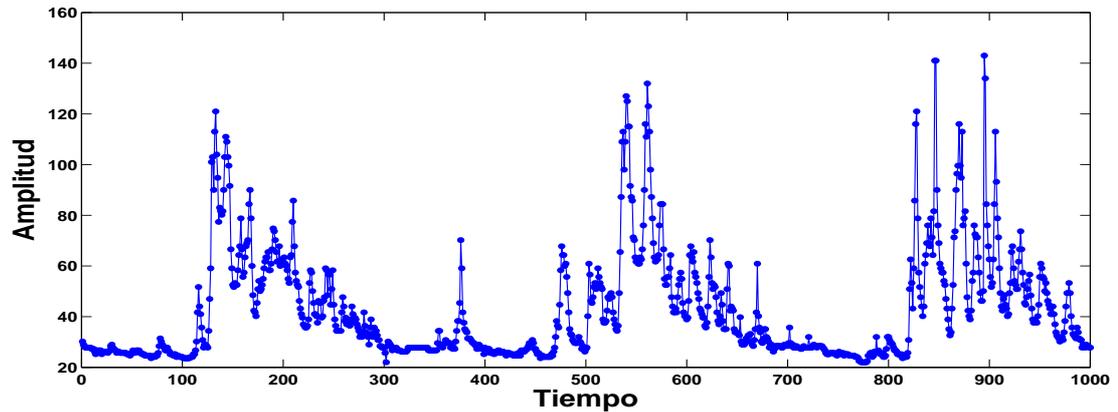


Figura 5.22: Serie de tiempo del Flujo Diario del Rio Jokulsa.

Flujo del Rio Colorado. Esta ST es de tipo hidrológica y ésta monitorea el flujo mensual del rio Colorado a partir del año 1911 al año 1972 [31] y ésta consta de 744 puntos como se observa en la Figura 5.23.

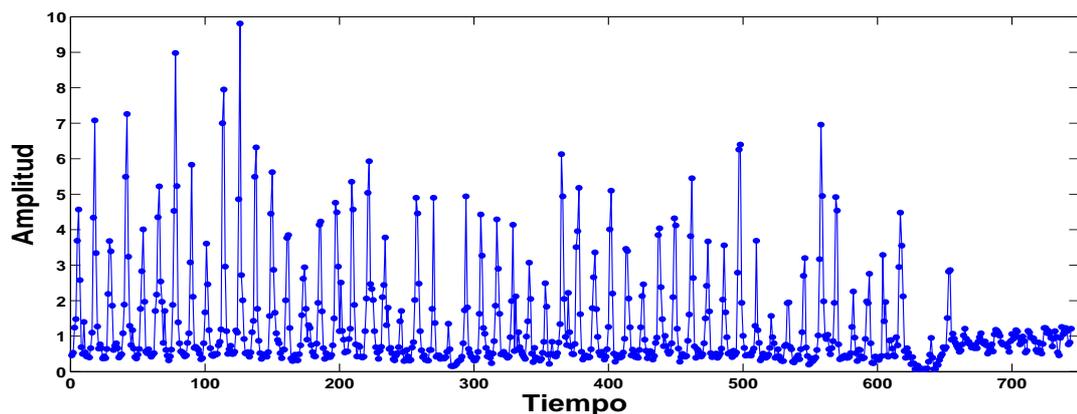


Figura 5.23: Serie de tiempo del Flujo del Rio Colorado.

Nivel del Agua del lago Erie. Esta ST es de tipo hidrológica y ésta monitorea el nivel de agua mensual del lago Erie durante los años 1921 hasta 1970 [31], por lo que esta ST cuenta con un total de 600 puntos como se observa en la Figura 5.24.

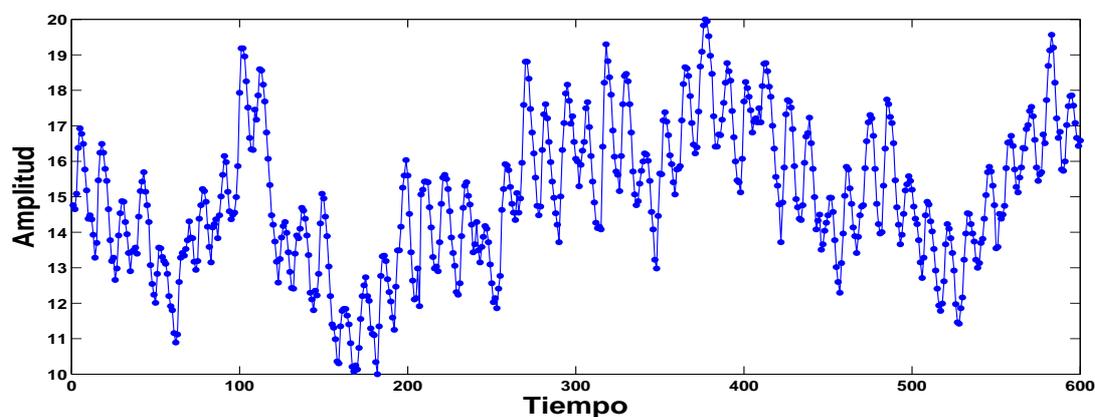


Figura 5.24: Serie de tiempo del nivel del agua del lago Erie.

Nivel de agua del río Nilo. Esta ST es de tipo hidrológica y ésta monitorea el nivel mínimo anual del agua del río Nilo durante el período del año 622 al año 1921 [31] y esta ST contiene 1,297 puntos y solo se observan los primeros mil puntos en la Figura 5.25.

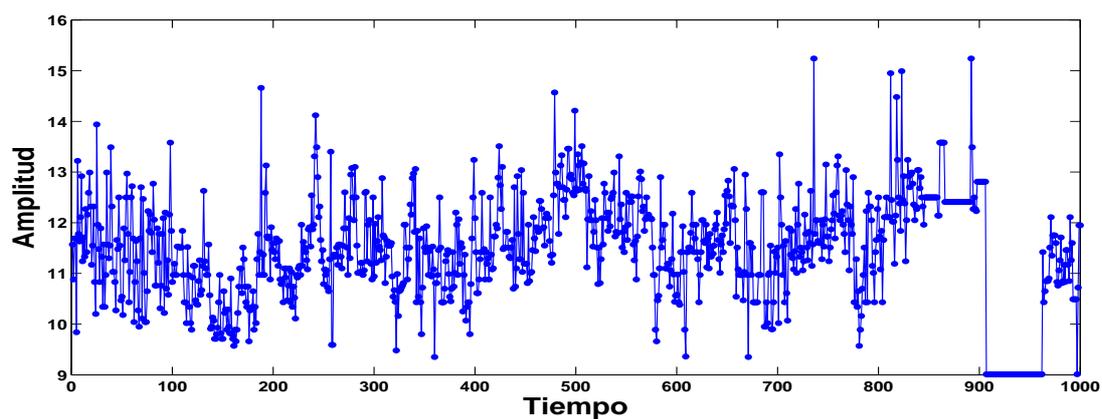


Figura 5.25: Serie de tiempo del agua del río Nilo.

Manchas Solares. Esta ST monitorea el promedio mensual de manchas solares por día y esta es una serie de categoría física, la cual está monitoreada de enero del año 1749 a marzo de 1977 [31], por lo tanto esta ST contiene 2,739 puntos y los primeros mil puntos se observan en la Figura 5.26.

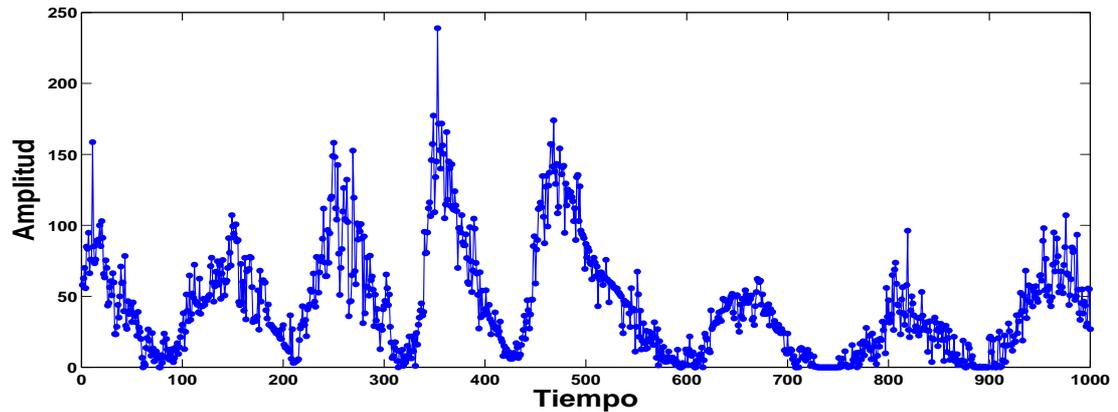


Figura 5.26: Serie de tiempo de Manchas Solares.

Temperatura máxima en Merlburne. Serie de tiempo que monitorea la temperatura máxima diaria en grados centígrados en Melburne-Australia a partir del año 1981 al año 1990 [31], y para esta ST se tienen 3,650 puntos de los cuales únicamente se observan los primeros mil en la Figura 5.27.

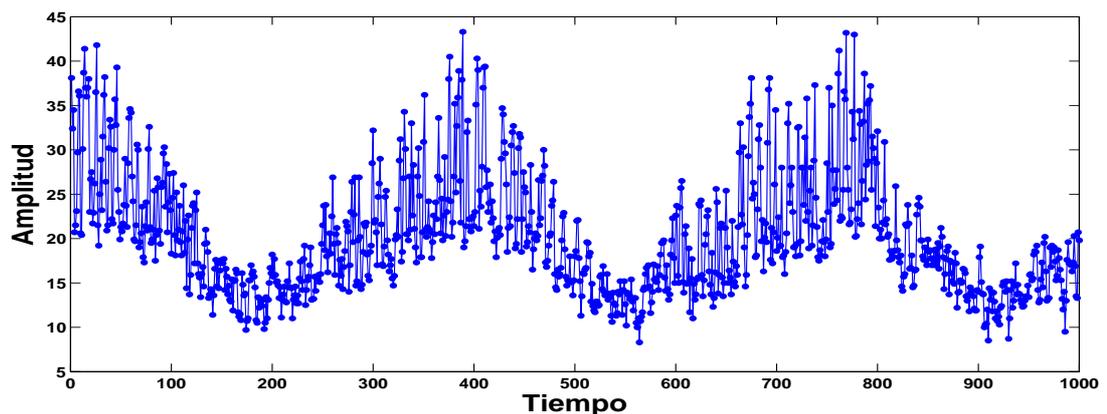


Figura 5.27: Serie de tiempo de la temperatura máxima en Merlburne.

Temperatura mínima en Merlburne. Serie de tiempo que monitorea la temperatura mínima diaria en grados centigrados en Melbourne-Australia a partir del año 1981 al año 1990 [31], y debido a eso, para esta ST se cuenta con un total de 3,650 y en la Figura 5.28 se observan los primeros mil puntos.

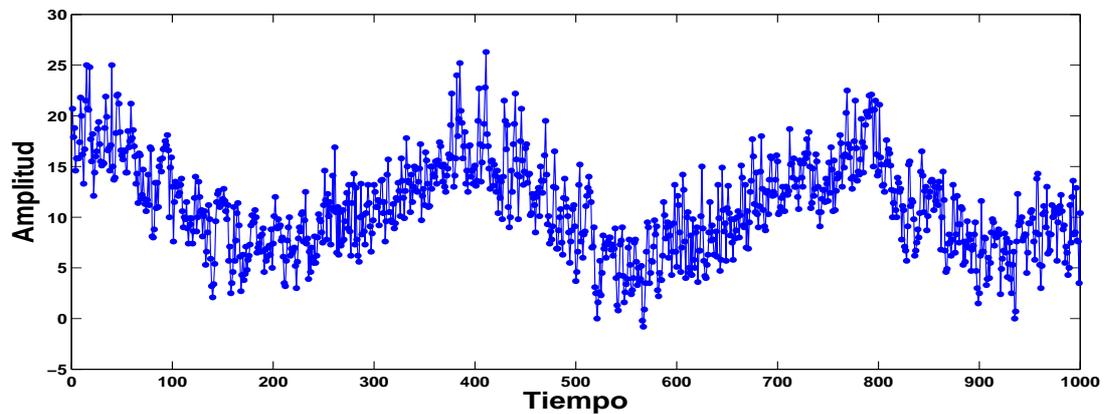


Figura 5.28: Serie de tiempo de la temperatura mínima en Merlburne.

Southern Oscillation. Esta ST monitorea oscilaciones mensuales en el meridiano, índice de diferencia de presión sobre la superficie del mar entre Darwin y Tahití y ésta ha sido monitoreada de enero de 1982 a Mayo de 1993 [31], por lo tanto esta ST contiene 1,235 puntos y en la Figura 5.29 se observan en los primeros mil puntos.

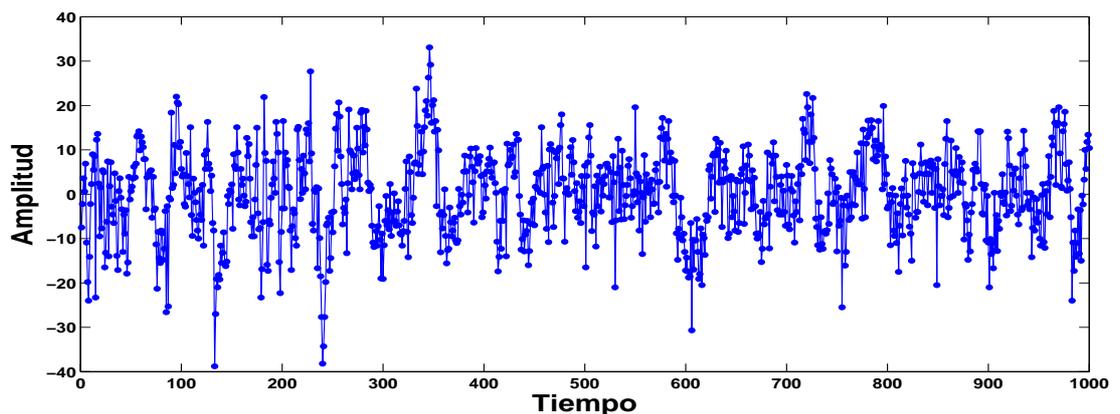


Figura 5.29: Serie de tiempo Southern Oscillation.

De las series de tiempo descritas anteriormente, es posible observar que se cuenta con un conjunto de prueba que contiene todos los comportamientos dinámicos que se definieron en la sección 2.1.3.

Para poder observar las series de tiempo que pertenecen a un comportamiento dinámico parecido, se generó la Tabla 5.1 en la cual se resume el conjunto de prueba utilizado en esta tesis. Como se observa en esta tabla, hay algunas ST que muestran la etiqueta “No definido” en la columna correspondiente al comportamiento dinámico y esto significa, que en la literatura aún no se ha definido un comportamiento dinámico para dicha ST.

Cuadro 5.1: Tabla del conjunto experimental de series de tiempo.

Número	Nombre	Comportamiento Dinámico
1	Seno	Periódico
2	Vanderpol	Periódico
3	Qperiodic2	Cuasi periódico
4	Qperiodic3	Cuasi periódico
5	Mackey&Glass	Caótico
6	Logistic	Caótico
7	Lorenz	Caótico
8	Rossler	Caótico
9	Ikeda	Caótico
10	Henon	Caótico
11	Tent	Caótico
12	A1	Complejo
13	D1	Complejo
14	Laser	Complejo
15	El niño	Complejo
16	Lovaina	Complejo
17	S&P 500	Complejo
18	Star	Complejo
19	Flujo diario del rio Jokulsa	Complejo
20	Brownian Motion	Estocástico
21	White Noise	Estocástico
22	Precio del Oro	No definido
23	Flujo del rio Colorado	No definido
24	Nivel de agua del lago Erie	No definido
25	Nivel mínimo de agua del rio Nilo	No definido
26	Número de manchas solares	No definido
27	Temperatura máxima de Melbourne	No definido
28	Temperatura mínima de Melbourne	No definido
29	Southern oscilation	No definido

5.2. Resultados del aprendizaje

Uno de los problemas dentro del área de “Machine Learning” es construir programas computacionales que automáticamente se mejoren con la experiencia. En estos últimos años se han realizado varias aplicaciones exitosas en el área de Machine Learning, como por ejemplo, en la minería de datos se han realizado programas que aprenden a detectar transacciones fraudulentas con tarjetas de crédito, otro ejemplo, son los programas para vehículos autónomos los cuales aprenden a manejar en autopistas públicas, entre otros [20]. Pero muy poco se ha reportado en la literatura sobre el aprendizaje del comportamiento dinámico de ST, el cual como ya se mencionó es útil para predicción y clasificación, por mencionar algunos.

Es debido a lo anterior que, en este trabajo, se estudia este problema y se propone una solución, en el caso particular del aprendizaje del comportamiento dinámico de ST, la cual se describió en la sección 3.2.2 y se puede resumir en los siguientes pasos:

1. Evaluar si la ST es estacionaria o no.
2. Normalizar las series de tiempo.
3. Dividir la imagen de la ST en el número adecuado de *cajas*.
4. Realizar el proceso de aprendizaje, en donde, se genera a partir de números aleatorios una secuencia que sigue el comportamiento dinámico parecido a la ST original.

Los pasos anteriores se ejemplifican en la Figura 5.30. Como se observa en el paso 4 (ver Figura 5.30d)) generamos a partir de números aleatorios una ST, que aprenda el comportamiento de una ST dada y ésta además sigue su comportamiento dinámico, demostrando así que esta nueva ST aprendió dicho comportamiento dinámico. A este proceso le hemos llamado *seguimiento* (“tracking”, en inglés). Este proceso de seguimiento sirvió de base para demostrar el rendimiento de la técnica de aprendizaje que se propone en este trabajo. Tomando en cuenta los pasos anteriores, es posible decir que en este trabajo se presenta un nuevo paradigma para obtener información de ST a partir de la información de su imagen y con esta nueva información es posible implementar una nueva técnica de aprendizaje de series de tiempo, en donde el resultado de ésta puede ser utilizado para el problema de predicción, como lo veremos en la siguiente sección de este capítulo.

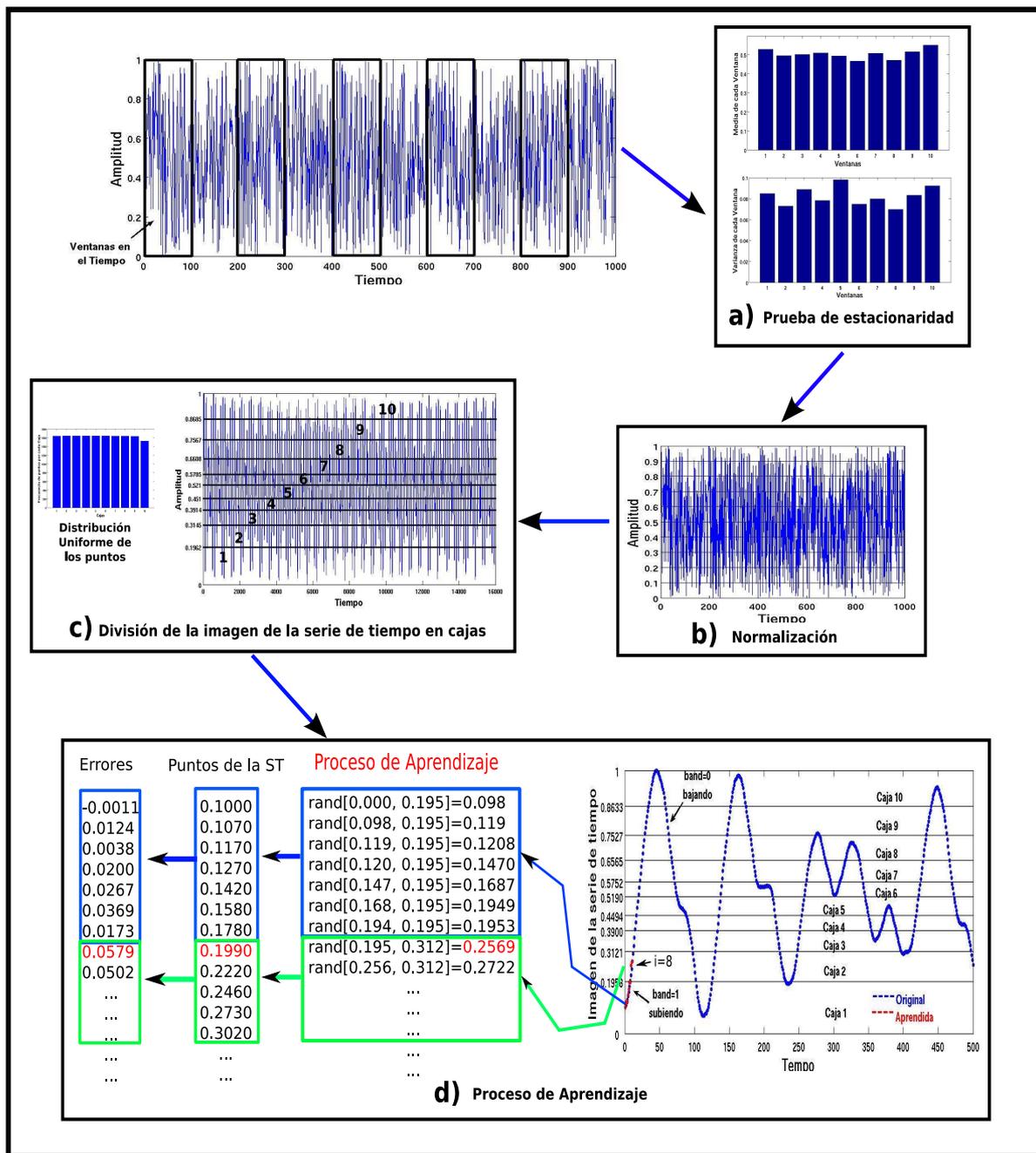


Figura 5.30: Esquema de los pasos a seguir para el aprendizaje de series de tiempo: a) se observa que en el paso 1 se realiza la prueba de estacionaridad, b) paso 2 correspondiente a la normalización de la serie de tiempo en el intervalo $[0, 1]$ ó $[-1, 1]$, c) el paso 3 que corresponde a la división de la imagen en cajas y finalmente d) se observa el proceso de aprendizaje de series de tiempo.

Para probar el rendimiento de esta nueva técnica de aprendizaje, ésta se sometió a un conjunto de prueba de series de tiempo (ver sección 5.1), que fuera una muestra representativa utilizada en el análisis y evaluación de técnicas de aprendizaje utilizadas en la literatura especializada [30][31] y además que éstas representaran diferentes comportamientos dinámicos [15] (periódico, cuasi-periódico, caótico, complejo y estocástico). Lo anterior se realizó con la finalidad de mostrar si se consigue nuestro objetivo principal, **el aprendizaje de series de tiempo**. Estas series de tiempo de prueba se observan en la Tabla 5.1.

Es posible observar que en los pasos anteriores para el aprendizaje de series de tiempo, el paso tres correspondiente a dividir la imagen de la ST en el número adecuado de cajas, es el de mayor importancia, debido a que esta división de la imagen representa las secciones en las que se irá aprendiendo algún comportamiento de la ST objetivo. Como se comentó en la sección 3.2.2, durante el desarrollo de la técnica de aprendizaje se manejaron diferentes métodos para la división de la imagen de la ST en un número adecuado de *cajas*.

Es debido a lo anterior, que los resultados del aprendizaje de series de tiempo se dividen en tres tipos de división. Cada uno de estos tipos representa el método que se utilizó para dividir la imagen de la ST generando un cierto número de *cajas* y a continuación se muestran los resultados del aprendizaje de ST para cada uno de estos tres tipos. La validación de los tres tipos es necesario para determinar cuál de los tipos de división de las ST es el más adecuado.

5.2.1. Primera tipo: anchura constante

Este tipo se refiere a los primeros experimentos que se realizaron con TABI, en donde las divisiones de la imagen de la ST fueron con respecto a la anchura de los intervalos definidos en la ecuación (3.10) de la sección 3.2.2, esto es, se dividió la imagen de la ST en 10 *cajas* en donde cada una de estas *cajas* contenían la misma anchura¹ y ésta se calculó de acuerdo a la ecuación (3.18).

Antes de que las ST pasaran por este proceso de división de la imagen, primero fueron normalizadas dentro del intervalo $[0, 1]$ y después fueron normalizadas dentro del intervalo $[-1, 1]$, por lo tanto, la anchura de las cajas para cada una de estas normalizaciones se definió de la siguiente manera:

- Para el primer intervalo de normalización de las ST, la anchura definida para cada *caja* fue de $a(0, 1] = \frac{1-0}{10} = 0.1$, quedando definidas las cotas de las cajas de la siguiente manera: $(0.0001, 0.1]$, $(0.1, 0.2]$, $(0.2, 0.3]$, $(0.3, 0.4]$, $(0.4, 0.5]$, $(0.5, 0.6]$, $(0.6, 0.7]$, $(0.7, 0.8]$ y $(0.8, 0.9]$, $(0.9, 1]$.
- Para el segundo intervalo de normalización de las ST la anchura de cada una de las 10 *cajas* fue de $a(-1, 1] = \frac{1-(-1)}{10} = 0.2$, quedando definidas las cotas de cada caja de la siguiente manera: $(-1.0001, -0.8]$, $(-0.8, -0.6]$, $(-0.6, -0.4]$, $(-0.4, -0.2]$, $(-0.2, 0.0]$, $(0.0, 0.2]$, $(0.2, 0.4]$, $(0.4, 0.6]$, $(0.6, 0.8]$, $(0.8, 1.0]$.

Los resultados de esta primera etapa para las ST normalizadas dentro del intervalo $[0, 1]$, se observan en los trabajos [15] y [16]. A lo largo de esta tesis se mostrarán únicamente los resultados para la segunda normalización de ST, esto es, cuando la imagen de la ST está dentro del intervalo $[-1, 1]$, esto debido a que durante la aplicación de esta técnica hacia la predicción de series de tiempo se realiza una comparativa de TABI con respecto a dos técnicas encontradas

¹Recordemos que la anchura de un intervalo $[y_1, y_2]$ se refiere a la longitud finita definida como $y_2 - y_1$.

en la literatura (redes neuronales y máquinas de soporte vectorial) y dichas técnicas realizan una normalización de la ST dentro del intervalo $[-1, 1]$, por lo cual consideramos más relevantes los resultados de este intervalo de normalización.

Como se comentó en secciones anteriores, el algoritmo que se propone en este trabajo es probabilístico, debido a que su resultado se basa en la solución de algunas decisiones considerando una función de densidad de probabilidad, de tal forma que en promedio, obtiene una buena solución al problema planteado para cualquier distribución de los datos de entrada [10]. Es decir, a partir de los mismos datos de entrada se pueden obtener distintas soluciones y en algunos casos soluciones erróneas.

Debido a lo anterior, los resultados se presentan con una gráfica de distribución del error de aprendizaje para cada ST dentro del conjunto experimental y esto se hizo de la siguiente manera: primero, ejecutamos el algoritmo de aprendizaje 50 veces para cada ST objetivo, posteriormente, graficamos el error que se obtuvo durante la ejecución del algoritmo de aprendizaje propuesto para cada experimento y finalmente graficamos la distribución de los errores de aprendizaje de los 50 experimentos. Esta última gráfica muestra las frecuencias obtenidas para cada conjunto de errores.

Recordemos que esta primera etapa se realizó para el conjunto de prueba de ST que se observa en la Tabla 5.1 y la cual contiene todos los comportamientos dinámicos de las series de tiempo. Por cuestiones de espacio no es posible mostrar el resultado de todos los experimentos realizados, es por eso que a continuación se mostrarán los resultados de las series de tiempo que son representativas al menos de cada uno de los comportamientos dinámicos antes mencionados.

En la Figura 5.31 observamos el resultado de la ST “Seno”, cuyo comportamiento dinámico es periódico y éste es característico de una función en la cual su forma se repite exactamente después de un periodo o ciclo [6]. Como se observa en la Figura 5.31, el algoritmo para el aprendizaje del comportamiento dinámico de ST se realizó 50 veces (ver Figura 5.31a)) y cada uno de estos 50 experimentos presentó un error de aprendizaje (RMSE) entre el rango de 0.001 y 0.0019. Se observa además, que en la Figura 5.31b) se genera la distribución de estos errores y se presenta en forma de gráfica en donde el eje de las “x” corresponde al valor de los errores de aprendizaje agrupados en cinco conjuntos (“bins”) y el eje de las “y” representa la frecuencia de ocurrencia de cada conjunto de los errores de aprendizaje obtenidos para el total de las 50 ejecuciones del algoritmo. Debido a que no es posible mostrar el resultado de las 50 corridas, únicamente se muestra el resultado del aprendizaje que obtuvo el error promedio, donde este error se calculó de la siguiente manera:

$$errorpromedio = \frac{\sum_{i=1}^N RMSE(i)}{N} \quad (5.10)$$

donde $RMSE$ corresponde al error de aprendizaje obtenido durante la ejecución de nuestro algoritmo de aprendizaje y N corresponde al número total de experimentos realizados, en este caso, igual a 50. Esta expresión representa al valor que tiene mayor probabilidad de ocurrir si volviéramos a ejecutar este algoritmo de aprendizaje.

Además, observamos que la Figura 5.31b), indica que para esta ST “Seno” el valor del error de aprendizaje que tiene mayor probabilidad de ocurrir si volviéramos a ejecutar este algoritmo de aprendizaje, sería igual a 0.00151 y donde el error de aprendizaje mínimo es de 0.0011 y el error de aprendizaje máximo es igual a 0.0019. Entonces, es posible afirmar que el rendimiento de la técnica de aprendizaje para la serie Seno es bastante aproximado a la

ST objetivo y donde aún cuando existe la posibilidad de obtener el error máximo al volver a ejecutar esta técnica de aprendizaje, aún así, ese resultado sería muy bueno.

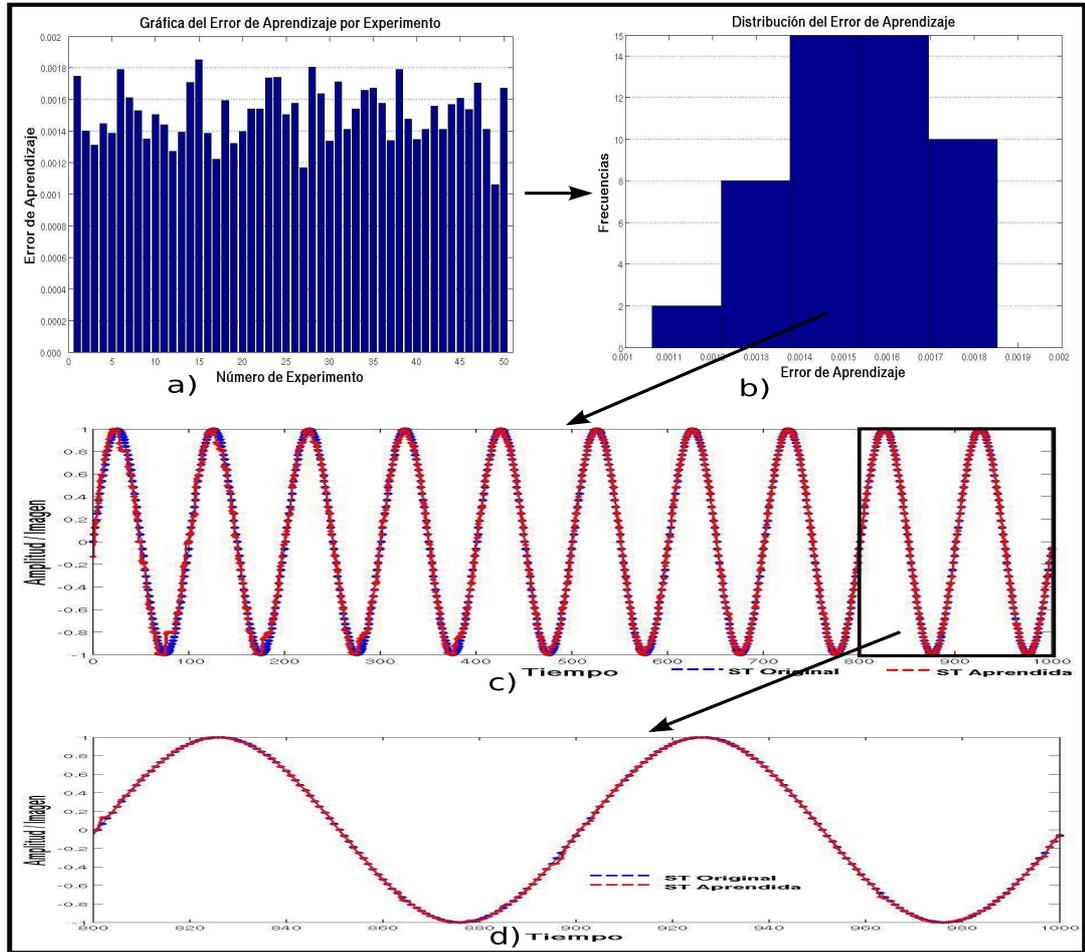


Figura 5.31: Resultados de la primera etapa para la serie Seno, la cual como se observa presenta un comportamiento dinámico periódico. En a) se observa la gráfica del error de aprendizaje de cada una de las veces que se ejecutó el algoritmo, en b) se observa la distribución de estos errores, en c) se observa el resultado promedio de la ST Seno para todos sus puntos y finalmente en d) se observa un acercamiento de los últimos 200 puntos de la ST original y de la ST generada.

En la Figura 5.32, se observan los resultados de aplicar la técnica TABI a una serie de tiempo cuyo comportamiento es cuasi-periódica. Esto significa que la evolución de ésta en el tiempo puede ser descompuesta en diferentes secciones las cuales son periódicas con si mismas, pero en diferentes períodos. Como se observa en dicha figura el algoritmo para el aprendizaje del comportamiento dinámico de la ST se realizó 50 veces, donde cada uno de estos 50 experimentos presentó un error de aprendizaje (RMSE) entre el rango de 0.00215 y 0.0022. También se observa la distribución de estos errores en forma de gráfica en donde el eje de las “x” corresponde al valor de los errores de aprendizaje agrupados en cinco conjuntos y el eje de

las “y” representa la frecuencia de ocurrencia para cada conjunto de los errores de aprendizaje obtenidos para el total de las 50 ejecuciones del algoritmo. Únicamente mostramos el error de aprendizaje (RMSE) promedio para esta serie de tiempo, el cual fue igual a 0.00216 (ver Figura 5.32d)). Si observamos una vez más la Figura 5.32b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje el error que se obtenga sea muy bajo es mayor, esto es, existe una mayor probabilidad de obtener un buen resultado.

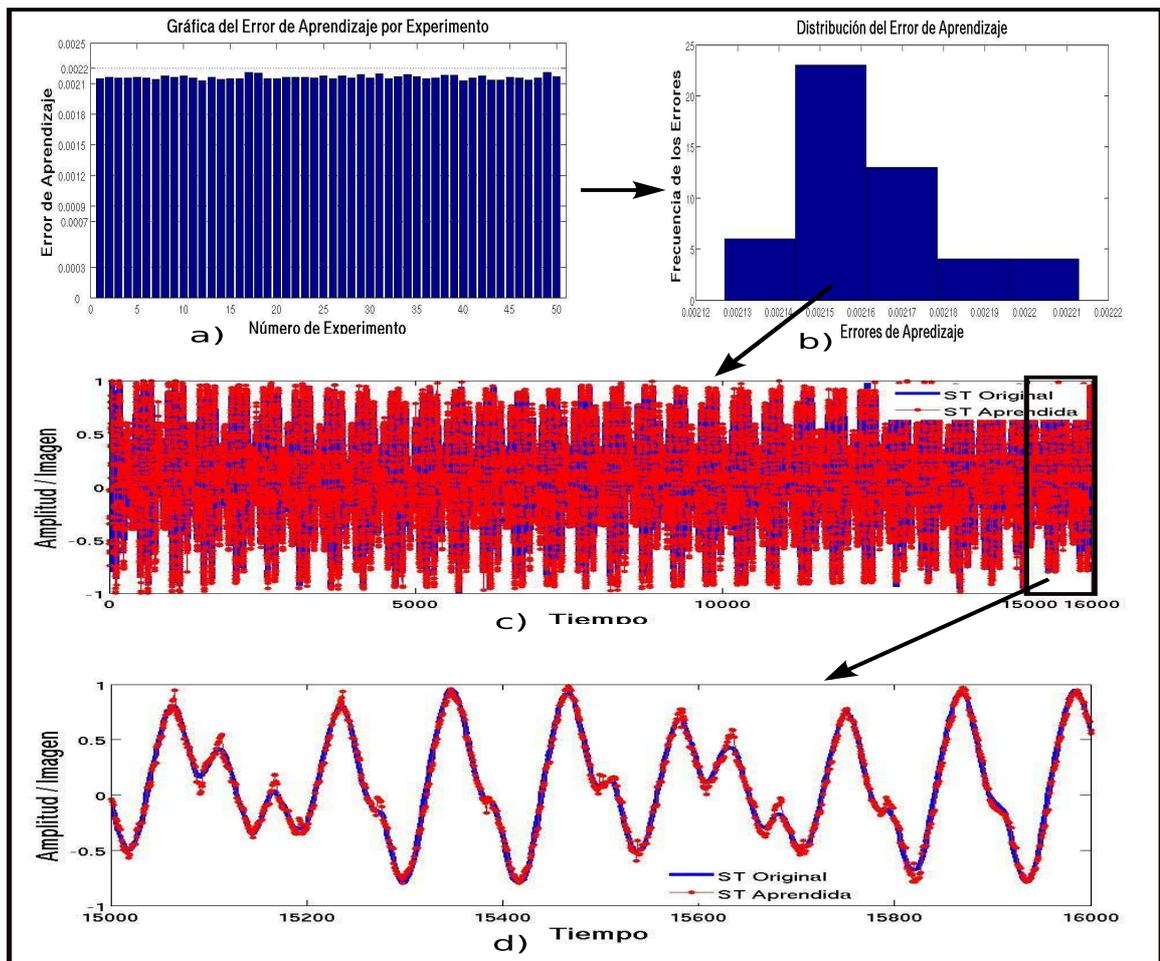


Figura 5.32: Resultados de la primera etapa para la ST Qperiodic2, la cual presenta un comportamiento cuasi-periódico, en esta figura observamos lo siguiente: a) muestra el resultado de los errores de aprendizaje que se obtienen para cada uno de los 50 experimentos, en b) se observa la distribución de los errores de aprendizaje, en c) se observa el resultado promedio de la serie aprendida y la ST original y en d) se observa un acercamiento de los últimos 1000 puntos de las ST.

En la Figura 5.33, se observan los resultados de aplicar la técnica TABI a una serie de tiempo cuyo comportamiento es cuasi-periódico igual que la ST anterior. Como se observa en dicha figura el algoritmo para el aprendizaje del comportamiento dinámico de ST se realizó 50 veces, donde cada uno de estos 50 experimentos presentó un error de aprendizaje (RMSE) dentro del rango de 0.00355 y 0.00362. De igual manera se observa que la distribución de los errores de aprendizaje después de ejecutar el algoritmo propuesto 50 veces genera un error promedio de aprendizaje (RMSE) para esta serie de tiempo, igual a 0.00355. Si observamos una vez más la Figura 5.33b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es muy alta debido a esta gráfica de distribución.

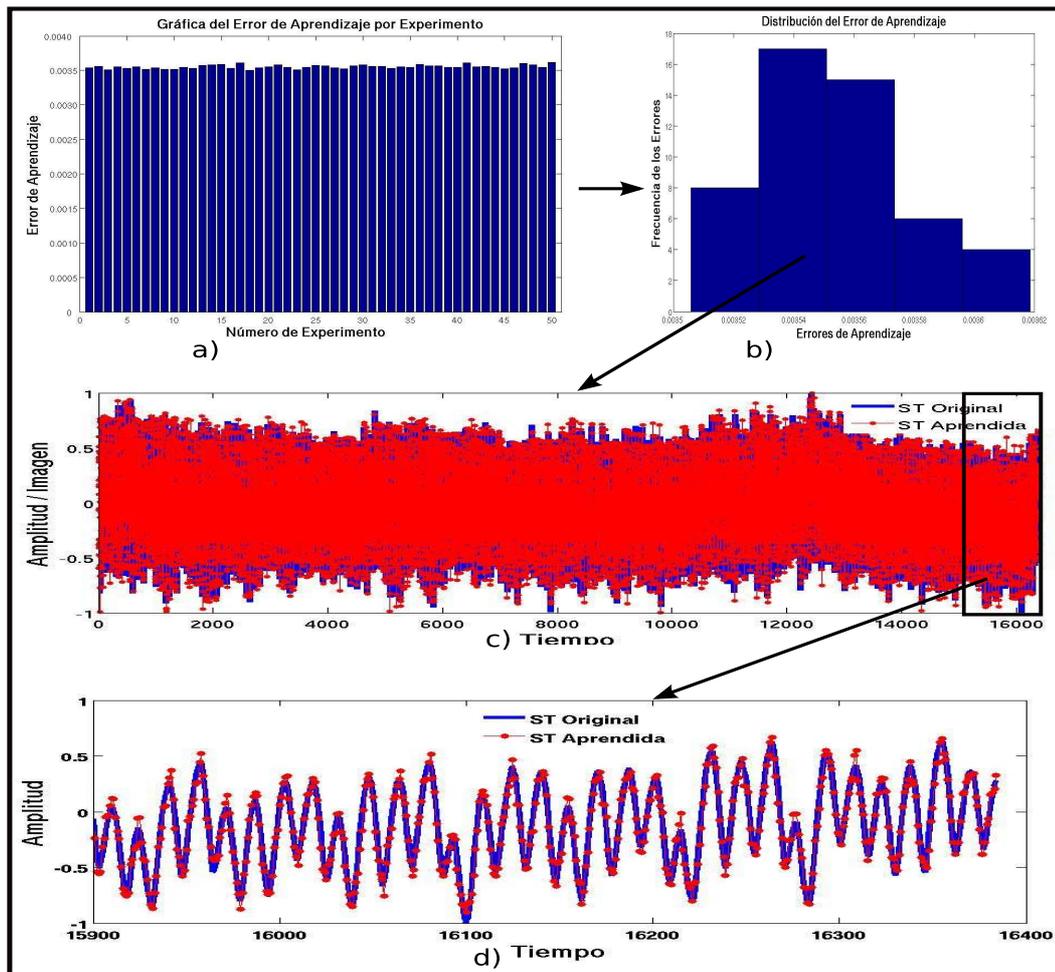


Figura 5.33: Resultados de la primera etapa para la ST Qperiodic3, la cual presenta un comportamiento cuasi-periódico. Observamos en a) la gráfica del error de aprendizaje para cada experimento, en b) se observa la distribución de estos errores de aprendizaje, c) muestra el resultado promedio para todos sus puntos y finalmente d) muestra el acercamiento de los últimos 500 puntos de la ST original (línea continua azul) y la predicha (línea punteada roja).

En la Figura 5.34, se observan los resultados de aplicar TABI a la ST “Lorenz” cuyo com-

portamiento dinámico es caótico. Éste describe un comportamiento de algún sistema dinámico no lineal que bajo ciertas condiciones es sensible a las condiciones iniciales, un ejemplo clásico de este comportamiento es el *efecto mariposa*. A pesar de su comportamiento dinámico, al ejecutar la técnica de aprendizaje propuesta 50 veces, éste presentó un error de aprendizaje (RMSE) dentro del rango de 0.00255 y 0.0029. De igual manera que las figuras anteriores, en ésta se observa que la distribución de los errores de aprendizaje después de ejecutar el algoritmo propuesto 50 veces, éste genera un error promedio (RMSE) igual a 0.00273, que es el error que obtuvo durante el primer experimento de la técnica de aprendizaje, la cual, se observa en la Figura 5.34c), para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 1,500 puntos como se observa en la Figura 5.34d). Si observamos una vez más la Figura 5.34b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es muy alta debido a esta gráfica de distribución.

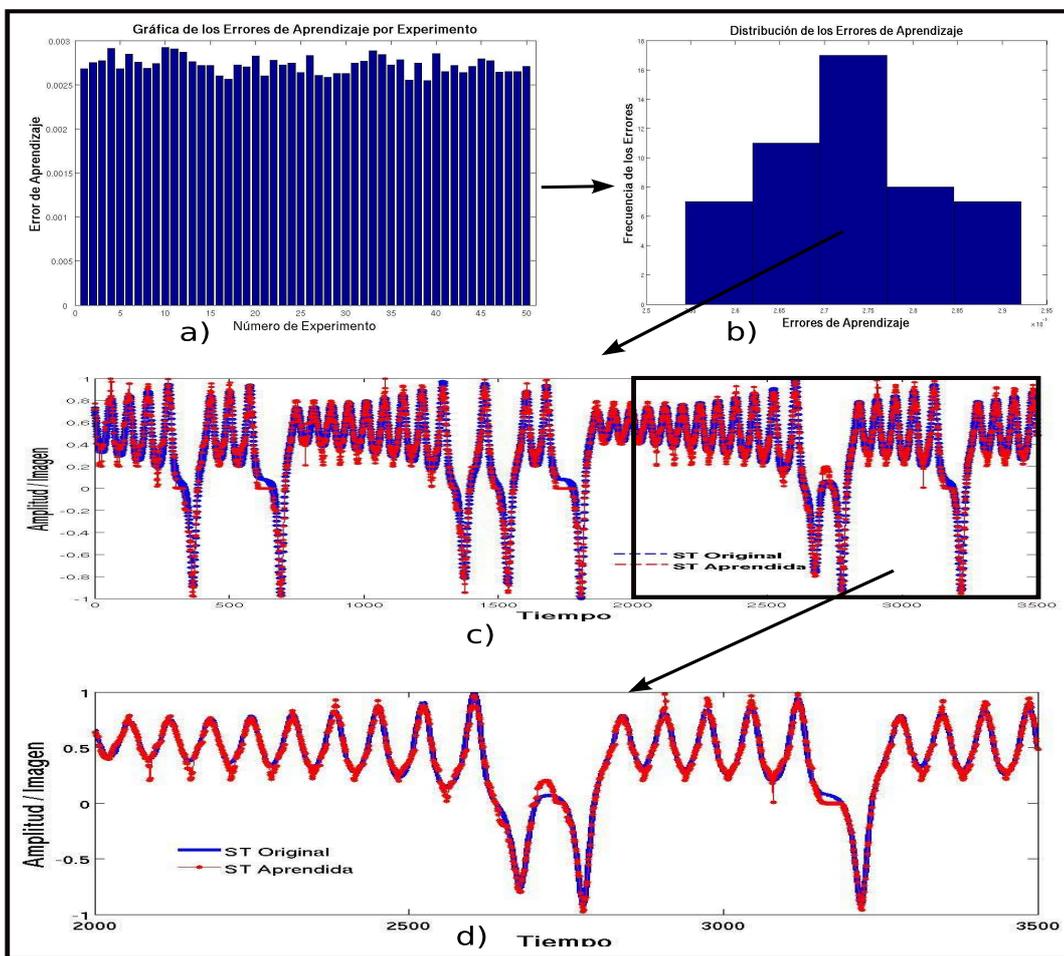


Figura 5.34: Resultados de la primera etapa para la ST "Lorenz", la cual presenta un comportamiento caótico. Se observa en a) la gráfica de los errores obtenidos durante cada experimento, b) muestra la distribución de estos errores, en c) observamos el aprendizaje de la ST "Lorenz" y finalmente d) muestra un acercamiento de los 500 puntos.

En la Figura 5.35, se observan los resultados de aplicar TABI a la ST “tent” cuyo comportamiento dinámico es caótico como la ST anterior y a pesar de su comportamiento dinámico, al ejecutar la técnica de aprendizaje propuesta 50 veces, éste presentó un error de aprendizaje (RMSE) dentro del rango de 0.0063 y 0.00655. De manera similar que las figuras anteriores, en ésta se observa que la distribución de los errores de aprendizaje después de ejecutar el algoritmo propuesto 50 veces, éste genera un error promedio (RMSE) igual a 0.00645, que es el error que obtuvo durante el cuarto experimento de la técnica de aprendizaje y la ST que se genera de este experimento se observa en la Figura 5.35c) y para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 100 puntos como se observa en la Figura 5.35d). Si observamos una vez más la Figura 5.35b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es muy alta debido a esta gráfica de distribución.

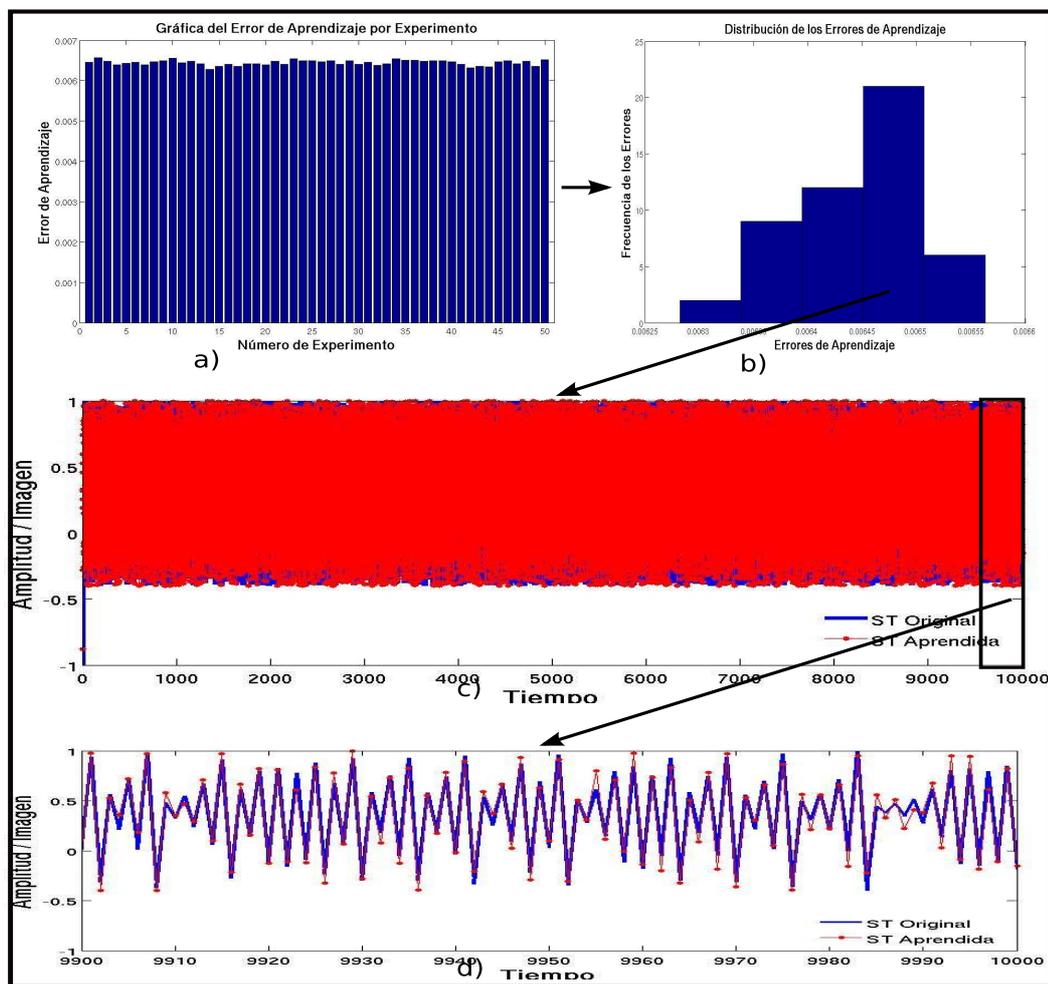


Figura 5.35: Resultados de la primera etapa de la ST “tent”, la cual tiene un comportamiento caótico. Se ilustra, en a) los errores obtenidos en cada experimento, en b) se observa la distribución de los errores, c) muestra el aprendizaje de toda la ST y finalmente d) muestra el acercamiento de los últimos 100 puntos.

En la Figura 5.36, se observan los resultados de aplicar TABI a la ST “S&P 500” cuyo comportamiento dinámico es complejo. Algunos autores utilizan este término para definir el comportamiento de alguna señal como no periódica o no constante pero sin llegar a ser aleatorio, a pesar de este comportamiento dinámico, al ejecutar la técnica de aprendizaje propuesta 50 veces, éste presentó un error de aprendizaje (RMSE) dentro del rango de 0.00445 y 0.00483. De manera similar que las figuras anteriores, en ésta se observa que la distribución de los errores de aprendizaje después de ejecutar el algoritmo propuesto 50 veces y genera un error promedio (RMSE) igual a 0.00462, que es el error que obtuvo durante el cuarto experimento de la técnica de aprendizaje y la ST que se genera de este experimento se observa en la Figura 5.36c) y para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 500 puntos como se observa en la Figura 5.36d). Si observamos una vez más la Figura 5.36b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es muy alta, debido a esta gráfica de distribución.

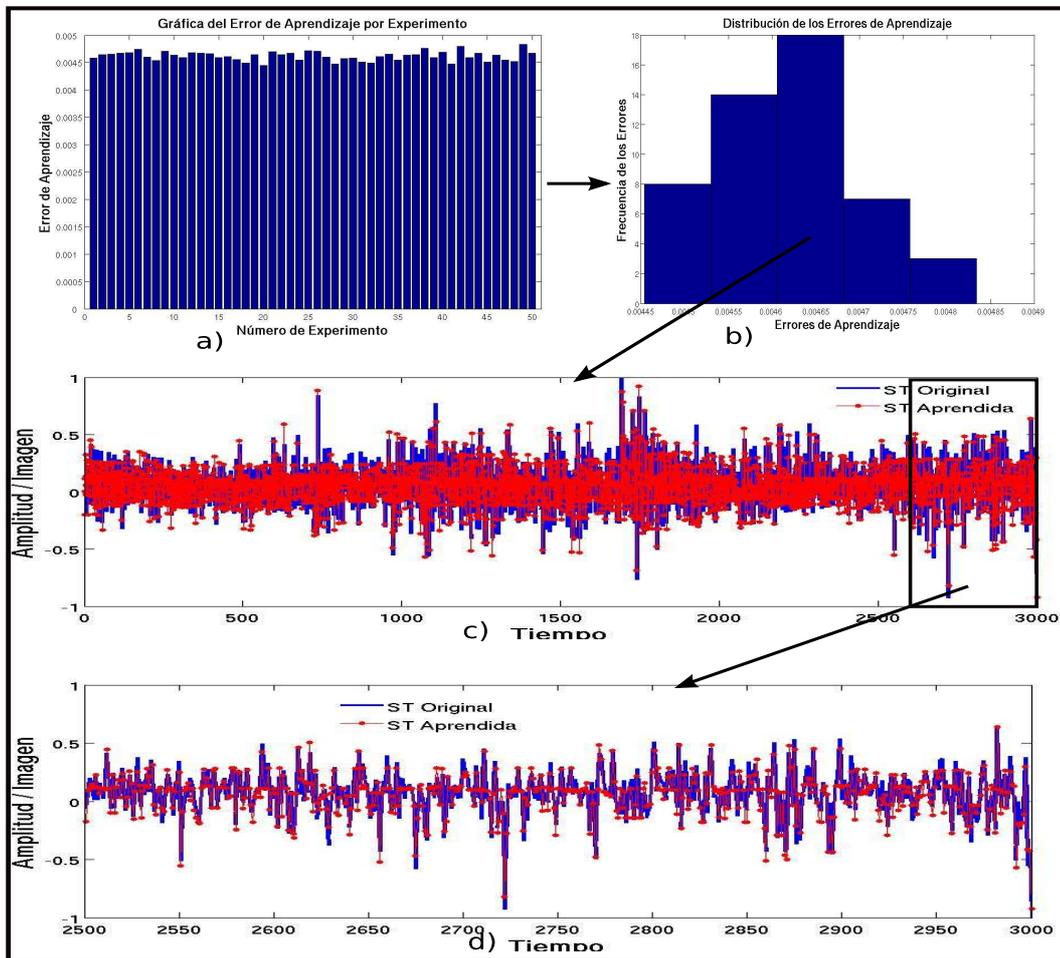


Figura 5.36: Resultado de la primera etapa de la ST “S&P 500”. a) exhibe los resultados obtenidos por cada experimento, b) muestra la distribución de estos errores, c) ilustra el resultado del aprendizaje para esta ST y d) muestra un acercamiento de los últimos 500 puntos.

En la Figura 5.37, se observan los resultados de aplicar TABI a la ST “Brownian Motion”, la cual presenta uno de los comportamientos dinámicos más complicados y nos referimos al estocástico. Un proceso estocástico, es aquél cuyo comportamiento es no determinístico, en el que un estado no determina completamente su siguiente estado y a pesar de ser uno de los comportamientos más complicados, al ejecutar la técnica de aprendizaje propuesta 50 veces, éste presentó un error de aprendizaje (RMSE) dentro del rango de 0.0038 y 0.00465. De manera similar que las figuras anteriores, en esta se observa que la distribución de los errores de aprendizaje, que para esta STes igual a igual a 0.00418 y este error se obtuvo durante el primer experimento de la técnica de aprendizaje propuesta y la ST que se genera de este experimento se observa en la Figura 5.37c) y para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 300 puntos como se observa en la Figura 5.37d). Si observamos una vez más la Figura 5.37b) es posible notar que existe una probabilidad muy alta de obtener un buen resultado de aprendizaje al volver a ejecutar la técnica de aprendizaje, debido a la gráfica de distribución para esta ST.

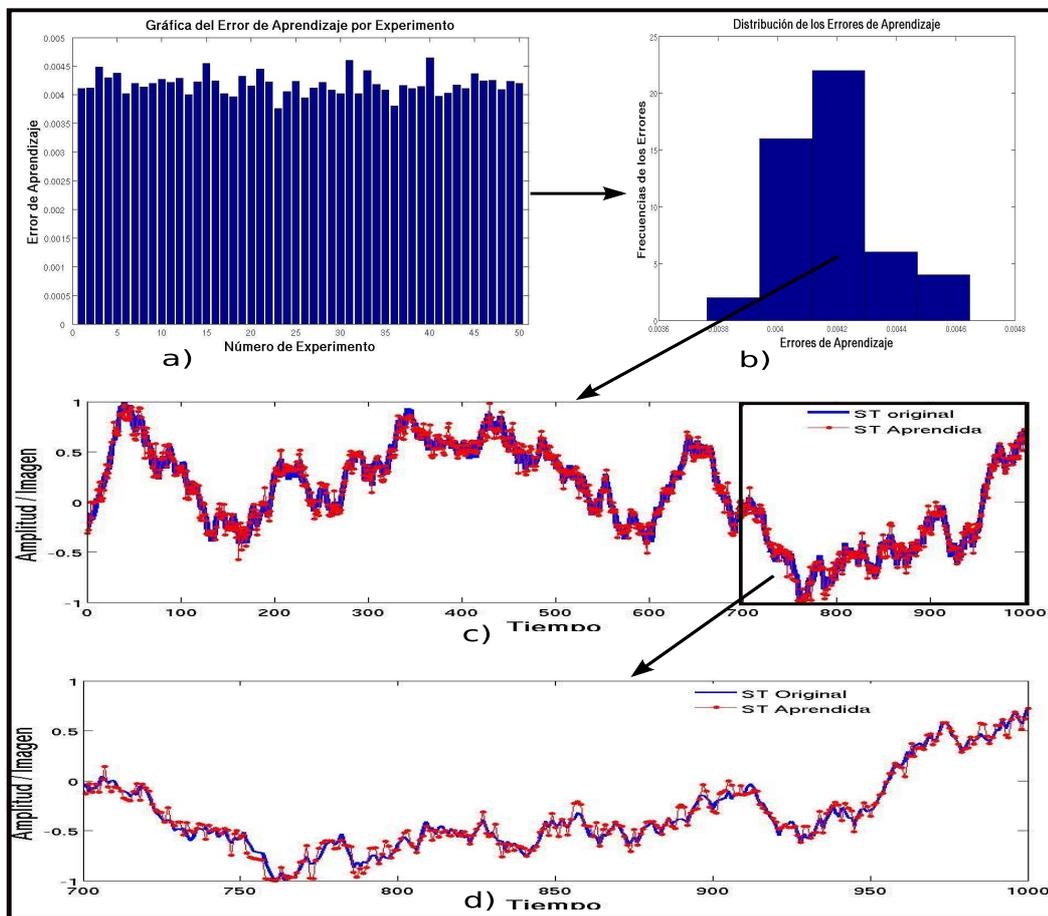


Figura 5.37: Resultado de la primera etapa de la ST “Brown Motion”, la cual presenta un comportamiento estocástico, en a) se observa el resultado del error de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) se observa el aprendizaje para la ST completa y finalmente en d) se observa un acercamiento de los últimos 300 puntos.

5.2.2. Segundo tipo: división por frecuencias

Recordemos que este tipo de división se realiza a través de la imagen de la ST en *cajas*, de acuerdo a los siguiente:

- Primero, se obtiene la distribución de los puntos para cada caja, la cual se generó al dividir la imagen de la ST en *cajas* con la misma anchura, como en la etapa anterior, de donde obtenemos una frecuencia máxima y una frecuencia mínima,
- A partir de estas frecuencias se obtiene la frecuencia promedio y ésta será igual al número de puntos que debe de contener cada *caja*.
- Posteriormente, se procede a generar los intervalos de cada *caja* de acuerdo al número de puntos establecidos por las frecuencias, haciendo así que la división de la serie de tiempo sea dinámica.

Una explicación más detallada se observa en la sección 3.2.2. Debido a lo anterior, a esta segunda etapa, la llamaremos *división por frecuencias*.

A continuación se mostrarán algunos resultados obtenidos durante la aplicación de la metodología de esta segundo tipo de división en las siguientes figuras. Estas figuras corresponden a las ST que fueron utilizadas y descritas en el paso anterior, esto con la finalidad de realizar una comparación en cuanto al rendimiento de este segundo tipo de división respecto del anterior.

Debido a lo anterior, para cada una de estas figuras se tiene lo siguiente:

- Primero, se muestra el resultado de los errores de aprendizaje para cada uno de los 50 experimentos realizados.
- Posteriormente, se genera una distribución de probabilidad de ocurrir de dichos errores de aprendizaje.
- Luego, se observará el resultado del experimento que obtuvo su error cercano al promedio. En esta misma gráfica se observan dos series, una corresponde a la ST original (línea continua azul) y la otra correspondiente a la ST aprendida (línea punteada roja).
- Finalmente, ilustraremos un acercamiento de dichas ST, con el fin de observar a mayor detalle la precisión de la técnica de aprendizaje para este tipo de división.

Siguiendo con el orden de las series vistas en la etapa anterior, primero observemos la Figura 5.38, la cual, muestra el resultado de ejecutar la metodología de la división por frecuencias para la ST “Seno”, cuyo comportamiento es periódico. Esta ST presentó un error de aprendizaje (RMSE) entre el rango de 0.002 y 0.0047, el cual si lo comparamos con la etapa anterior es mayor. Además si observamos nuevamente la Figura 5.38b), ésta indica que para esta ST “Seno” el valor del error de aprendizaje que tiene mayor probabilidad de ocurrir si volviéramos a ejecutar este algoritmo sería igual a 0.00351, el cual es muy acertado. Entonces, es posible afirmar que el rendimiento de la técnica de aprendizaje para la serie Seno es bastante aproximado a la ST objetivo, aun que no tan bueno como en la etapa anterior y donde aun cuando existe la posibilidad de obtener el error máximo al volver a ejecutar esta técnica de aprendizaje, aun así, este resultado es bueno.

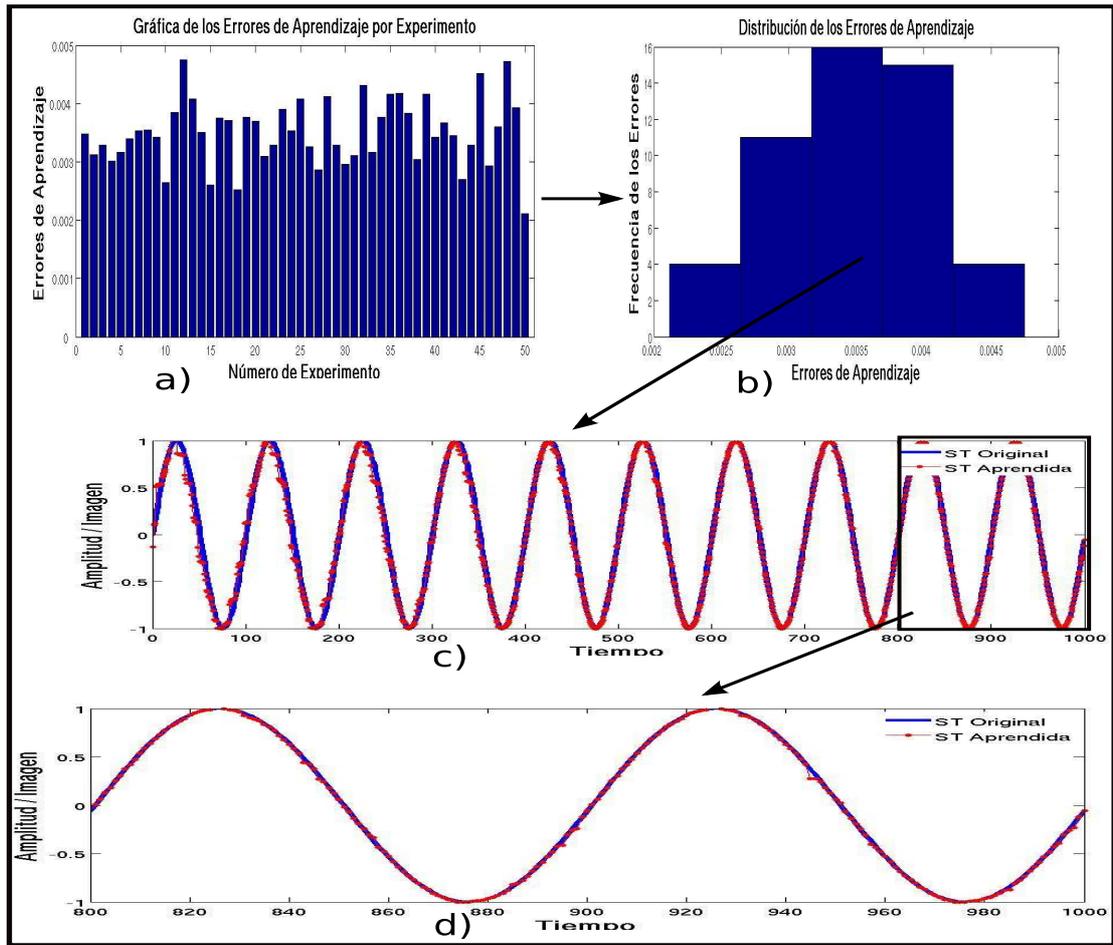


Figura 5.38: Resultado de la segunda etapa para la ST “Seno”. Se muestra en a) se observa el resultado del error de aprendizaje para cada experimento, b) muestra la distribución del error de aprendizaje, c) muestra el resultado de la ST que obtuvo el error promedio y finalmente en d) muestra un acercamiento de los 200 puntos.

En la Figura 5.39, se observan los resultados de aplicar este tipo de división a la ST “Qperiodic3”, cuyo comportamiento es cuasi-periódico. Ésta presentó un error de aprendizaje (RMSE) dentro del rango de 0.00312 y 0.0035. De igual manera se observa que la distribución de los errores de aprendizaje después de ejecutar el algoritmo propuesto 50 veces genera un error promedio de aprendizaje (RMSE) igual a 0.00327 y es el correspondiente al experimento que se observa en la Figura 5.39d), el cual es menor que la división anterior, lo que significa que esta metodología aplicado a esta ST tiene una mayor precisión que la división anterior.

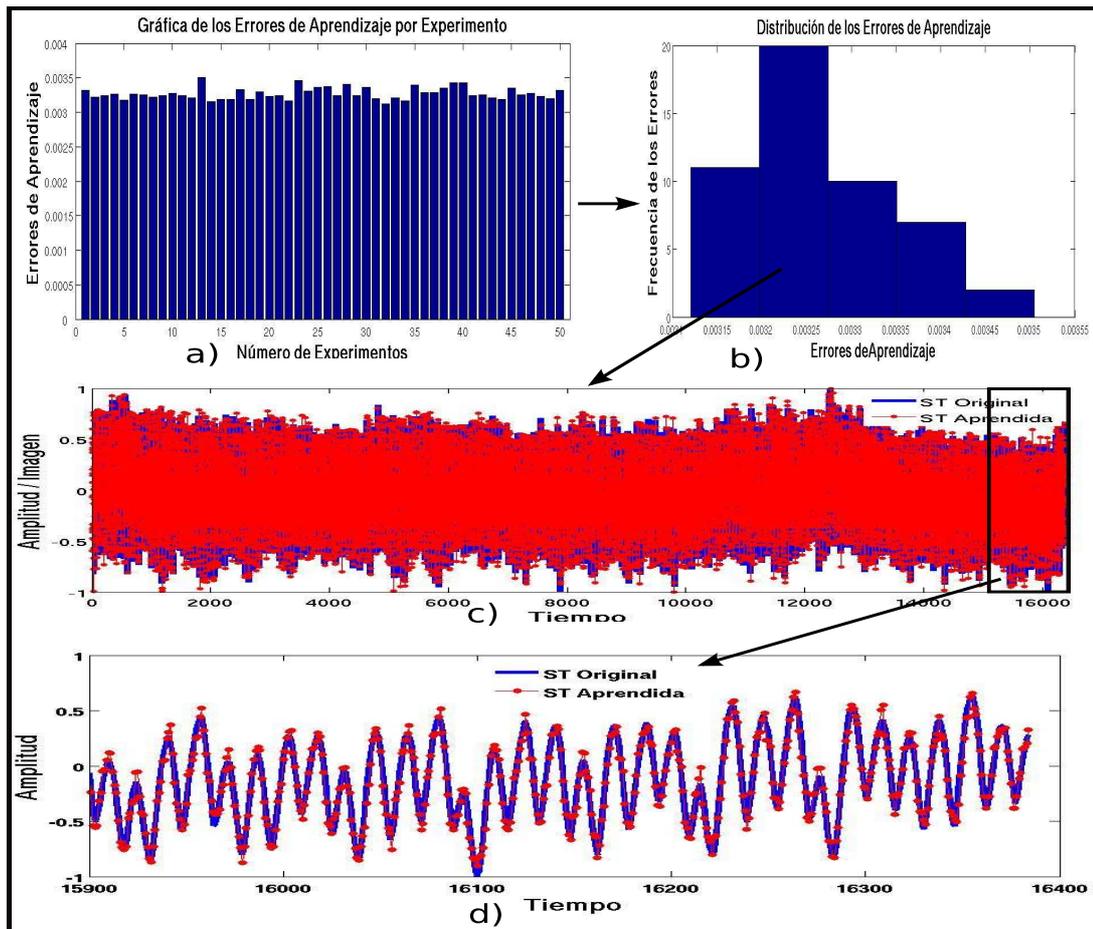


Figura 5.39: Resultados de la segunda etapa para la ST “Qperiodic3”, que tiene un comportamiento cuasi-periódico, en a) se observa los resultados de los errores de aprendizaje obtenidos para cada experimento, en b) observamos la distribución de estos errores, en c) se muestra la ST generada que obtuvo el error promedio y en d) se observan los últimos 400 puntos.

En la Figura 5.40, se observan los resultados de la ST “Lorenz” cuyo comportamiento dinámico es caótico y a pesar de su comportamiento dinámico, al ejecutar la técnica de aprendizaje, éste presentó un error dentro del rango de 0.035 y 0.047, el cual se observa es mayor que en la división anterior. El error promedio para esta ST es igual a 0.04047, y este resultado se observa en la Figura 5.40c). Además en la Figura 5.40b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es alta debido a su gráfica de distribución.

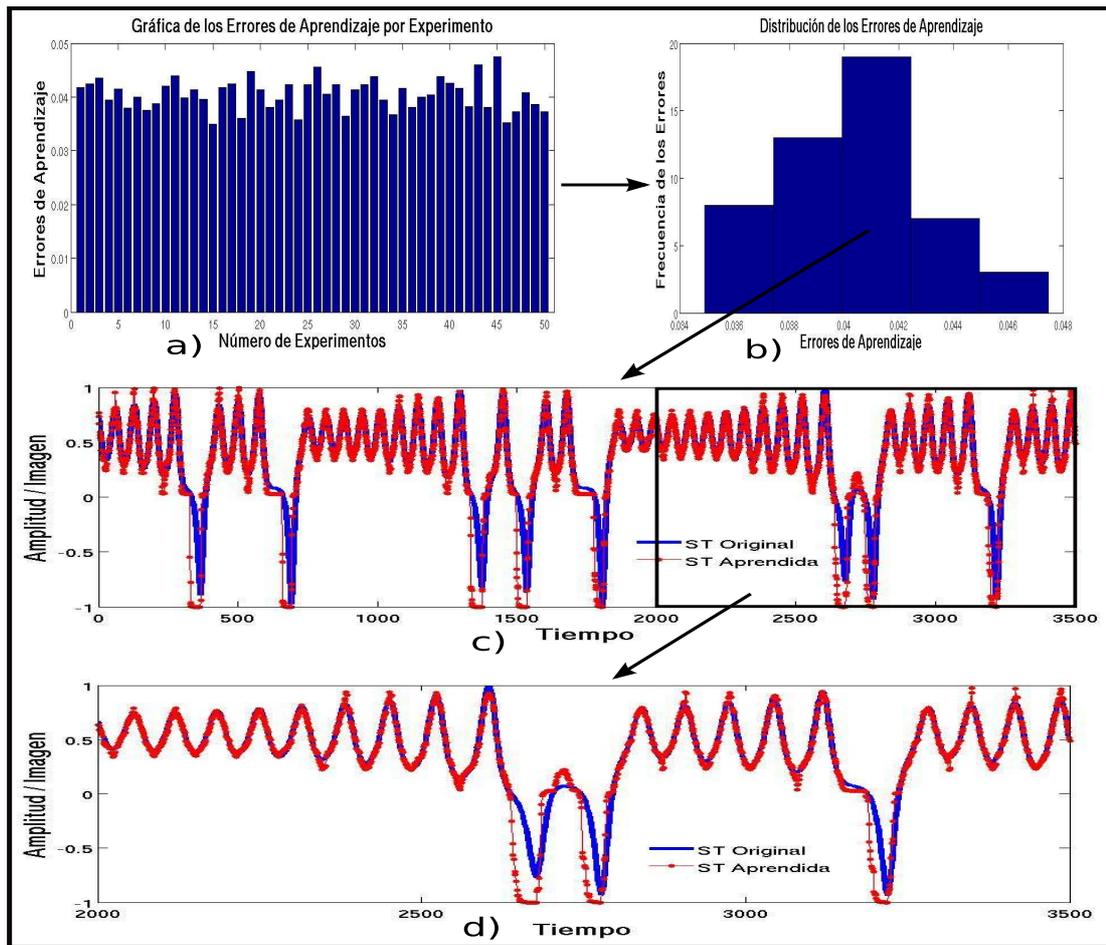


Figura 5.40: Resultado de la segunda etapa para la ST “Lorenz”, la cual tiene un comportamiento caótico, en a) se observa el resultado de los errores por experimento, en b) observamos la distribución de estos errores, en c) se muestra el resultado de la ST que obtuvo el error promedio y en d) se observan los últimos 1,500 puntos.

En la Figura 5.41, se observan los resultados de la ST “tent” cuyo comportamiento dinámico es caótico como la ST anterior. Esta ST presentó un error de aprendizaje (RMSE) dentro del rango de 0.0074 y 0.0089, el cual es mayor con respecto al tipo de división anterior. Ésta generó un error promedio (RMSE) igual a 0.00823, y se observa en la Figura 5.41c) y para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 100 puntos como se observa en la Figura 5.41d). Si observamos una vez más la Figura 5.41b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es buena debido a esta gráfica de distribución, aun cuando no es mejor que la división anterior.

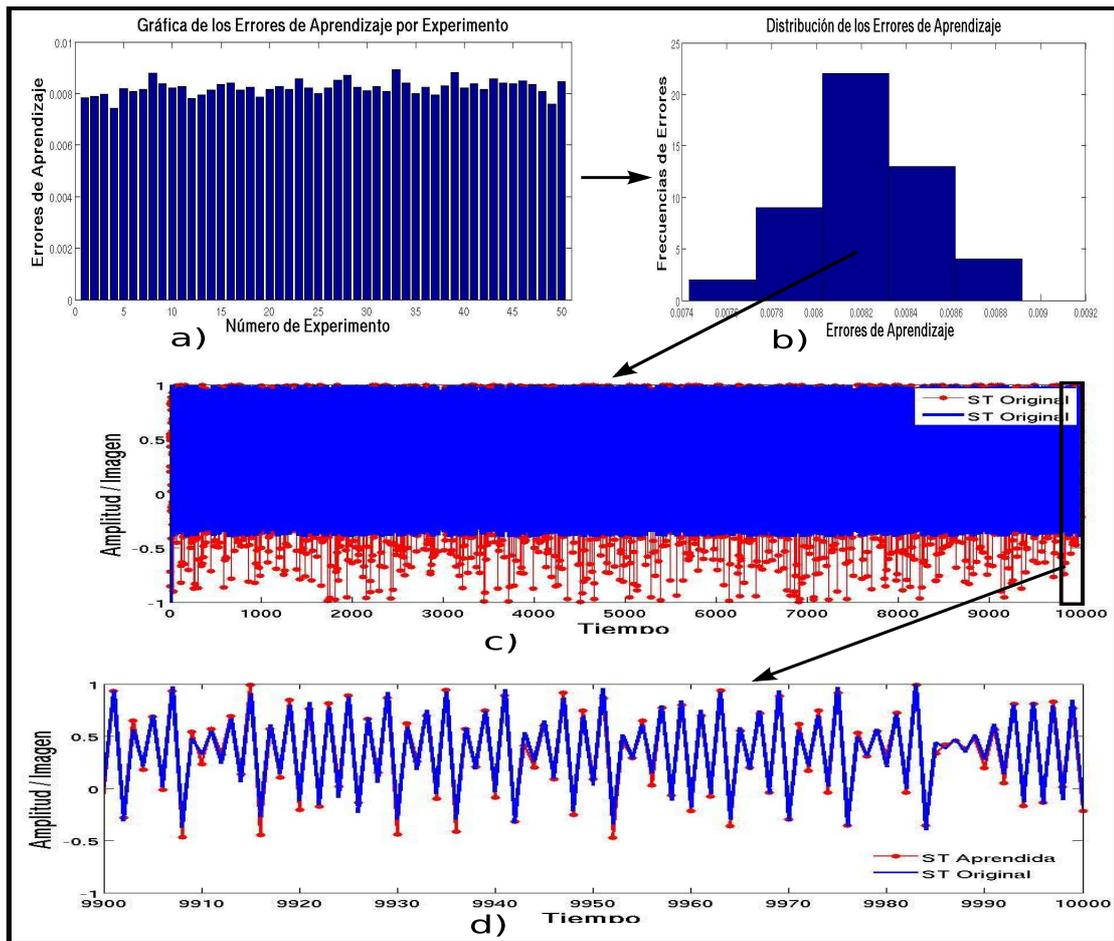


Figura 5.41: Resultado de la segunda etapa para la ST “tent”, la cual tiene un comportamiento caótico, en a) se observa el resultado del error de aprendizaje para cada experimento, en b) se observa la distribución de este error, en c) observamos la ST que obtiene el error promedio y en d) se muestran los últimos 100 puntos.

En la Figura 5.42, se observan los resultados de aplicar TABI a la ST “S&P 500” cuyo comportamiento dinámico es complejo. Ésta presenta un error de aprendizaje (RMSE) dentro del rango de 0.0125 y 0.021 y el error promedio (RMSE) es igual a 0.01686, que es el error que obtuvo durante el cuarto experimento de la técnica de aprendizaje y la ST que se genera de este experimento se observa en la Figura 5.42c) y para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 500 puntos como se observa en la Figura 5.42d).

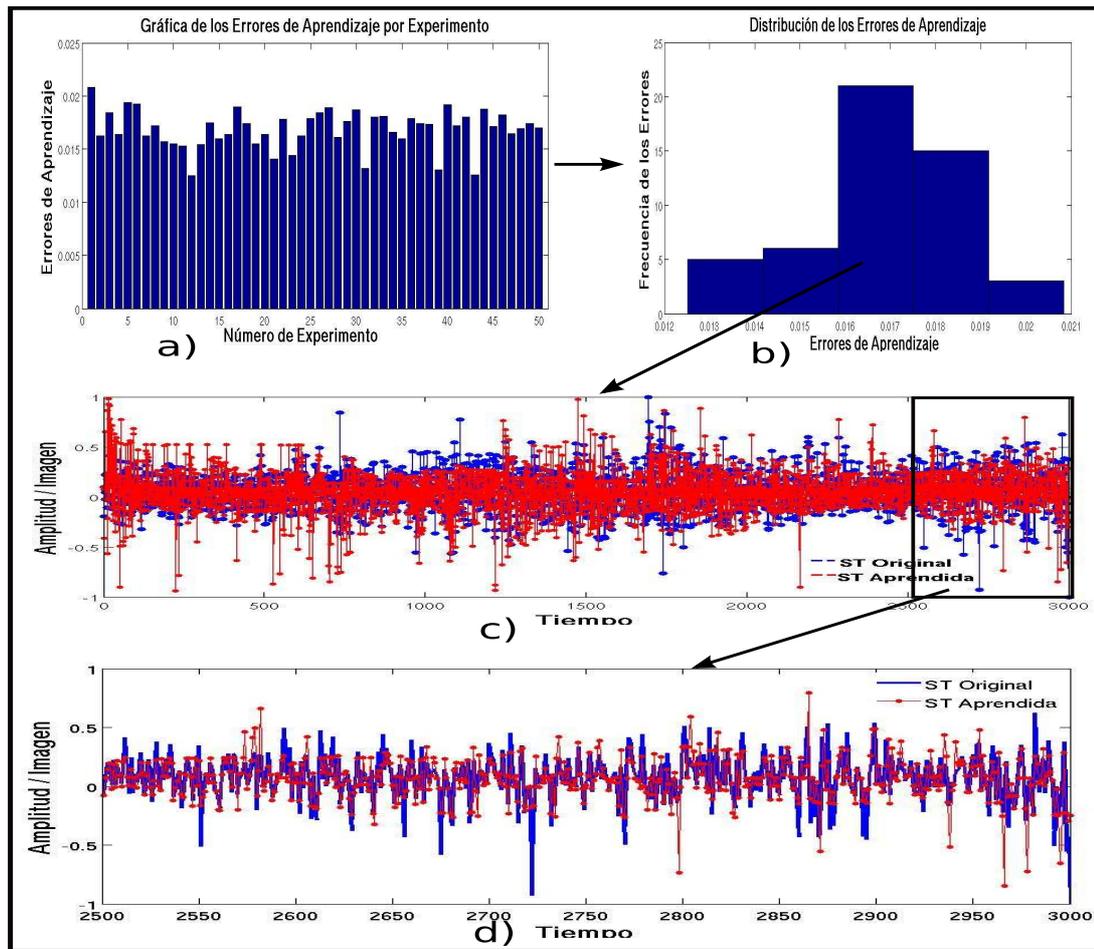


Figura 5.42: Resultado de la segunda etapa para la ST “S&P 500”, la cual presenta un comportamiento complejo, en a) se observa el resultado del error de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) se observa la ST que obtuvo el error promedio y en d) se observan los últimos 500 puntos.

En la Figura 5.37, se observan los resultados de aplicar TABI a la ST “Brownian Motion”, la cual presenta uno de los comportamientos dinámicos más complicados y nos referimos al estocástico y a pesar de ser uno de los comportamientos más complicados, al ejecutar la técnica de aprendizaje propuesta presentó un error de aprendizaje (RMSE) dentro del rango de 0.00525 y 0.0075, el cual es mayor que en la división anterior. Ésta genera un error promedio (RMSE) igual a 0.00592 y este error se obtuvo durante el tercer experimento de la técnica de aprendizaje propuesta y la ST que se genera de este experimento se observa en la Figura 5.37c) Si observamos una vez más la Figura 5.37b) es posible notar que existe una probabilidad muy alta de obtener un buen resultado de aprendizaje al volver a ejecutar la técnica de aprendizaje, debido a la gráfica de distribución para esta ST.

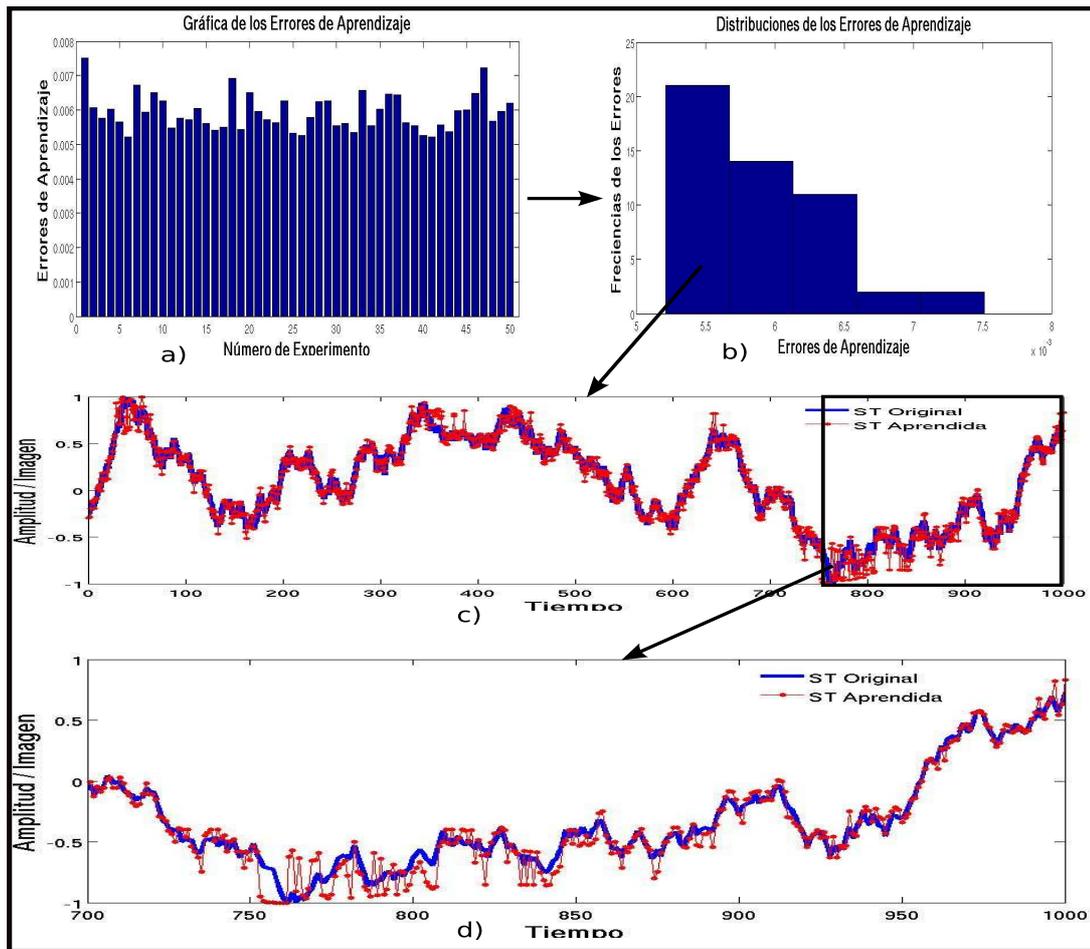


Figura 5.43: Resultado de la segunda etapa de la ST “Brownian Motion”, el cual presenta un comportamiento estocástico, en a) se observa el error de aprendizaje obtenido para cada experimento, en b) se observa la distribución de este error, en c) observamos la ST que obtuvo el error promedio y en d) observamos los últimos 300 puntos.

5.2.3. Tercera tipo: división por porcentaje de información total

Este tercer tipo división de la imagen de la serie de tiempo en un cierto número de *cajas*, se realiza de la siguiente manera: por un lado, se define el número de puntos dentro de cada *caja*, mediante la función área_f controlando que todas las *cajas* tengan igual número de información (puntos) y esto se realiza de dos maneras distintas:

1. Definiendo la función $\text{área}_f = 10\%$ del total de puntos de la series de tiempo.
2. Definiendo la función $\text{área}_f = 5\%$ del total de puntos de la series de tiempo.

Es por esta razón que este tipo de división se llama *división por porcentaje de información total*. Debido a que son dos maneras distintas para dividir la imagen de la ST a partir de su información total, mostraremos de manera consecutiva los resultados para ambas opciones.

A continuación se mostrarán algunos resultados obtenidos durante la aplicación de este tipo de división en las siguientes figuras. Estas figuras corresponden a las ST que fueron utilizadas y descritas en el paso anterior, esto con la finalidad de realizar una comparación en cuanto al rendimiento de este tipo de división respecto de los anteriores. Estas Figuras muestran la misma estructura que las anteriores. Para mostrar la diferencia entre la división al 10% y 5% del total de la información se procederá a mostrar ambos resultados de manera consecutiva.

Siguiendo con el orden de las series vistas en las divisiones anteriores y su correspondiente comportamiento, primero, observemos la Figura 5.44, la cual, muestra el resultado de ejecutar la división por porcentaje con el 10% de la información total de la ST “Seno”, la cual obtuvo un error de aprendizaje (RMSE) entre el rango de 0.0009 y 0.00205 y si lo comparamos con las divisiones anteriores podemos decir que éstos son mejores que en la segunda división. Además observamos que para esta ST se obtiene un error promedio igual a 0.00145. Entonces, es posible afirmar que el rendimiento de la técnica de aprendizaje para la serie “Seno” utilizando este tipo de división es bastante aproximado a la ST objetivo.

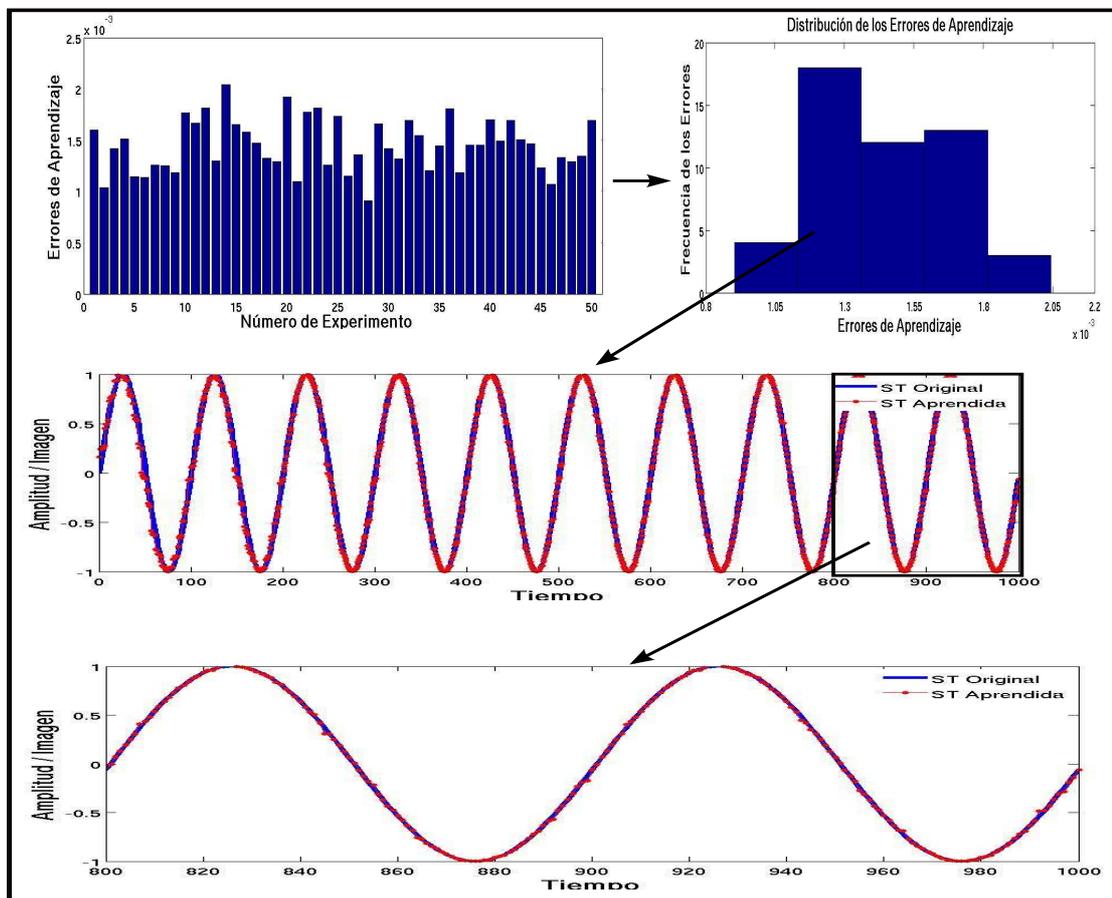


Figura 5.44: Resultado de la tercera etapa para el 10% de la información total, en a) se observan los resultados del error de aprendizaje obtenido de cada experimento, en b) se observan la distribución de estos errores, en c) se muestra la ST que obtiene el error de aprendizaje promedio y d) muestra los últimos 200 puntos.

La Figura 5.45 muestra el resultado de ejecutar la división por porcentaje pero ahora con el 5% de la información total de la ST “Seno”. Se observa que utilizando este tipo de división se presenta un error de aprendizaje (RMSE) entre el rango de 0.00025 y 0.0005, el cual si lo comparamos con todas las divisiones anteriores, ésta es la que arroja los mejores resultados. Ésta ST obtiene un error promedio (RMSE) igual a 0.00037. Si observamos una vez más la Figura 5.45b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un muy buen resultado es muy alta, debido a su gráfica de distribución.

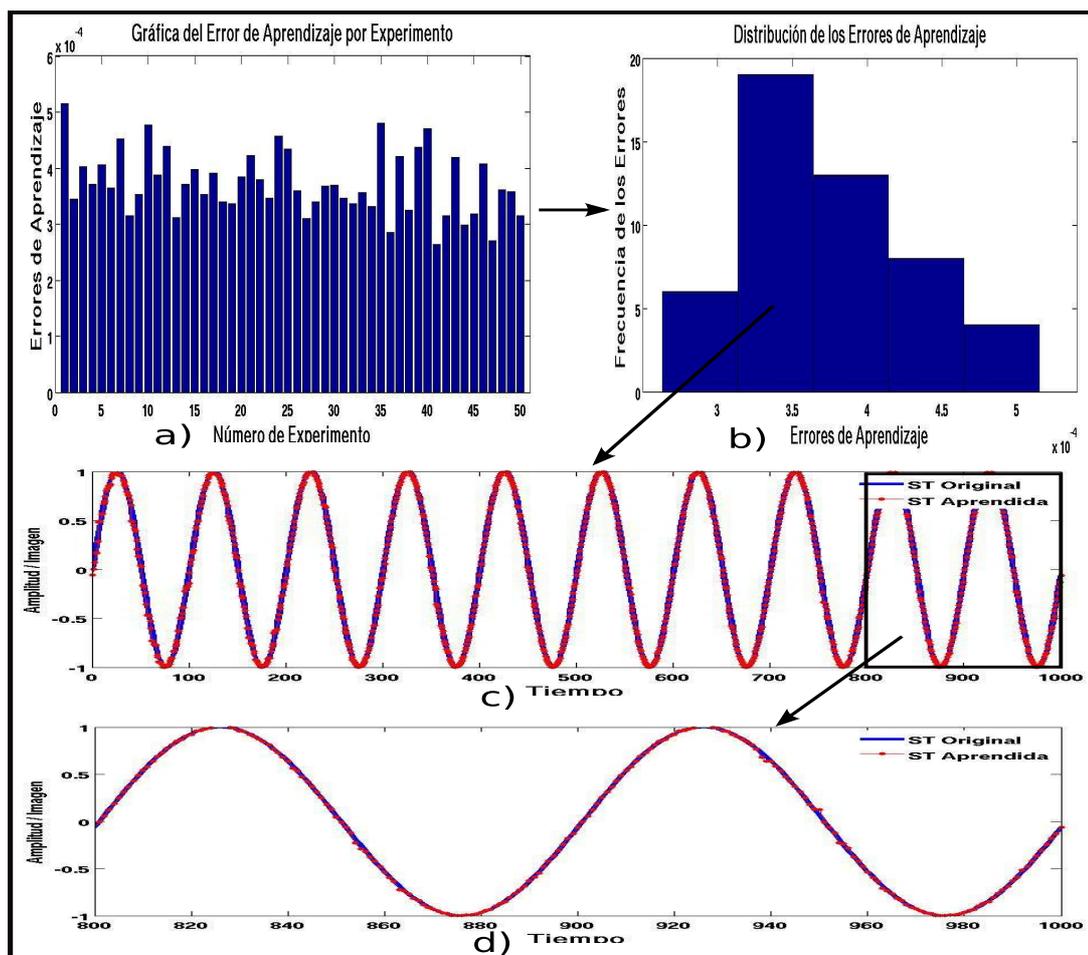


Figura 5.45: Resultado de la tercera etapa para el 5% de la información total, en a) se observan los resultados del error de aprendizaje obtenido de cada experimento, en b) se observan la distribución de estos errores, en c) se muestra la ST que obtiene el error de aprendizaje promedio y d) muestra los últimos 200 puntos.

En la Figura 5.46, se observan los resultados de aplicar esta tercera división para el 10 % del total de la información de la ST “Qperiodic3”. Utilizando este tipo de división, ésta presentó un error de aprendizaje (RMSE) dentro del rango de 0.00345 y 0.004. De igual manera se observa que la distribución de los errores de aprendizaje después de ejecutar el algoritmo propuesto 50 veces genera un error promedio de aprendizaje (RMSE) igual a 0.00367. Si observamos una vez más la Figura 5.46b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es muy alta debido a esta gráfica de distribución.

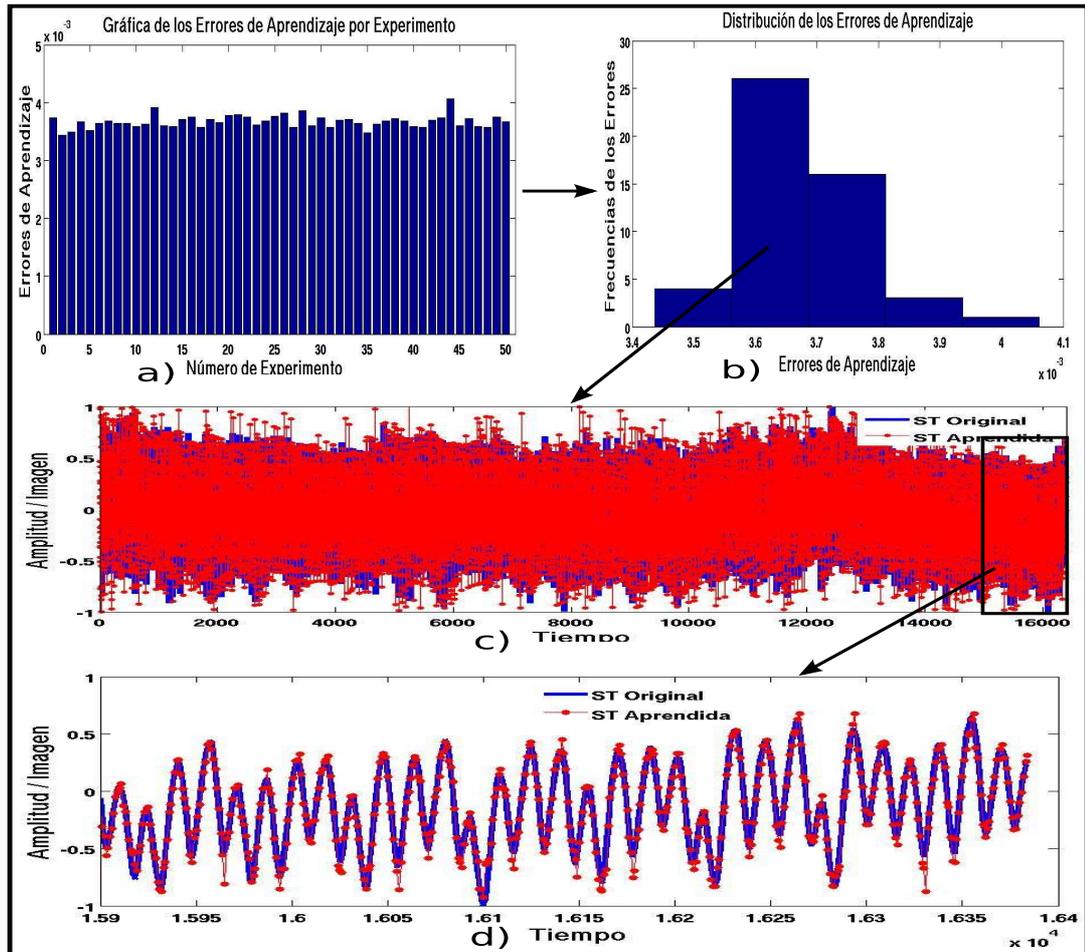


Figura 5.46: Resultado de la ST “Qperiodic3” durante la etapa de la división de la imagen con el 10 % del total de la información, en a) se observan los resultados de los errores de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) se observa la ST que generó el error promedio y en d) observamos los últimos 400 puntos.

En la Figura 5.47, se observan los resultados de la etapa tres pero ahora con el 5% del total de la información de la ST “Qperiodic3”. Esta sT presentó un error de aprendizaje (RMSE) dentro del rango de 0.00122 y 0.0014, el cual es muchísimo mejor que aplicar cualquiera de los tipos de división anterior. De igual manera se observa que la distribución de los errores de aprendizaje genera un error promedio de aprendizaje (RMSE) igual a 0.00132.

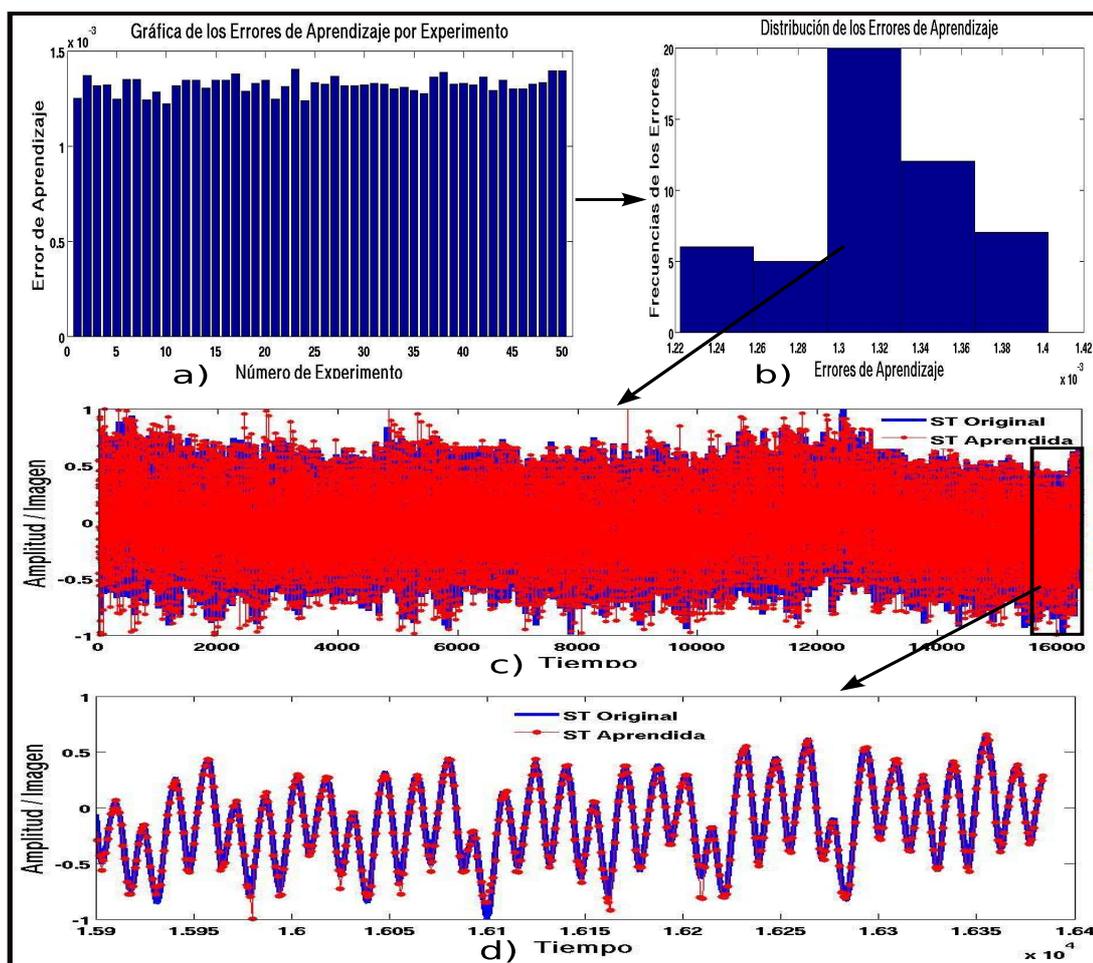


Figura 5.47: Resultados obtenidos durante la tercera etapa al dividir la imagen en al 5% de la información total, en a) observamos el resultado de los errores de aprendizaje durante cada experimento, en b) se muestra la distribución de estos puntos, en c) observamos la ST que obtuvo el error promedio y en d) se muestran los últimos 400 puntos.

En la Figura 5.48, se observan los resultados de aplicar la tercera división para el 10% de los puntos a la ST “Lorenz”. Ésta presenta un error de aprendizaje (RMSE) dentro del rango de 0.016 y 0.026, el cual se observa es menor que la segunda división, pero no que la primera. Además ésta genera un error promedio (RMSE) igual a 0.01946. Para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 1,500 puntos como se observa en la Figura 5.48d). Es posible observar que al ejecutar esta división, la precisión del aprendizaje no es tan buena como en la primera, pero si es mejor que la segunda.

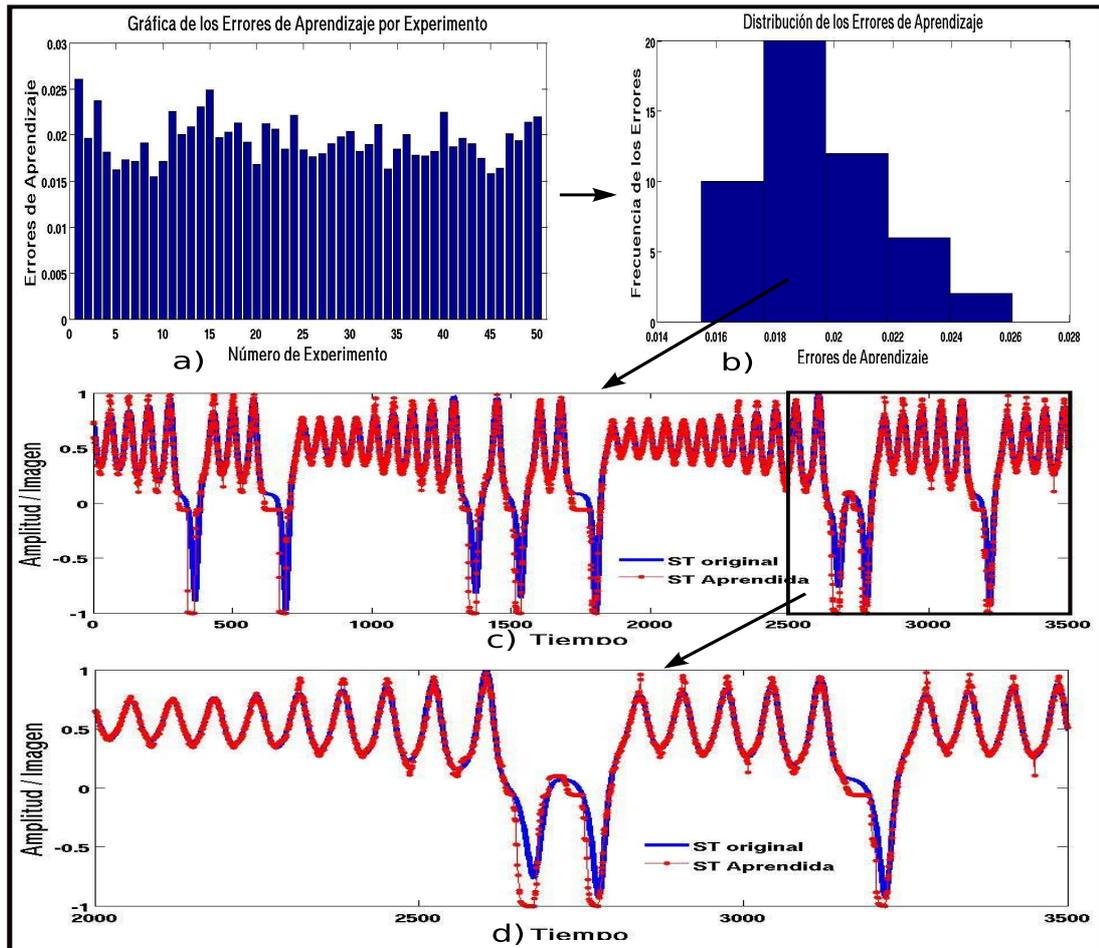


Figura 5.48: Resultado de aplicar la tercera etapa para el 10% del total de la información a la ST Lorenz, en a) se observa el resultado de los errores de aprendizaje obtenidos durante cada experimento, en b) se observa la distribución de estos errores, en c) se muestra la ST que obtuvo el error promedio y en d) se observan los últimos 1,500 puntos.

En la Figura 5.49, se observan los resultados de aplicar la tercera división ahora con el 5% de los puntos a la ST “Lorenz”. Ésta presentó un error de aprendizaje (RMSE) dentro del rango de 0.0018 y 0.032, el cual se observa es el mejor resultado comparado con todas las divisiones anteriores, donde se generó un error promedio (RMSE) igual a 0.00224. Si observamos una vez más la Figura 5.49b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es alta debido a su gráfica de distribución. Además es posible observar que al ejecutar esta división la precisión del aprendizaje es como ya comentamos mejor que las demás.

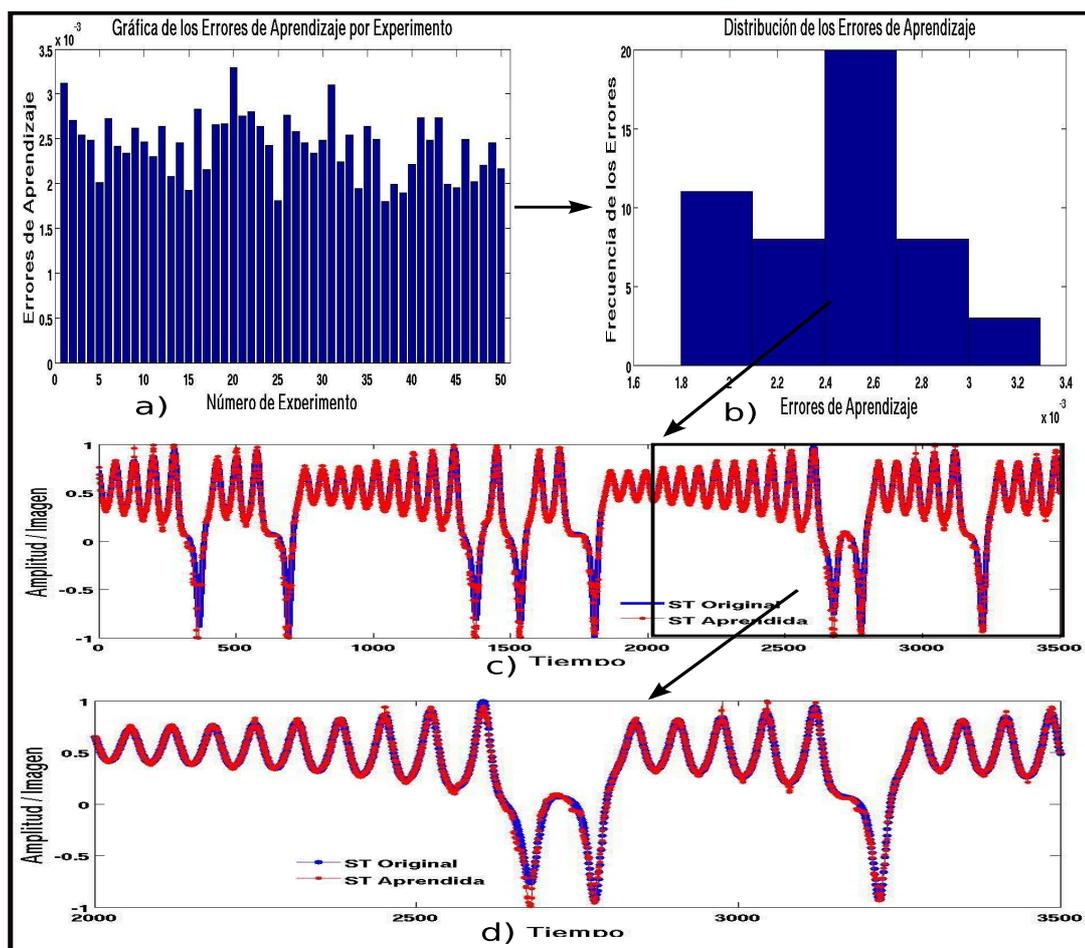


Figura 5.49: Restados obtenidos de la tercera etapa, pero con la división del 5% del total de la información, en a) se observan los errores de aprendizaje para cada experimento, en b) se observa la distribución de dichos errores, en c) se observa la ST que obtuvo el error promedio y en d) se muestran los últimos 1,500 puntos.

En la Figura 5.50, se observan los resultados de aplicar la TABI a la ST “tent”, el cual presenta un error de aprendizaje (RMSE) dentro del rango de 0.008 y 0.0091 y por lo tanto, se genera un error promedio (RMSE) igual a 0.00856, que es muy parecido al obtenido durante la segunda división. Para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 100 puntos como se observa en la Figura 5.50d).

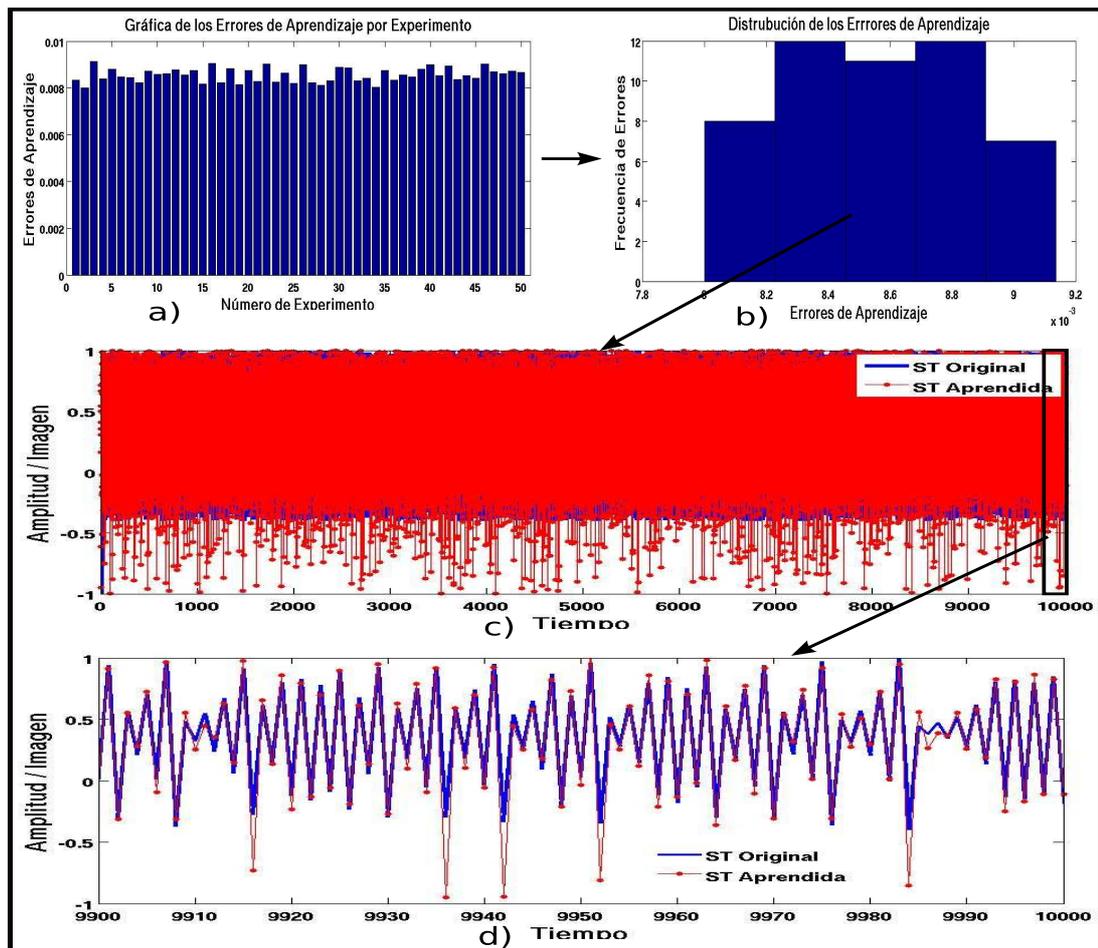


Figura 5.50: Resultado obtenido de aplicar la tercera etapa con el 10% del total de la información, en a) se muestran los errores de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) observamos la ST que obtuvo el error promedio y en d) se observan los últimos 100 puntos.

En la Figura 5.51, se observan los resultados de aplicar la técnica TABI a la ST “tent” pero ahora con el 5% del total de la información y a pesar de su comportamiento dinámico, al ejecutar la técnica de aprendizaje se presentó un error dentro del rango de 0.0025 y 0.0034, el cual es menor con respecto a las divisiones anteriores. Ésta genera un error promedio (RMSE) igual a 0.00296 y si observamos una vez más la Figura 5.51b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un muy buen resultado es alta.

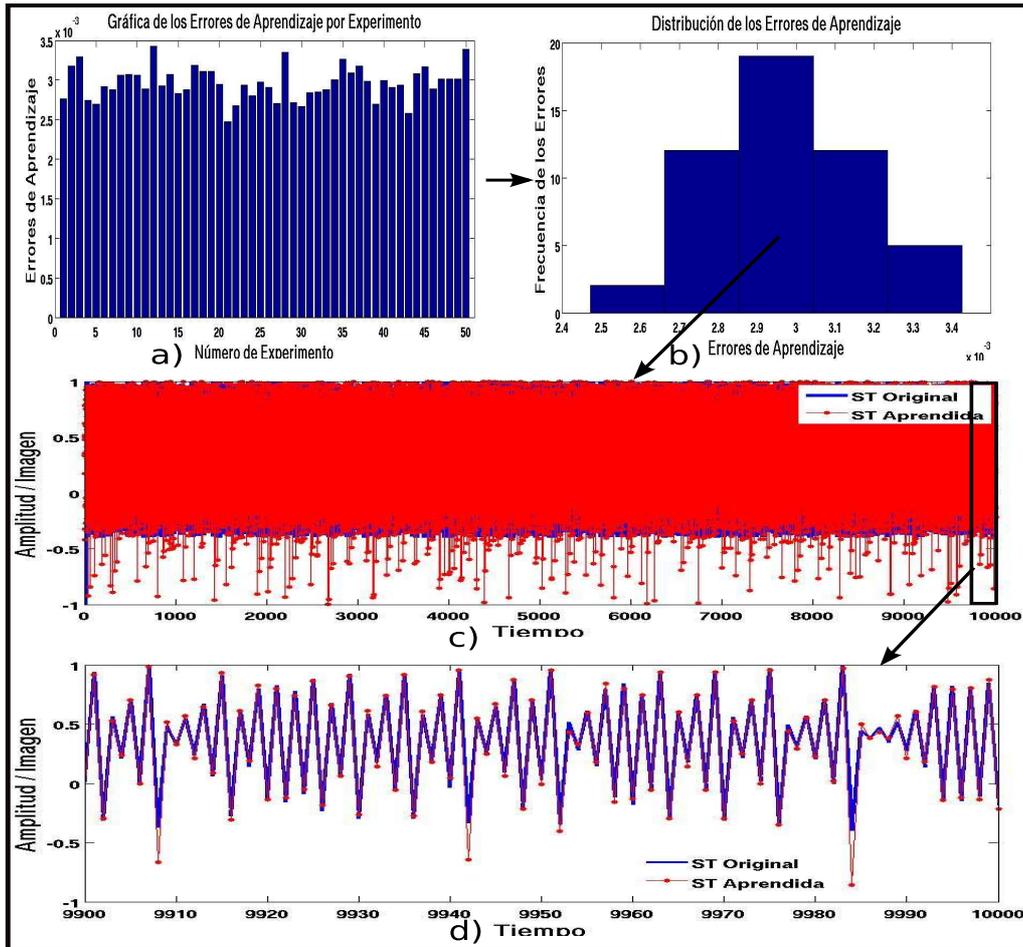


Figura 5.51: Resultado obtenido de aplicar la tercera etapa con el 5% del total de la información, en a) se muestran los errores de aprendizaje para cada experimento, en b) se observa la distribución de estos errores, en c) observamos la ST que obtuvo el error promedio y en d) se observan los últimos 100 puntos.

En la Figura 5.52, se observan los resultados de aplicar TABI a la ST “S&P 500”, el cual, presentó un error de aprendizaje (RMSE) dentro del rango de 0.0054 y 0.0095, por lo tanto, ésta genera un error promedio (RMSE) igual a 0.00678. Para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 500 puntos como se observa en la Figura 5.52d).

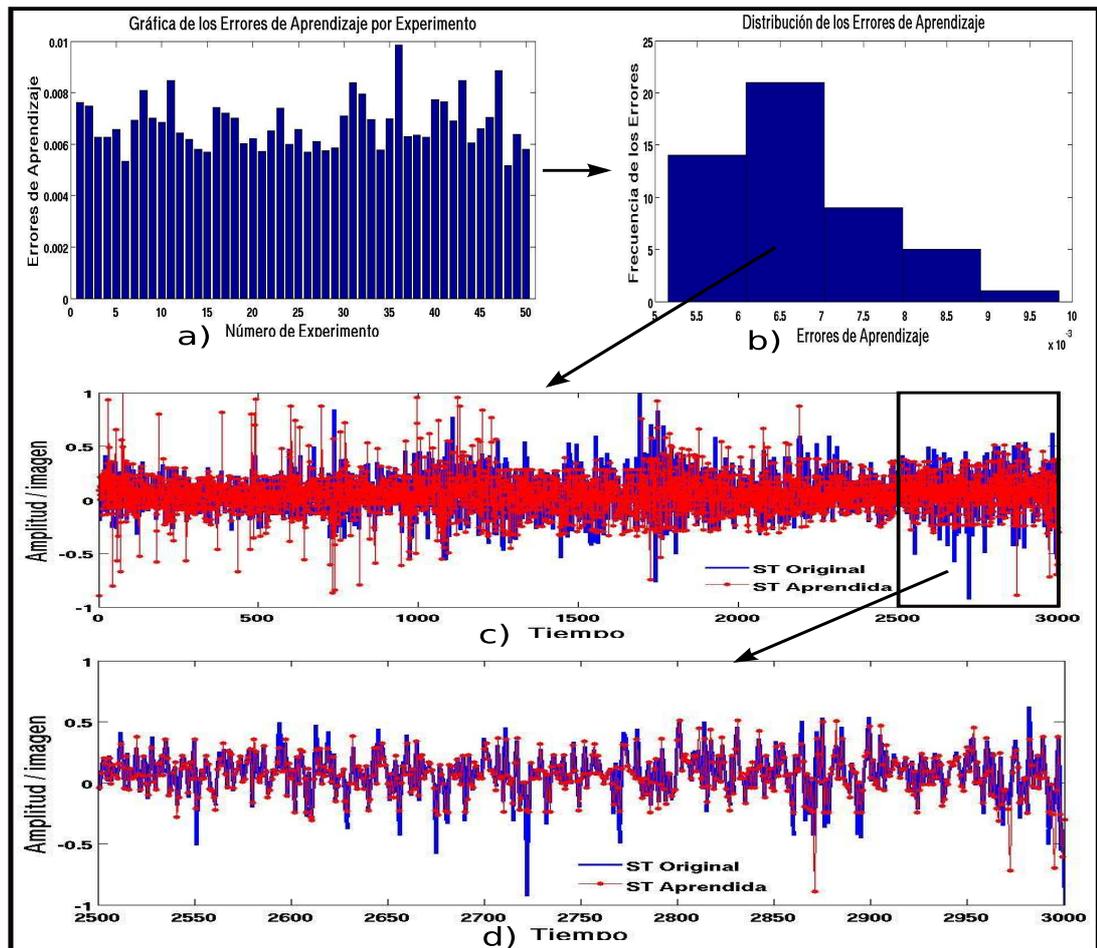


Figura 5.52: Resultado obtenido de la tercera etapa para el 10 % de la información de la ST S&P 500, en a) se observan los resultados de los errores de aprendizaje obtenidos durante cada experimento, en b) se observa la distribución de estos errores, en c) se muestra la ST que obtuvo el error de aprendizaje promedio y d) muestra los últimos 500 puntos.

En la Figura 5.53, se observan los resultados de aplicar el tercera tipo de división ahora con el 5 % del total de la información a la ST “S&P 500” la cual, presentó un error de aprendizaje (RMSE) dentro del rango de 0.0025 y 0.005. Se obtuvo un error promedio (RMSE) igual a 0.00360, que si observamos los resultados de las divisiones anteriores, es posible notar que este error es el más bajo obtenido para esta ST y ésta se observa en la Figura 5.53. Si observamos una vez más la Figura 5.53b) es posible notar de que la probabilidad de que al volver a ejecutar la técnica de aprendizaje obtengamos un buen resultado es muy alta.

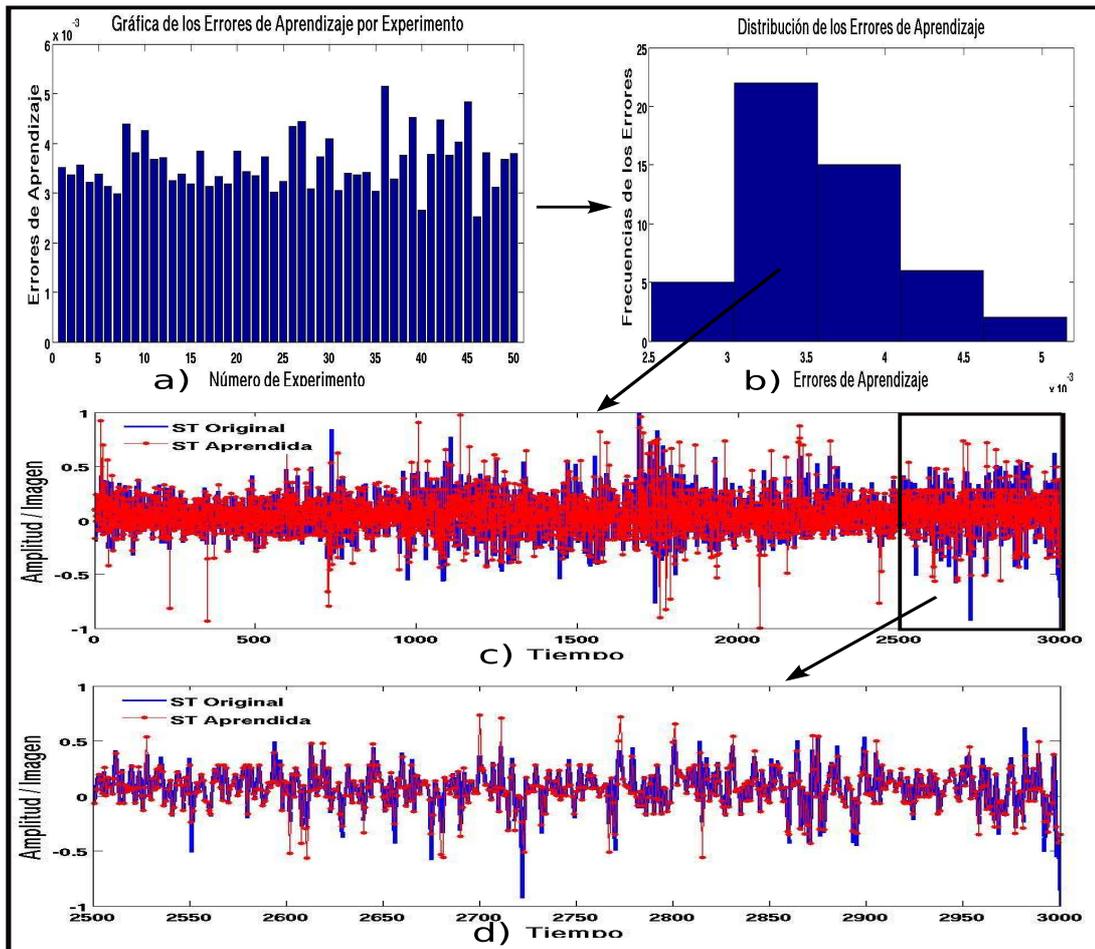


Figura 5.53: Resultado obtenido de la tercera etapa para el 5% de la información de la ST S&P 500, en a) se observan los resultados de los errores de aprendizaje obtenidos durante cada experimento, en b) se observa la distribución de estos errores, en c) se muestra la ST que obtuvo el error de aprendizaje promedio y d) muestra los últimos 500 puntos.

En la Figura 5.54, se observan los resultados de aplicar TABI a la ST “Brownian Motion”, la cual presenta uno de los comportamientos dinámicos más complicados y apesar de eso presentó un error de aprendizaje (RMSE) dentro del rango de 0.005 y 0.0066. Además se obtuvo un error promedio (RMSE) igual a 0.00582 y para observar a mayor detalle este proceso de aprendizaje se realizó un acercamiento de los últimos 300 puntos como se observa en la Figura 5.54d).

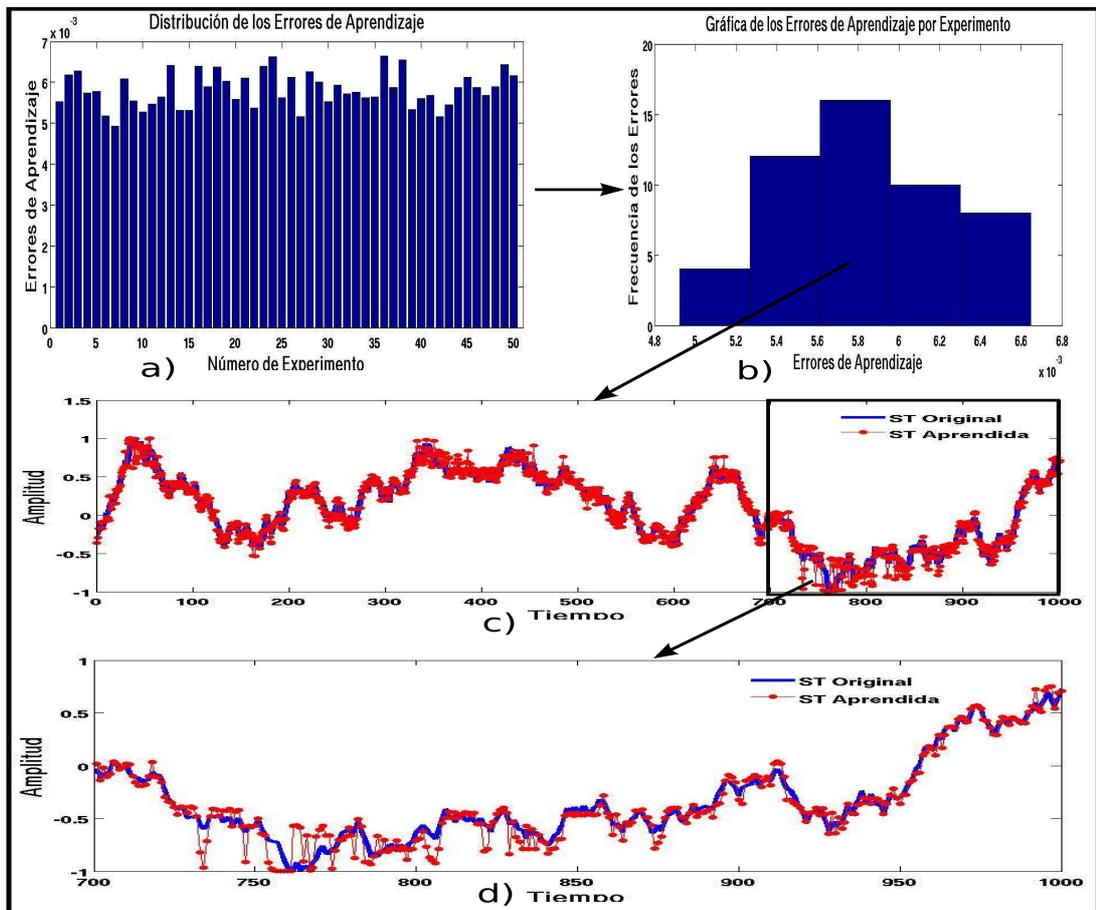


Figura 5.54: Resultado de la tercera etapa para el 10 % del total de la ST Brownian Motion, en a) se observan los resultados de los experimentos obtenidos para cada experimento, en b) se observa la distribución de estos, en c) se observa la ST que obtuvo el error de aprendizaje promedio y en d) se muestran los últimos 300 puntos.

En la Figura 5.37, se observan los resultados de aplicar TABI utilizando la metodología de la tercera división ahora con el 5% del total de la información a la ST “Brownian Motion”, la cual presenta un error de aprendizaje (RMSE) dentro del rango de 0.0013 y 0.002, y se observa que cuantitativamente éste es mejor, no por mucho, que los obtenidos durante las demás divisiones, generando un error promedio (RMSE) igual a 0.00168.

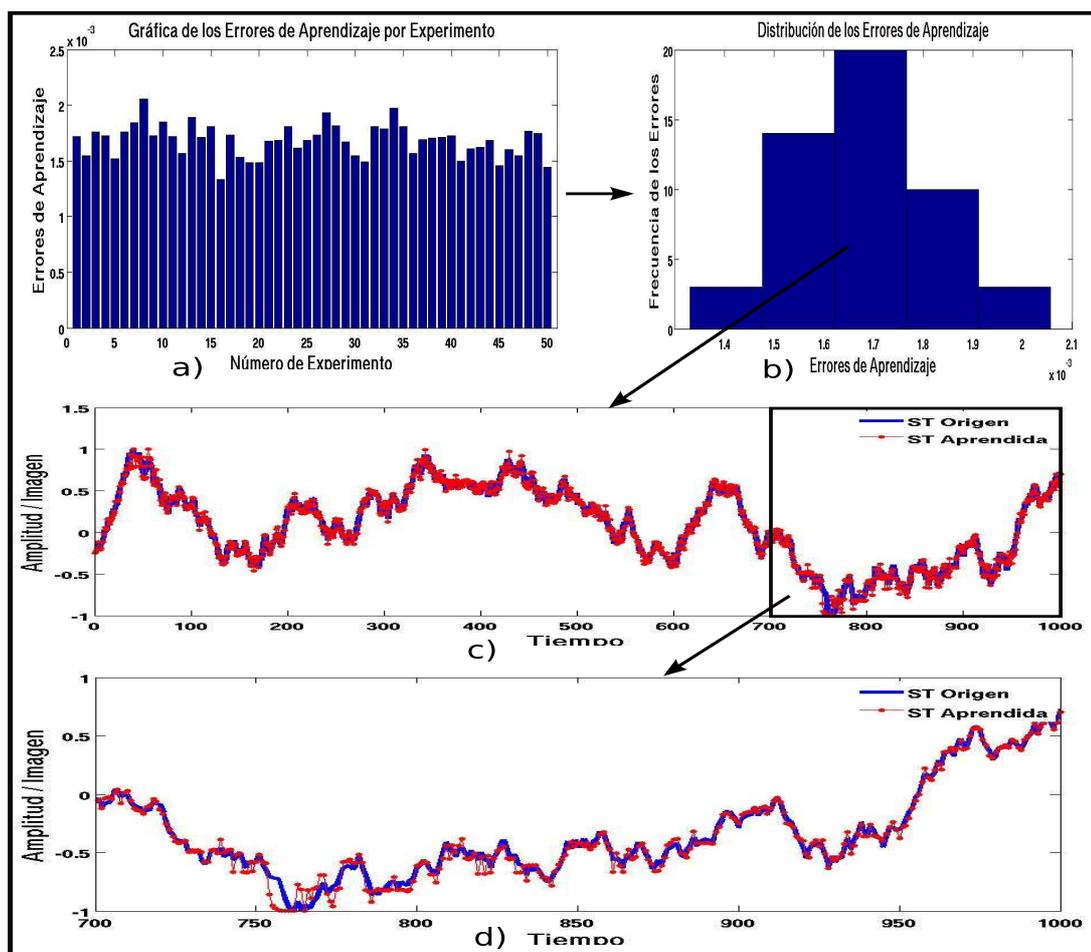


Figura 5.55: Resultado de la tercera etapa para el 5% del total de la ST Brownian Motion, en a) se observan los resultados de los experimentos obtenidos para cada experimento, en b) se observa la distribución de estos, en c) se observa la ST que obtuvo el error de aprendizaje promedio y en d) se muestran los últimos 300 puntos.

5.3. Análisis de los resultados del seguimiento de ST

Para este proceso de aprendizaje, es posible resumir los resultados promedios obtenidos de los 50 experimentos realizados para cada ST del conjunto de prueba, en la siguiente Tabla 5.2.

Cuadro 5.2: Resultados del proceso de aprendizaje para cada tipo de división.

N°	Serie de Tiempo	Tipo de División			
		MismaAnchura	Frecuencia	10 % del total	5 % del total
1	Seno	0.00151	0.00351	0.00145	0.00037
2	Vanderpol	0.00456	0.00708	0.00562	0.00255
3	Qperiodic2	0.00216	0.00237	0.00238	0.00068
4	Qperiodic3	0.00355	0.00327	0.00367	0.00132
5	Mackey & Glass	0.00508	0.00546	0.00545	0.00150
6	Logistic	0.00652	0.01816	0.00854	0.00224
7	Lorenz	0.00273	0.04047	0.01946	0.00243
8	Rossler	0.00266	0.00320	0.00283	0.00096
9	Ikeda	0.00538	0.00550	0.00544	0.00175
10	Henon	0.00616	0.01046	0.00765	0.00186
11	Tent	0.00645	0.00823	0.00856	0.00296
12	A1	0.00299	0.01607	0.00892	0.00446
13	D1	0.00406	0.00544	0.00380	0.00136
14	Laser	0.00415	0.01621	0.00888	0.00327
15	El niño	0.00437	0.00548	0.00515	0.00169
16	Lovaina	0.00231	0.00393	0.00378	0.00185
17	S&P 500	0.00462	0.01686	0.00678	0.00360
18	Star	0.00503	0.00954	0.00648	0.00239
19	Flujo Rio Jokulsa	0.00345	0.03885	0.01286	0.00599
20	Brownian Motion	0.00418	0.00592	0.00582	0.00168
21	White Noise	0.00603	0.00641	0.00643	0.00143
22	Precio del Oro	0.00338	0.00803	0.00761	0.00509
23	Rio Colorado	0.00367	0.05955	0.01861	0.00524
24	Lago Erie	0.00423	0.00543	0.00456	0.00164
25	Rio Nilo	0.00538	0.02303	0.00885	0.00435
26	Manchas Solares	0.00438	0.01333	0.01004	0.00356
27	Temp. Max. Melbourne	0.00463	0.00737	0.00626	0.00245
28	Temp. Min. Melbourne	0.00487	0.00867	0.00573	0.00295
29	Souther Oscilation	0.00499	0.01312	0.00796	0.00371

En la Tabla 5.2, se observa que la primera columna corresponde al número de ST, en la segunda columna se observa el nombre de la ST a analizar, en la tercera columna se observa la división obtenida del primer tipo de división, esto es, mediante la división de la imagen con respecto a la anchura, posteriormente se observa la división de la imagen concerniente al segundo tipo de división (división por frecuencias), en la columna cinco se observa la división

de la imagen en el 10 % del total de la información para cada caja (tercer tipo de división) y finalmente, en la columna seis, se observa la división de la imagen en el 5 % del total de la información para cada caja.

Además, es posible observar en la Tabla 5.2 que los tipos de división de la imagen que arrojaron los mejores resultados para el aprendizaje de ST fueron para el caso de las ST, Santa fe, Flujo del río Jokulsa, Precio del Oro, Río colorado, la primera división, esto es dividir la imagen de la ST en la misma anchura entre cada caja. Por otro lado, en el caso de las demás series de tiempo se obtuvo un mejor resultado al aplicar la tercera división con la opción de dividir la imagen de la ST en el 5 % del total de la información dentro de cada caja. Los resultados anteriores se observan resaltadas en negritas en la Tabla 5.2. Es remarcable mencionar que aun cuando los resultados que se muestran en dicha tabla son los promedios obtenidos de ejecutar TABI propuesta 50 veces, éstos resultan muy acertados para el caso del aprendizaje de ST con distintos comportamientos dinámicos.

Ahora analicemos estos resultados, la siguiente manera:

1. Analizamos el número de puntos dentro de cada caja, pues es posible notar que este número de puntos varía con respecto al tipo de división de la imagen que se emplee, de acuerdo a lo siguiente:
 - Recordemos que el primer tipo de división de la imagen, la cual llamamos *división por misma anchura*, define el intervalo de las cajas de acuerdo a alguna anchura previamente establecida, es posible notar que la división de la imagen utilizando este método genera intervalos de caja estáticos, es decir, son los mismos intervalos de caja, aun cuando estemos analizando alguna ST con diferente comportamiento dinámico. De acuerdo a eso, el número de puntos dentro de cada caja es distinto para cada ST y como es de esperar para cada caja también. Debido a que, como comentamos anteriormente, el número de puntos dentro de cada caja es variado, se procedió a calcular el promedio de los puntos dentro de cada caja, esto con el fin de poder realizar alguna comparación con las demás divisiones.
 - Como se comentó en los capítulos anteriores, el segundo tipo de división, la cual se llamó *división por frecuencias*, divide la imagen de la ST de acuerdo al promedio de la frecuencia de los puntos dentro de cada caja como su nombre lo indica y la diferencia de esta división con respecto a la anterior es que ésta genera de manera dinámica los intervalos de las cajas, en donde además la distribución de los puntos dentro de cada caja es uniforme. Estas mismas características la presenta la tercera división, la cual recordemos divide la imagen de la ST en cajas en donde el número de puntos que pertenecen dentro de éstas contiene el 10 % o el 5 % del total de puntos de la ST (información).
2. Posteriormente, si observamos nuevamente los resultados promedios de la Tabla 5.2, donde se muestran de manera resaltada los mejores resultados durante el proceso de aprendizaje y si procederemos a compararlos con la Tabla 5.3 en donde se observa el número total de puntos entrenados para cada ST. Además, se observan los puntos que se encuentran dentro de cada caja de acuerdo al tipo de división que se empleó, entonces, esta última tabla nos va a ayudar a entender el porqué la técnica TABI funciona mejor con algún tipo de división.

Cuadro 5.3: Se observa el número total de puntos analizados durante este proceso de aprendizaje para cada ST. Además, se observa el número de puntos dentro de cada caja según la etapa empleada. Como se comentó, el resultado de la primera división correspondiente a la misma anchura genera cajas con distinto número de puntos, es por ello que se muestra el promedio que se calculó de los puntos dentro de cada caja.

N°	Serie de Tiempo	TotalPuntos	Tipo de División			
			MismaAnch.	Frec.	10 %	5 %
1	Seno	1,000	90	140	100	57
2	Vanderpol	2,000	199	259	195	99
3	Qperiodic2	16,384	1638	1638	1638	819
4	Qperiodic3	16,384	1638	1479	1663	831
5	Mackey & Glass	2,000	199	211	197	102
6	Logistic	1,000	99	145	98	47
7	Lorenz	3,500	350	469	354	171
8	Rossler	5,000	499	483	501	255
9	Ikedda	5,000	499	438	499	249
10	Henon	5,000	499	582	499	247
11	Tent	10,000	999	960	999	495
12	A1	25,000	2497	3380	2435	1364
13	D1	50,000	4999	5258	4979	2492
14	Laser	5,000	499	887	496	237
15	El niño	2,000	199	195	196	96
16	Lovaina	2,000	199	176	201	102
17	S&P 500	3,000	299	712	295	152
18	Star	2,000	199	236	195	105
19	Flujo Rio Jokulsa	1,096	109	299	112	57
20	Brownian Motion	1,000	99	95	95	48
21	White Noise	1,000	99	99	99	47
22	Precio del Oro	1,074	107	133	107	55
23	Rio Colorado	744	73	247	75	41
24	Lago Erie	600	59	61	55	31
25	Rio Nilo	1,297	129	179	130	61
26	Manchas Solares	2,739	273	483	271	139
27	Temp. Max. Melbourne	3,650	364	475	356	176
28	Temp. Min. Melbourne	3,650	364	450	361	172
29	Souther Oscilation	1,235	123	172	123	61

Analizando la Tabla 5.2 con respecto a la Tabla 5.3 nos podemos dar cuenta que las ST que presentan un mejor resultado de aprendizaje (ver Tabla 5.2, columna 6) son aquellas que presentan el menor número de puntos dentro de cada caja como es el caso de la tercera división en donde el número de puntos dentro de cada caja es igual al 5 % del total de los puntos de la ST (ver Tabla 5.3, columna 7). Esta observación es importante debido a que el mejor resultado de aprendizaje está en función del número de puntos que se necesitan aprender dentro de cada caja y como pudiera parecer lógico mientras menos puntos o patrones se tengan que aprender, mejor será este aprendizaje, lo cual como comentamos se cumple en este caso.

Ahora analizaremos las ST que obtuvieron los mejores resultados del aprendizaje con la primera división, la cual si observamos nuevamente la Tabla 5.3 los puntos promedios son mayores que los puntos que se obtienen con la tercera división con el 5 % del total de puntos. Por ejemplo, la ST Santa fe, la cual tiene para la primera división en promedio 2,497 puntos dentro de cada caja y como podemos observa en la Figura 5.56, este promedio es válido para las cajas 2, 3, 4 y posiblemente la 5.

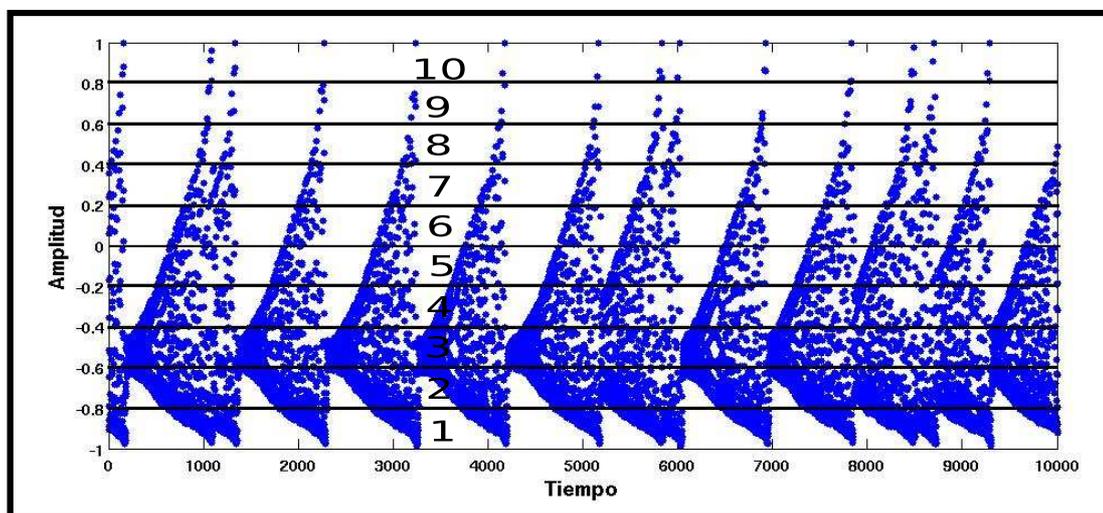


Figura 5.56: División de la imagen de la ST “Santa fe”, en 10 cajas con la misma anchura (primera etapa).

Además, en la Figura 5.56 es posible observar que la distribución de los primeros mil puntos de esta ST, que se encuentran dentro de cada caja, no son uniformes y por lo tanto como explicamos en la sección 3.2.2, el proceso de aprendizaje es un poco más complicado, debido a la saturación del número de puntos dentro de cada caja como por ejemplo en las caja 2, 3 y posiblemente la caja 4 que se observan en la Figura 5.56, en donde el número de puntos dentro de estas cajas es mayor que la demás cajas.

Por otro lado, si observamos la Figura 5.57, la cual nos muestra el número de puntos que se encuentran dentro del intervalo de cada caja previamente definido al momento de ejecutar nuestro algoritmo entonces, es posible notar que las tres primeras cajas contienen hasta tres veces más puntos que el promedio y donde además dentro de dichas cajas no se observan cambios tan bruscos² (ver, Figura 5.56), la caja 4 contiene el número de puntos casi igual al

²Con un cambio brusco nos referimos a que de repente el sentido de los puntos cambie.

promedio, pero si observamos las siguientes cajas, esto es, las cajas 5 y 6 éstas contienen la mitad de los puntos promedio y finalmente, las últimas cuatro cajas contienen menos de la cuarta parte de los puntos promedios.

N° Caja	Intervalo	N° Puntos
caja 1	(-1.0001 -0.8000]	6699.0000
caja 2	(-0.8000 -0.6000]	6502.0000
caja 3	(-0.6000 -0.4000]	4824.0000
caja 4	(-0.4000 -0.2000]	2679.0000
caja 5	(-0.2000 0.0000]	1742.0000
caja 6	(0.0000 0.2000]	1257.0000
caja 7	(0.2000 0.4000]	740.0000
caja 8	(0.4000 0.6000]	333.0000
caja 9	(0.6000 0.8000]	129.0000
caja 10	(0.8000 1.0000]	66.0000
Promedio:		2497.1000

Figura 5.57: Resultado de dividir los puntos de la ST “Santa fe” en 10 cajas, las cuales se observan en la primera columna de esta tabla, en la segunda columna se observan los intervalos definidos para cada caja y finalmente en la tercera columna observamos los puntos que pertenecen dentro de cada caja.

Por lo que podríamos concluir que, para esta ST los puntos se acumulan básicamente en las primeras tres cajas. Y esta división, es lo que hace que la técnica funcione mejor, pero como vemos en la Tabla 5.3 únicamente funciona para algunas ST, en este caso en específico para 4 de un total de 29 ST analizadas con la técnica TABI. Otro punto importante que es necesario tomar en cuenta, es que si observamos nuevamente la Tabla 5.2 podemos notar que las primeras tres divisiones de la imagen de la ST tienen resultados parecidos, pero si fuera necesario escoger entre alguna de estas tres divisiones lo más recomendable sería la tercera, es decir, dividir la imagen de la ST en el 10% del total de los puntos.

Es importante mencionar que la información que se obtiene del proceso de aprendizaje, sólo representa la información local de la serie de tiempo, es decir, es la información que se obtiene únicamente de cada *caja* de la serie de tiempo. Esta información local es posible utilizarla para el problema de predicción de series de tiempo, el cual analizaremos a continuación.

5.4. Resultados de la predicción

La predicción de series de tiempo ha sido una de las aplicación más utilizadas en el área del aprendizaje, especialmente en las ST debido a las múltiples aplicaciones que éste conlleva en varias áreas tales como: economía, control, medicina, entre otras.

Como se comentó en la sección 2.3, la predicción de una ST $x(t_1), x(t_2), \dots, x(t_n)$ se define como el proceso de estimar un valor $x(t_{n+1})$ o varios valores futuros $x(t_{n+h})$. Recordemos que el tipo de predicción que se va a manejar en este trabajo sigue la filosofía de la *predicción iterada* (ver sección 2.3.1), la cual, predice un punto adelante y esta predicción es utilizada como parte

del pasado para predecir el siguiente punto y así sucesivamente hasta tener h predicciones. La diferencia entre la predicción iterada clásica con respecto a la *predicción iterada multipunto* (que se emplea en este trabajo), como su nombre lo indica, los múltiples puntos que se pueden predecir, en una predicción, en vez de predecir de manera iterada solo un punto. Lo anterior se observa en la Figura 5.58.

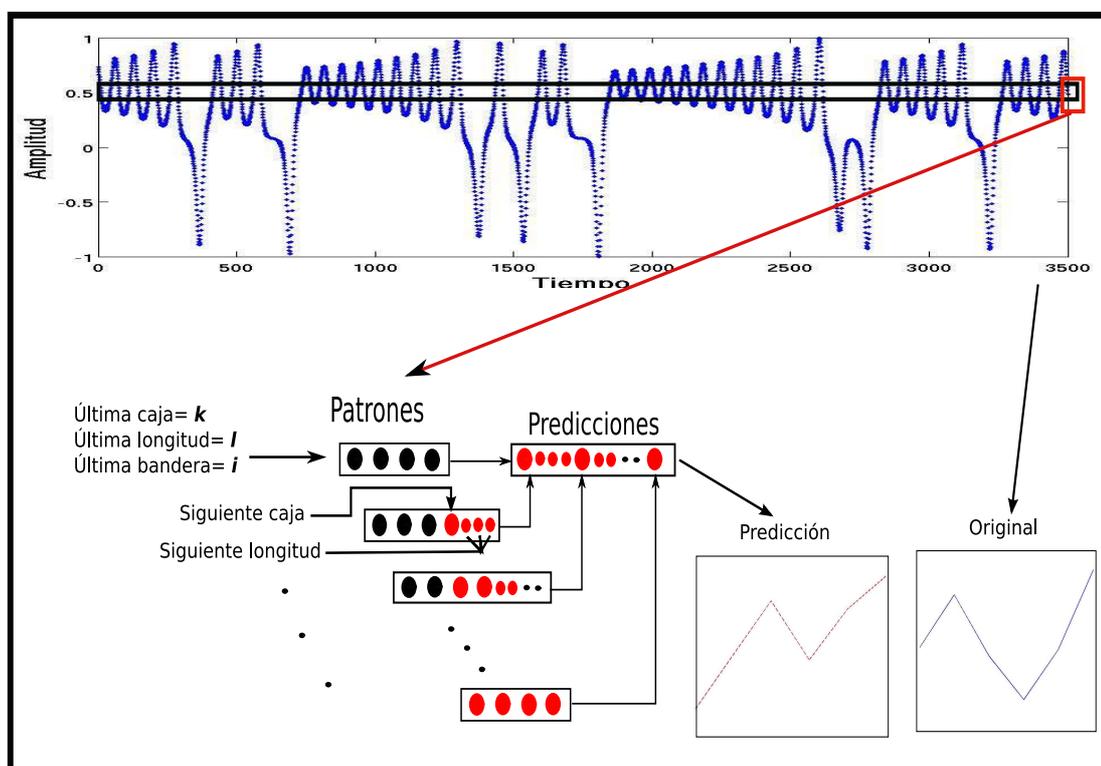


Figura 5.58: Esquema de la predicción iterada multipaso que se utilizó en este trabajo, la cual predice de manera iterada multiples puntos (puntos rojos pequeños).

Para evaluar la técnica de predicción que se propone en este trabajo, se probó el rendimiento de ésta mediante un conjunto de ST que por sus características dinámicas, son series con estructuras muy variadas, desde las más simples como las de tipo periódico, hasta las más complejas como las correspondientes a fenómenos naturales. Como se comentó anteriormente, este conjunto de ST se utiliza para probar las técnicas de predicción reportadas en la literatura, por lo que representan una muestra significativa debido a que este conjunto de ST llevan al límite a las técnicas de predicción.

Recordemos que, dentro de los pasos del proceso de aprendizaje de series de tiempo, el paso tres es el correspondiente a dividir la imagen de la ST en el número adecuado de cajas, el cual es el de mayor importancia, debido a que esta división de la imagen representa las secciones en las que se irá aprendiendo algún comportamiento de la ST objetivo. Durante el desarrollo de la técnica de aprendizaje se manejaron diferentes métodos para la división de la imagen de la ST en un número adecuado de *cajas*.

Es debido a lo anterior, que los resultados para la predicción de ST también se dividirá en tres partes. Cada una de estas partes representa el método que se utilizó para dividir la imagen de la ST generando un cierto número de *cajas* y a continuación se muestran los resultados de la predicción de ST para cada una de estas tres divisiones.

5.4.1. Primer tipo: división por anchura

Este primer tipo de división recordemos que se refiere a los primeros experimentos que se realizaron con la técnica Predicción Basada en Imagen (PBI), en donde estas primeras divisiones de la imagen de la ST fueron con respecto a la anchura de los intervalos definidos en la ecuación (3.10) de la sección 3.2.2, esto es, se dividió la imagen de la ST en 10 *cajas* en donde cada una de estas *cajas* contenían la misma anchura³ y ésta se calculó de acuerdo a la ecuación (3.18).

A continuación se mostrarán algunos resultados obtenidos durante la aplicación de la metodología de esta primera división en las siguientes figuras. Estas figuras corresponden a las ST que fueron utilizadas y descritas durante el proceso de aprendizaje, esto con la finalidad de observar cómo nos ayuda o afecta dicho proceso de aprendizaje al momento de realizar la predicción.

Un punto interesante sobre la predicción de ST, es el número de puntos que dicha técnica vaya a predecir, esto es, el número de puntos que se desea predecir a partir del último punto conocido, en este caso, el último punto aprendido de la ST. En nuestro caso particular y con el fin de comparar los resultados obtenidos de nuestra técnica PBI con los resultados encontrados en la literatura, se procedió a calcular los últimos 50 puntos de la ST, que son los que se mostrarán en las siguientes figuras.

Por lo tanto, las siguientes figuras siguen la misma filosofía de los resultados mostrados en el proceso de aprendizaje, es decir, para cada una de estas figuras se tiene lo siguiente:

- Primero, se muestra el resultado de los errores de predicción para cada uno de los 50 experimentos realizados.
- Posteriormente, se genera una distribución de probabilidad de ocurrir de dichos errores de predicción.
- Luego, se observará el resultado del experimento que obtuvo el error más bajo. En esta misma gráfica se observan dos series, una corresponde a la ST original (línea continua azul) y la otra correspondiente a la ST predicha (línea punteada roja).

Siguiendo con el orden de las series aprendidas, primero observemos la Figura 5.59, la cual, muestra el resultado de ejecutar la metodología de esta primera división a la ST “Seno”, cuyo comportamiento es periódico, la cual presentó un error de predicción (RMSE) entre el rango de 0.000041 y 0.00060 y éste se observa como una gráfica de distribución con un valor único pues si observamos la diferencia entre el error mayor y el menor, éste es mínimo. En la Figura 5.59c) se observa la gráfica de los puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul), la cual presenta un error de predicción igual a 0.000051 y como es posible observar los puntos predichos son bastante acertados con respecto a los originales.

³Recordemos que la anchura de un intervalo $[y_1, y_2]$ se refiere a la longitud finita definida como $y_2 - y_1$.

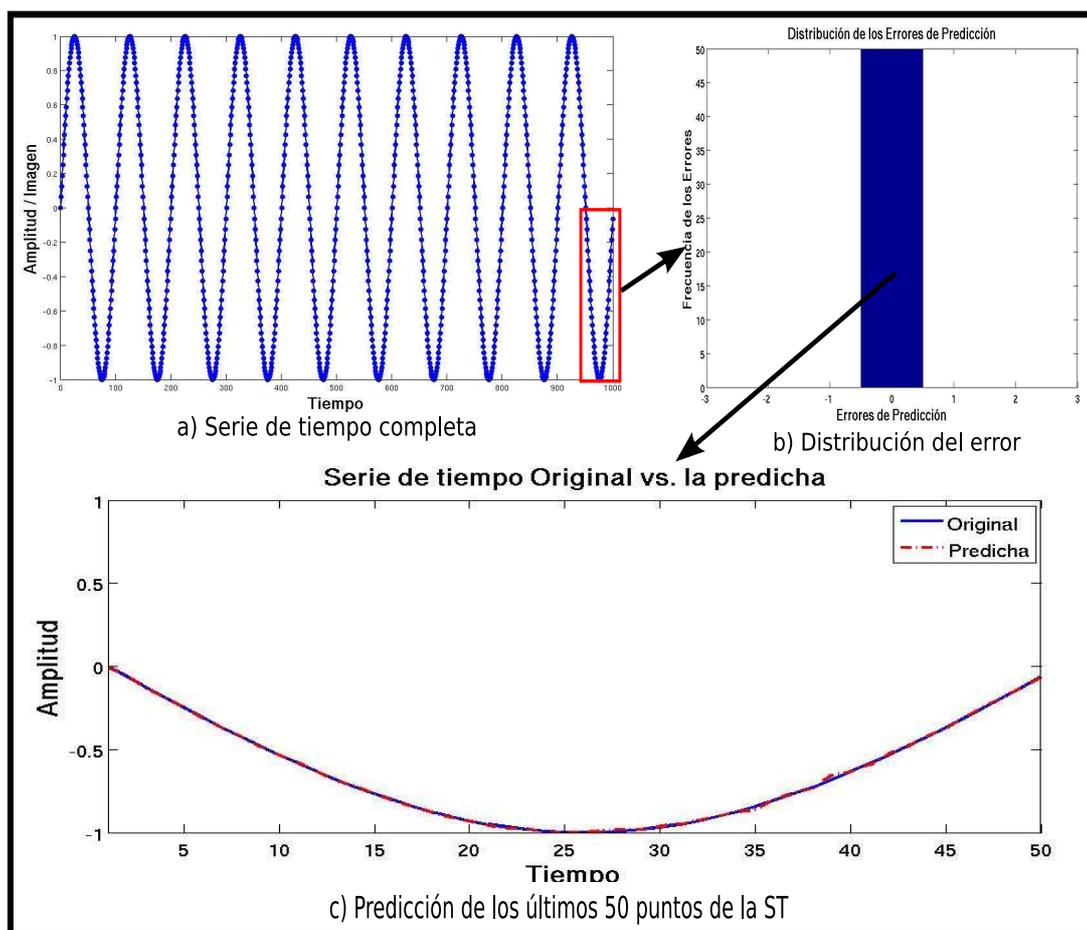


Figura 5.59: Resultado de aplicar el algoritmo de predicción a la ST "Seno". En a) se observa el comportamiento dinámico de la ST completa, en b) se observa la distribución de los errores de predicción después de ejecutar 50 veces este algoritmo y en c) se observa la predicción de los últimos 50 puntos de la ST.

En la Figura 5.60, se observan los resultados de aplicar la técnica de predicción propuesta a la ST "Qperiodic3" (comportamiento cuasi-periódico). Ésta presentó un error de predicción (RMSE) dentro del rango de 0.0715 y 0.37 (ver Figura 5.60b)). En la Figura 5.60c) observamos el resultado de esta predicción y como podemos observar en esta predicción se presentan dos fenómenos muy interesantes, como son la amplitud y la tendencia. Lo anterior significa que por un lado, los puntos predichos son bastante acertados respecto de los originales (amplitud), pero por otro lado, la tendencia nos indica la dirección de dichos puntos, que como podemos observar los puntos predichos siguen bastante bien la dirección de los puntos originales. Como se comentó en el capítulo anterior, la métrica que utilizamos el RMSE únicamente mide la amplitud, es decir la cercanía de los puntos predichos con respecto a los originales y para esta ST es igual a 0.0813 (ver Figura 5.60c)).

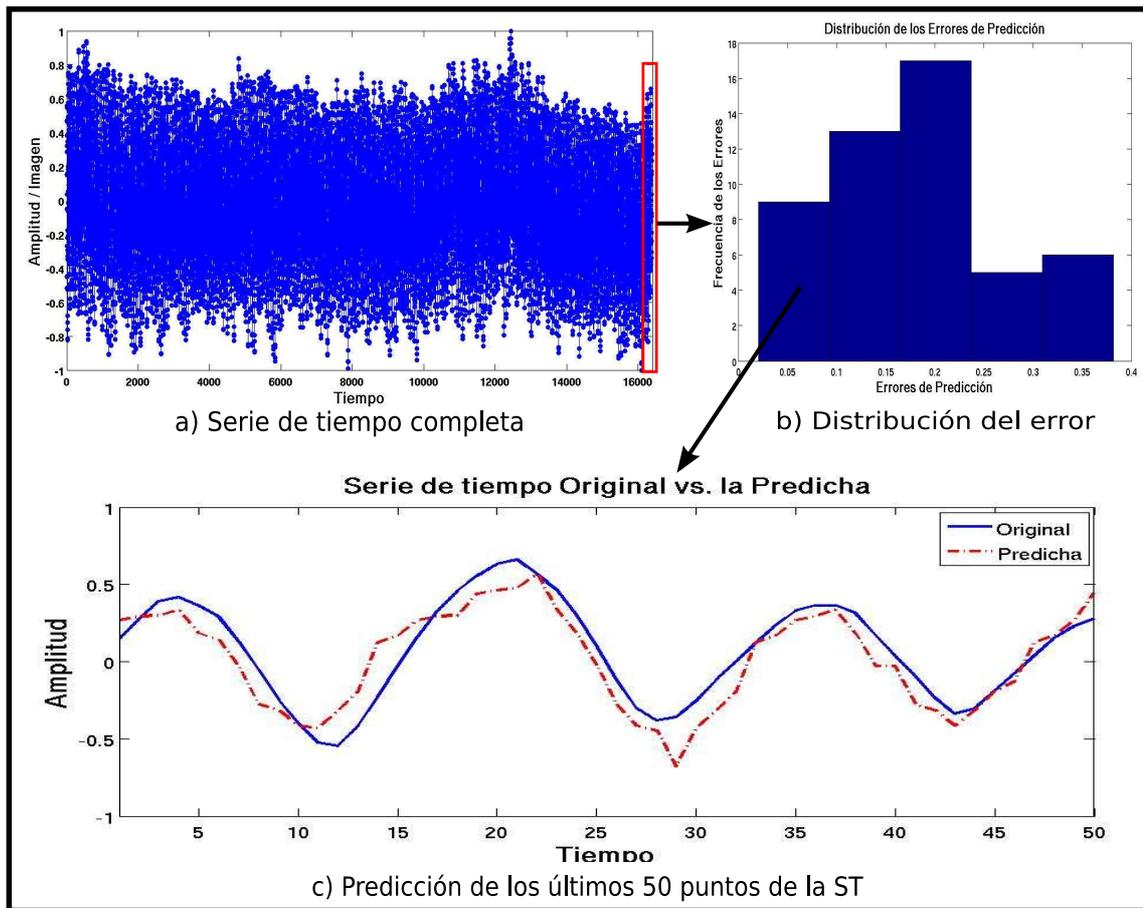


Figura 5.60: Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3” se observa en a) la ST completa y además se observa un rectángulo rojo el cual indica los 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.61, se observan los resultados de aplicar la técnica PBI a la ST “Lorenz” cuyo comportamiento dinámico es caótico. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.00325 y 0.042 (ver Figura 5.61b)). Como podemos observar en esta predicción la amplitud y la tendencia de los puntos predichos con respecto a los originales son casi iguales. Para la ST que se observa en la Figura 5.61c) se tiene un error de predicción igual a 0.0034. Lo anterior significa que los resultados que se obtuvieron para esta ST fueron bastante buenos y como es posible observar de la gráfica de distribución de los errores, la probabilidad de que al volver a ejecutar la técnica se obtenga un buen resultado es bastante alta.

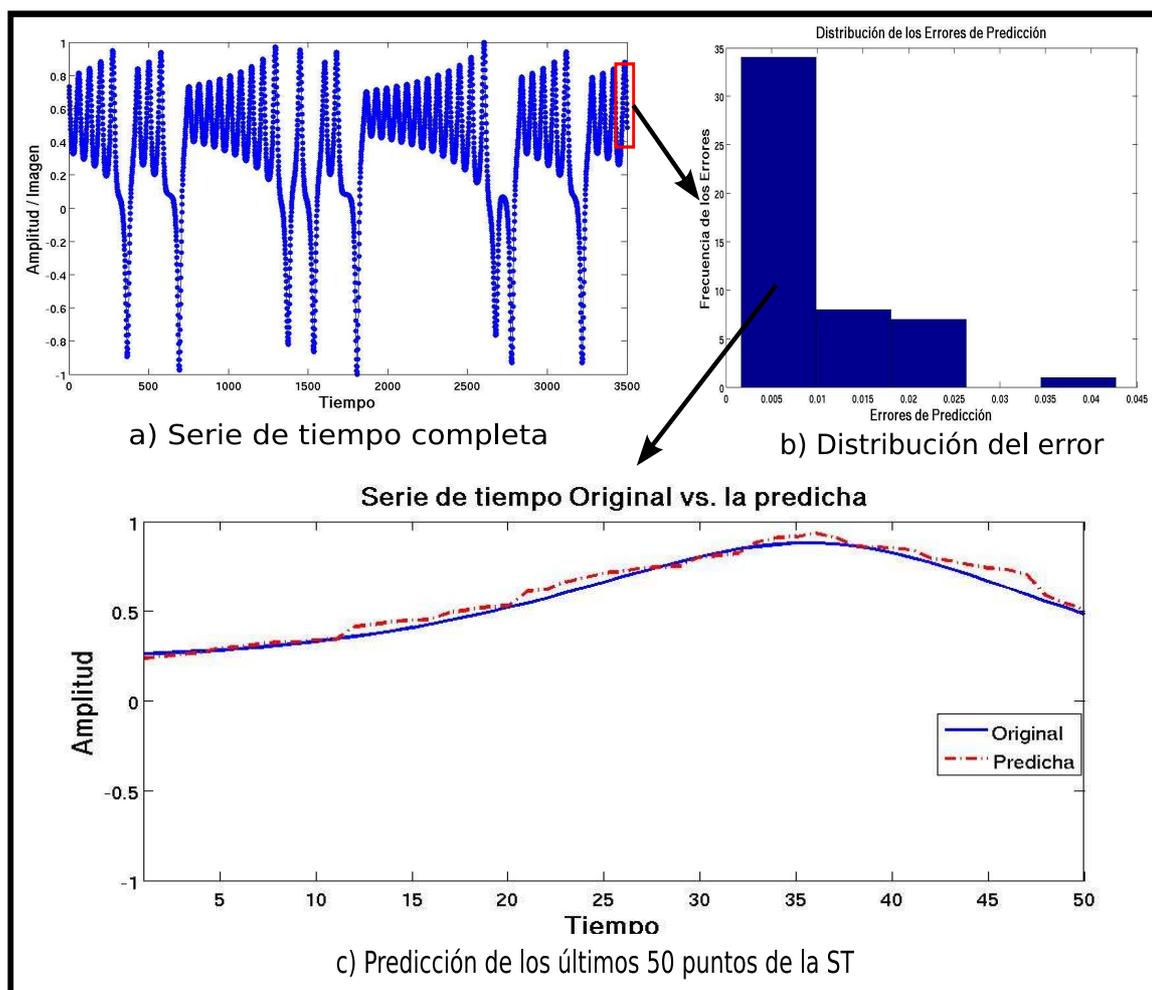


Figura 5.61: Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.

En la Figura 5.62, se observan los resultados de aplicar la técnica de predicción propuesta a la ST “tent” cuyo comportamiento dinámico es caótico. Ésta presentó un error de predicción dentro del rango de 0.05 y 0.11 (ver Figura 5.62b)). En el resultado de predicción de esta ST, se observa que la tendencia arroja mejores resultados que la amplitud. Para esta ST se presenta un error de predicción igual a 0.0502.

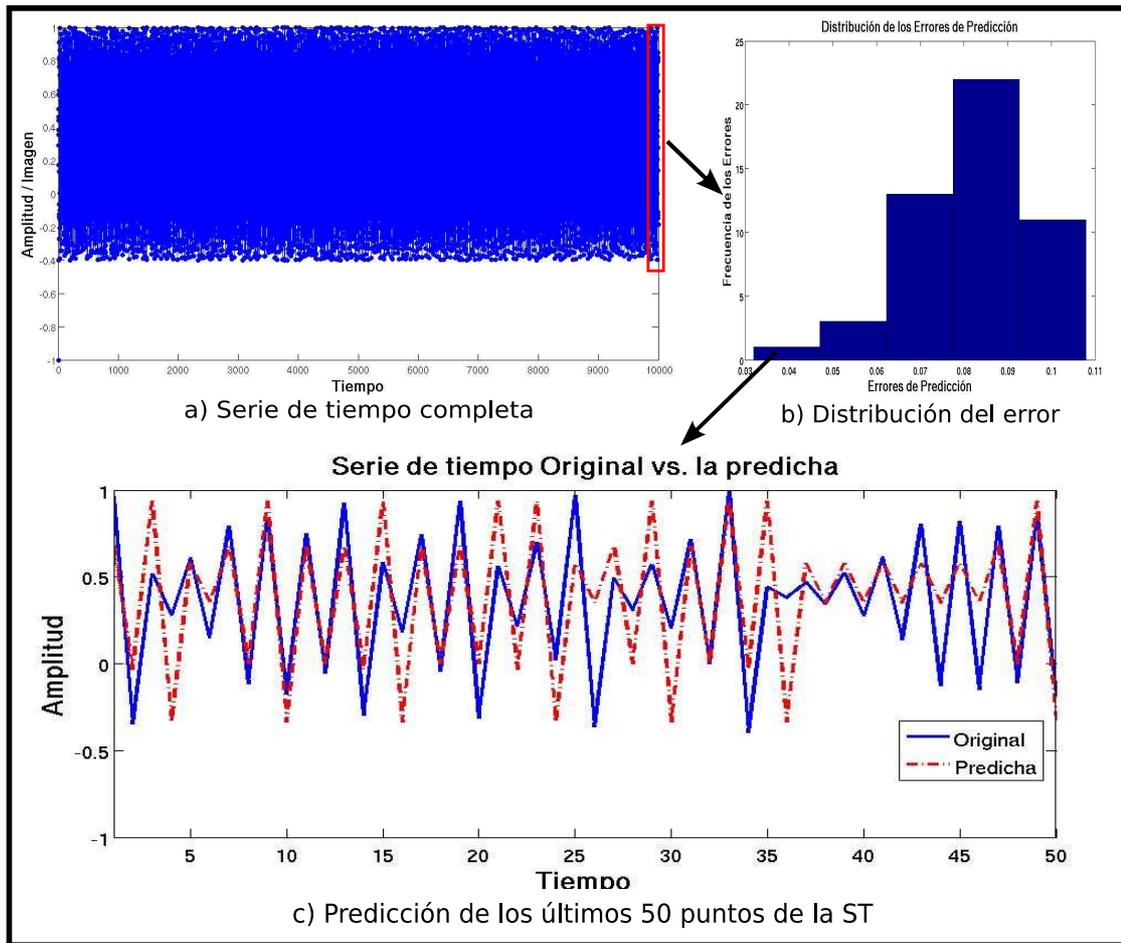


Figura 5.62: Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.63, se observan los resultados de aplicar la técnica PBI a la ST “S&P 500” cuyo comportamiento dinámico es complejo. Ésta presentó un error de aprendizaje (RMSE) dentro del rango de 0.090 y 0.120. En la Figura 5.63c) se observa la gráfica de los puntos predichos, la cual presenta un error de predicción igual a 0.0924 y como es posible observar a pesar de este comportamiento dinámico, se obtienen buenos resultados.

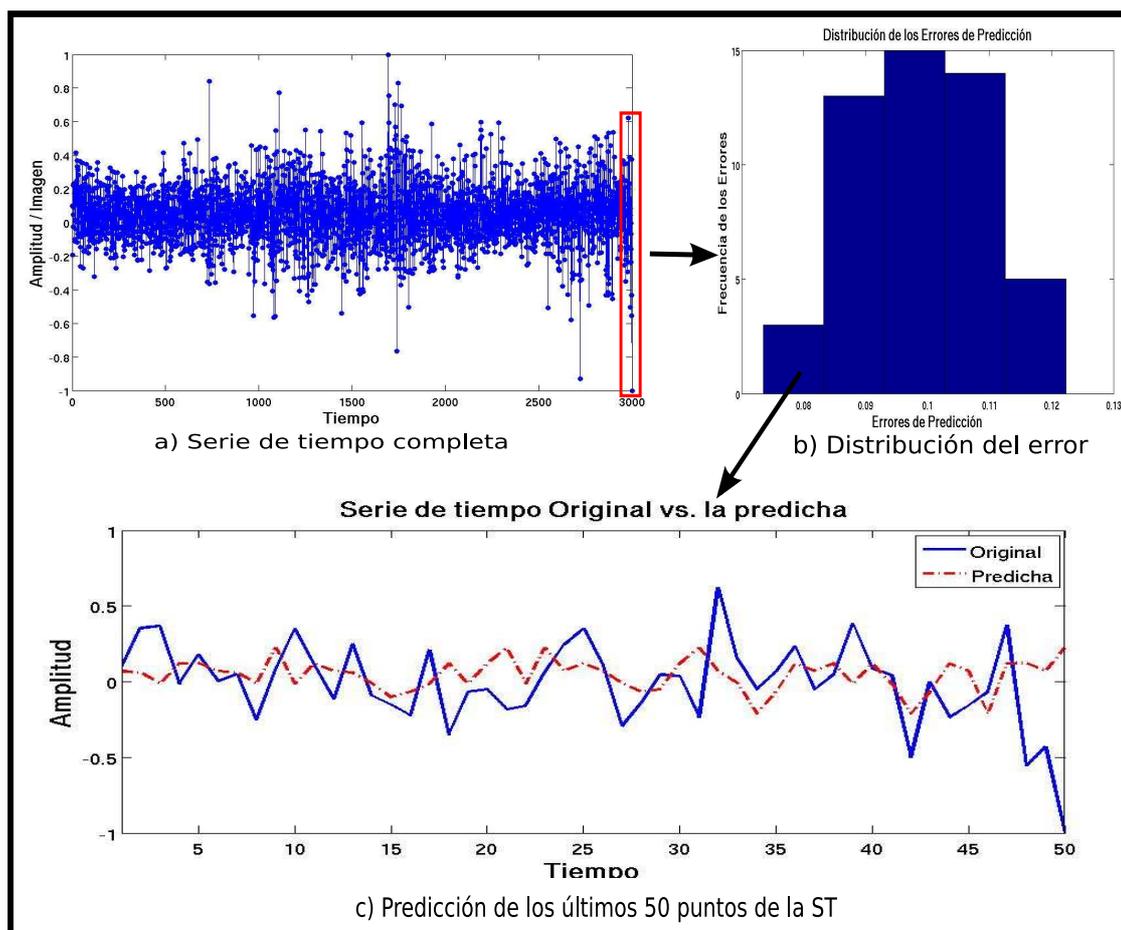


Figura 5.63: Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.64, se observan los resultados de aplicar la técnica PBI a la ST “Brownian Motion”, la cual presenta uno de los comportamientos dinámicos más complicados, el estocástico. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.050 y 0.75 (ver Figura 5.64b)). En la Figura 5.64c) se puede observar que los primeros 10 puntos predichos parecieran tener la tendencia y la amplitud erróneas, pero la técnica de predicción que proponemos tiene la capacidad de encontrar el camino adecuado eventualmente, en este caso si observamos nuevamente la Figura 5.64c) el punto 13 encuentra nuevamente la amplitud deseada de los puntos originales así como su tendencia. Ésta es otra de las propiedades con las que cuenta la técnica PBI y que la hace mejor que las demás, debido a que otras técnicas una vez que predicen un punto erróneo los futuros puntos serán erróneos, pues el primer error se acumula al momento de calcular la siguiente predicción. Para esta ST, el error de predicción es igual a 0.0397 (ver Figura 5.64c)). Lo anterior significa que los resultados que se obtuvieron para esta ST fueron bastante buenos y como es posible observar de la gráfica de distribución de los

errores, la probabilidad de que al volver a ejecutar la técnica se obtenga un buen resultado es alta.

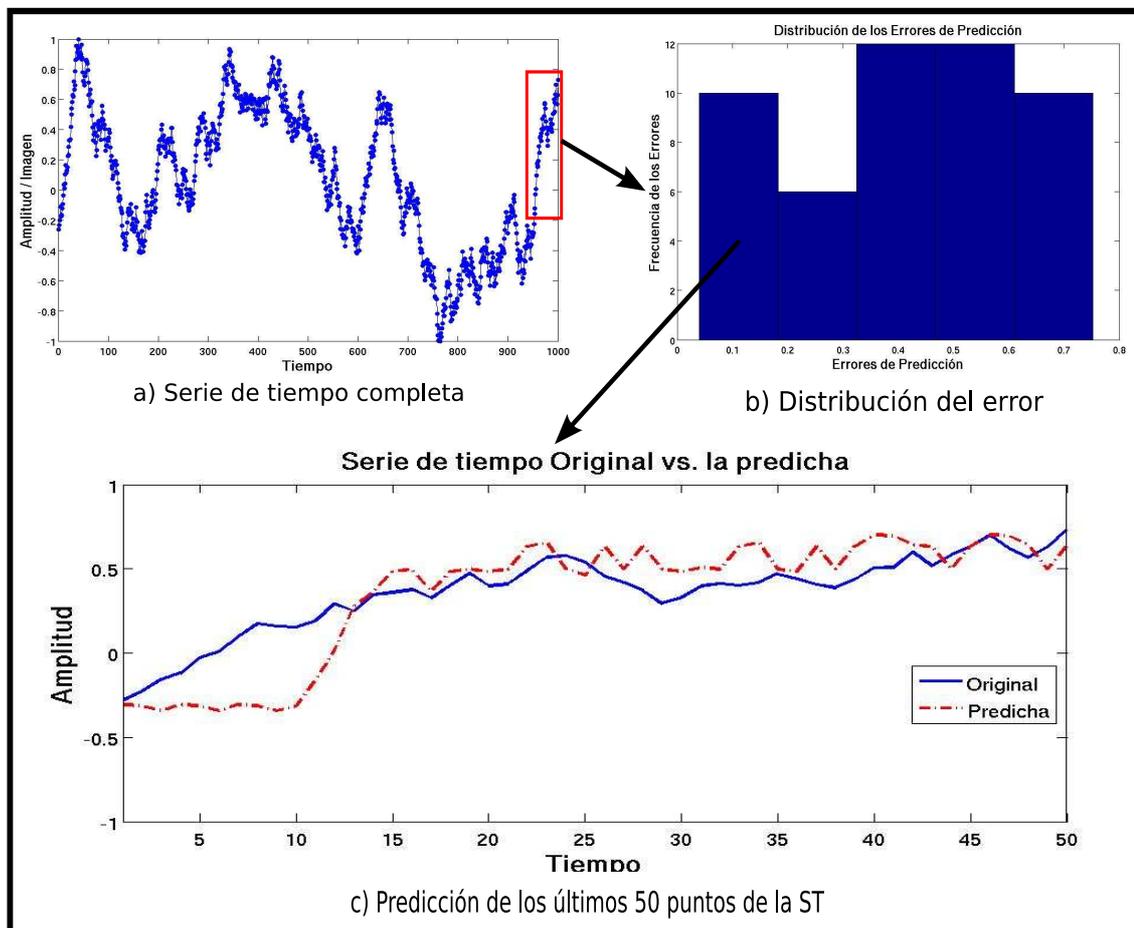


Figura 5.64: Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

5.4.2. Segundo tipo: división por frecuencias

Recordemos que durante esta segunda etapa, la división de la imagen de la ST en *cajas*, se realiza de la siguiente manera: primero, se obtiene la distribución de los puntos para cada caja, la cual se generó al dividir la imagen de la ST en *cajas* con la misma anchura, como en la etapa anterior, de donde obtenemos una frecuencia máxima y una frecuencia mínima. A partir de estas frecuencias se obtiene la frecuencia promedio, el cual es el número de puntos que debe de contener cada *caja*. Posteriormente, se procede a calcular de manera dinámica los intervalos de cada *caja* de acuerdo al número de puntos establecidos por las frecuencias

promedios, haciendo así que la división de la serie de tiempo sea dinámica. Una explicación más detallada se observa en la sección 3.2.2. Es debido a lo anterior, que a esta segunda etapa, la llamaremos *división por frecuencias*.

A continuación se mostrarán algunos resultados obtenidos durante la aplicación de este tipo de división en las siguientes figuras. Estas figuras corresponden a las ST que fueron utilizadas y descritas en la división anterior, esto con la finalidad de realizar una comparación en cuanto al rendimiento de este tipo de división respecto de los anteriores. Estas Figuras muestran la misma estructura que las anteriores.

Siguiendo con el orden de las series predichas de la división anterior, primero observemos la Figura 5.65, la cual, muestra el resultado de ejecutar la técnica de predicción empleando esta segunda etapa llamada *división por frecuencias*. a esta ST “Seno”, la cual presentó un error de predicción único igual a 0.000253 y como es posible observar los puntos predichos son bastante acertados con respecto a los originales.

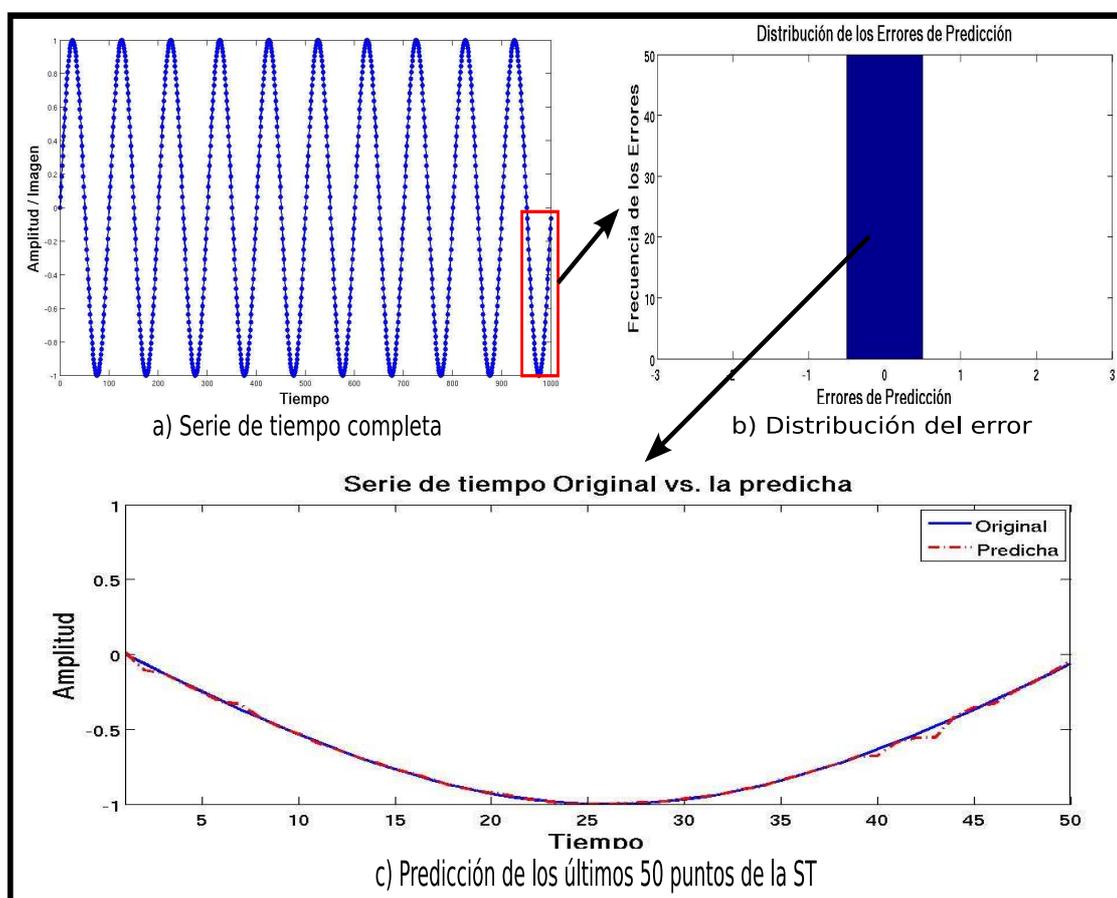


Figura 5.65: Resultados de la predicción de los últimos 50 puntos de la ST “seno”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.66, se observan los resultados de aplicar la técnica de predicción propuesta utilizando la metodología de esta segunda etapa a la ST “Qperiodic3”. Ésta ST presentó un error de predicción (RMSE) dentro del rango de 0.050 y 0.40 (ver Figura 5.66b)). En la Figura 5.66c) observamos el resultado de la predicción de lo últimos 50 puntos de la ST “Qperiodic3” la cual presenta un error de predicción igual a 0.0592. Como podemos observar en esta predicción se observa que la tendencia de los puntos predichos respecto de los originales es bastante buena, aunque éstos no presenten la misma amplitud. Lo anterior significa que los resultados obtenidos para esta ST fueron bastante buenos y como es posible observar de la gráfica de distribución de los errores, la probabilidad de que al volver a ejecutar la técnica se obtenga un buen resultado es alta.

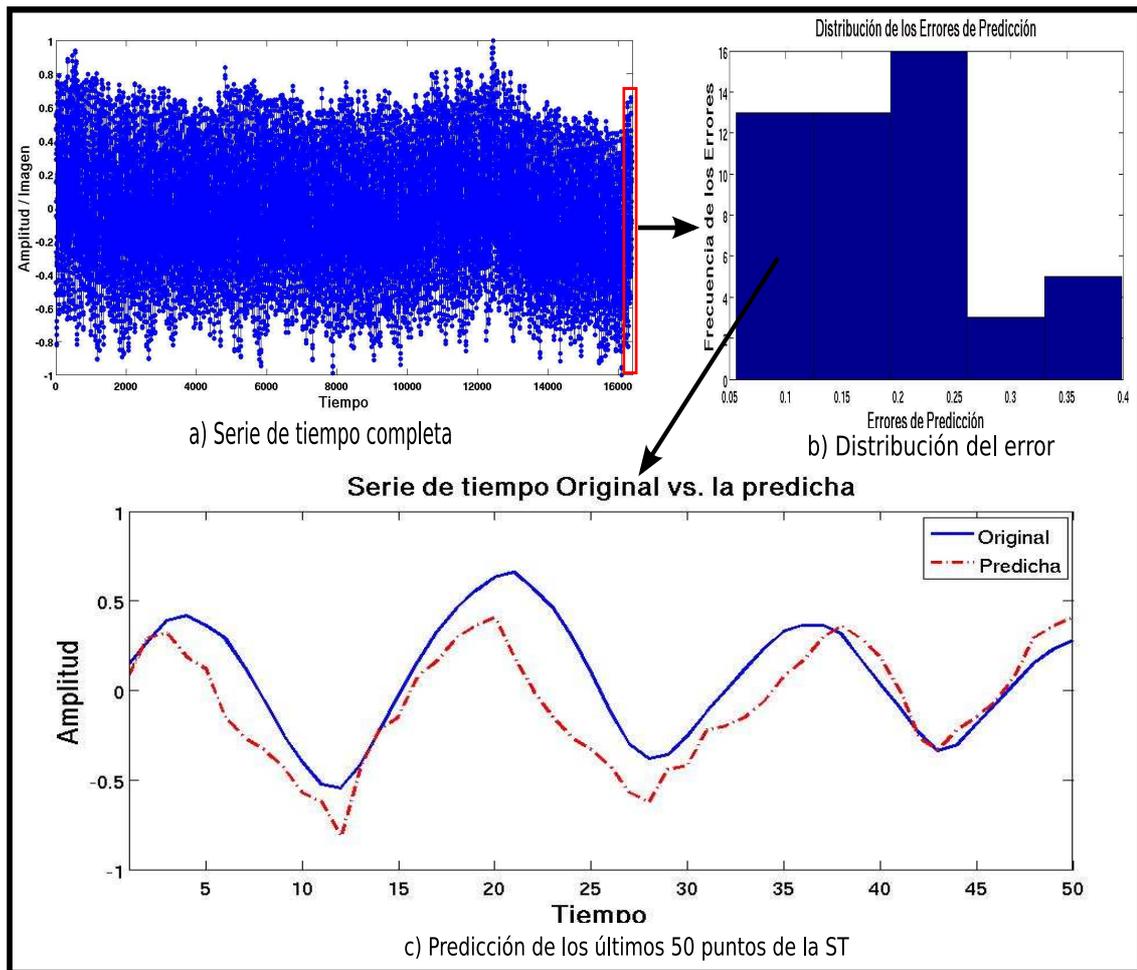


Figura 5.66: Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.67, se observan los resultados de aplicar la técnica PBI empleando la metodología de esta segunda división a la ST “Lorenz”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.0005 y 0.055 (ver Figura 5.67b)). En la Figura 5.67c) observamos el mejor resultado de la predicción, con un error igual a 0.00067, el cual es muchísimo menor que el resultado de la división anterior. Como podemos observar en esta predicción los valores de amplitud y tendencia para dichos puntos predichos es casi exacta. La probabilidad de que al volver a ejecutar la técnica se obtenga un buen resultado es cercano al 100 %.

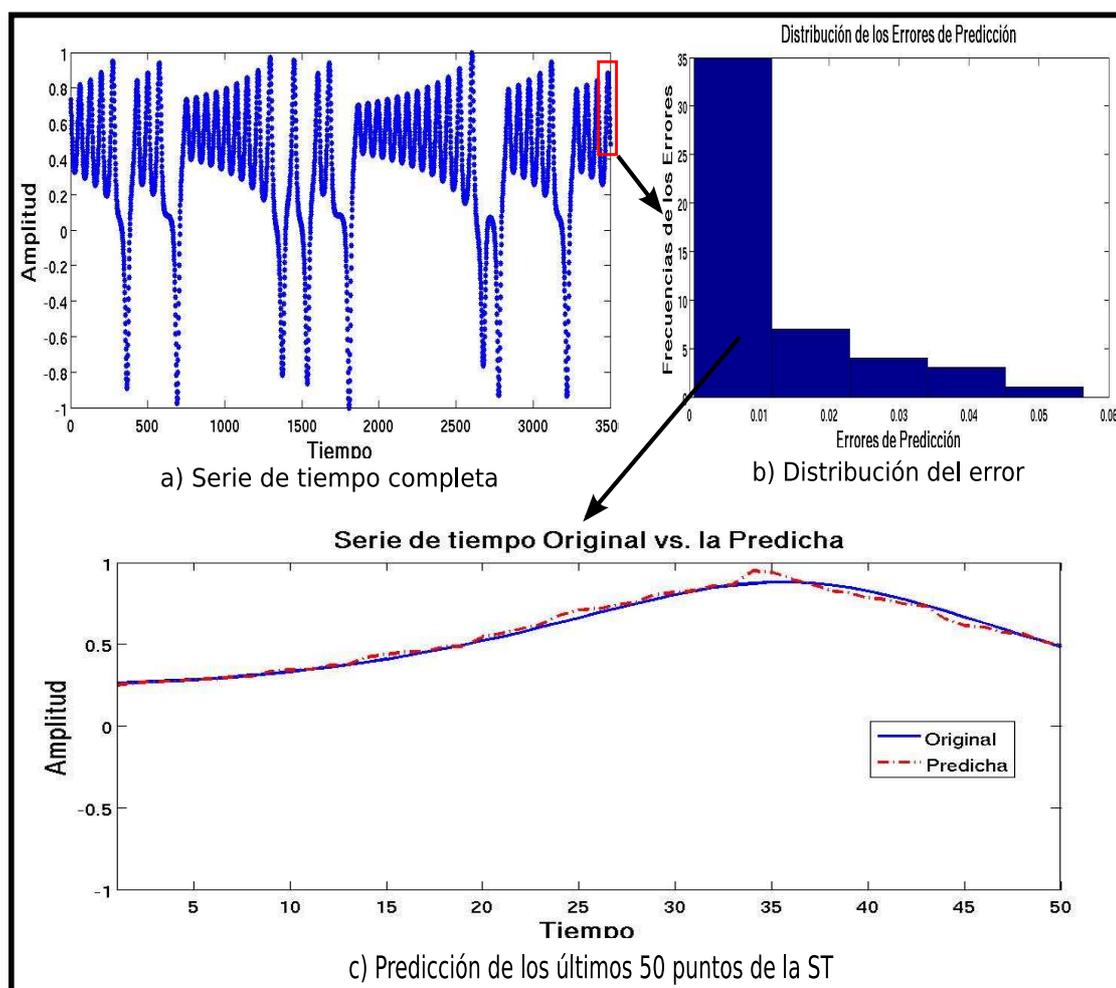


Figura 5.67: Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.

En la Figura 5.68, se observan los resultados de aplicar la técnica de predicción propuesta a la ST “tent”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.035 y 0.105 (ver Figura 5.68b)). En la Figura 5.68c) se observa la gráfica de los puntos predichos con respecto a los puntos originales, la cual presenta un error de predicción igual a 0.03868 y como es posible observar los puntos predichos son bastante acertados con respecto a los originales.

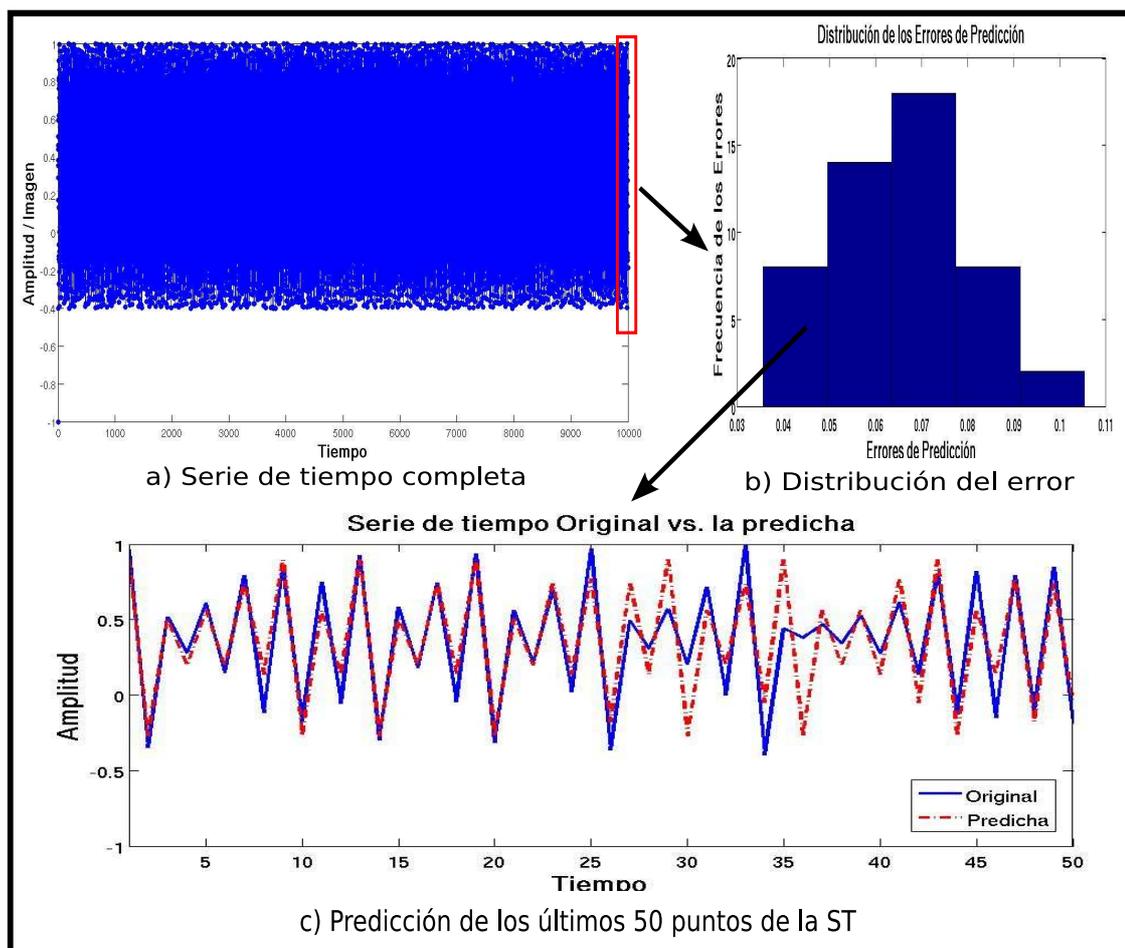


Figura 5.68: Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.69, se observan los resultados de aplicar la técnica PBI a la ST “S&P 500”. Ésta presentó un error de aprendizaje (RMSE) dentro del rango de 0.090 y 0.190. En la Figura 5.69c) se observa la gráfica de los puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul), la cual presenta un error de predicción igual a 0.1063.

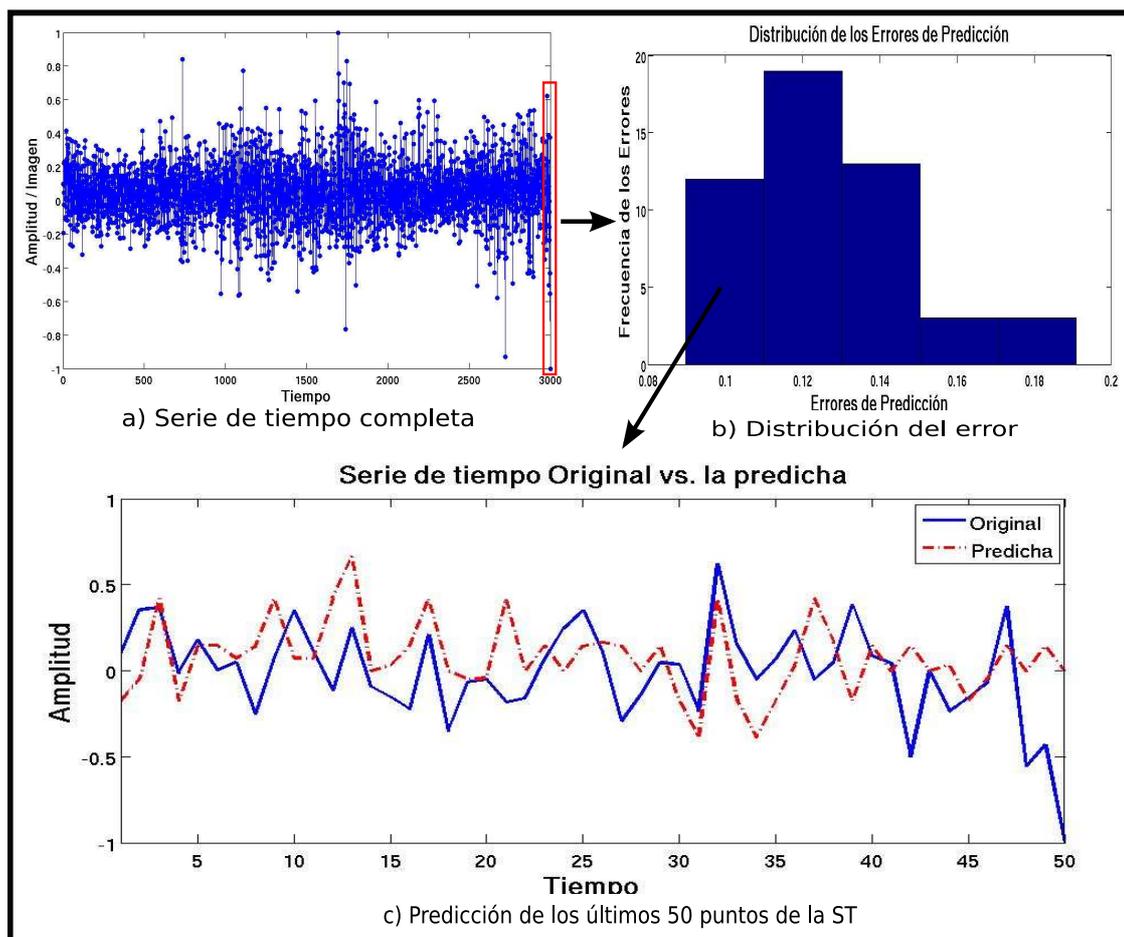


Figura 5.69: Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.70, se observan los resultados de aplicar la técnica PBI a la ST “Brownian Motion”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.0050 y 1.11 (ver Figura 5.70b)). En la Figura 5.70c) observamos el resultado de la predicción de lo últimos 50 puntos de la ST cuyo error de predicción es igual a 0.0154. En este resultado se puede observar que la amplitud y la tendencia de los puntos predichos con respecto a los puntos originales son muy acertados, incluso podríamos decir que estos resultados son mejores que los obtenidos en la división anterior. La probabilidad de que al volver a ejecutar la técnica se obtenga un buen resultado es alta.

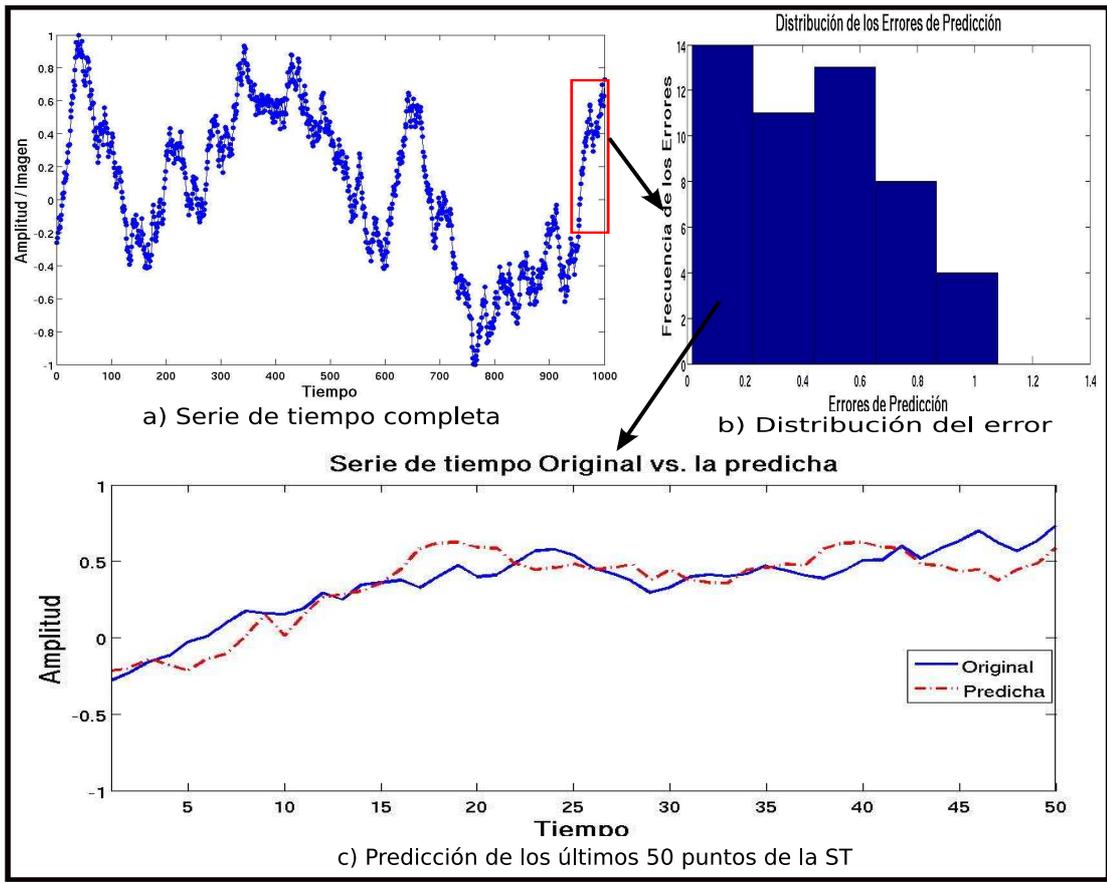


Figura 5.70: Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

5.4.3. Tercer tipo: división por porcentaje de información total

Durante esta división, se define el número de puntos dentro de cada *caja*, mediante la función área_f controlando que todas las *cajas* tengan igual número de información (puntos) y esto se realiza de dos maneras distintas:

1. De acuerdo al 10% de la información.
2. De acuerdo al 5% del total de la información.

A continuación se mostrarán algunos resultados obtenidos durante la aplicación de esta tercera división. Las siguientes figuras corresponden a las ST que fueron utilizadas en los pasos anteriores. Para mostrar la diferencia entre la división al 10% y 5% del total de la información se procederá a mostrar ambos resultados de manera consecutiva.

Siguiendo con el orden de las series predichas en las etapas anteriores, observemos la Figura 5.71, la cual, muestra el resultado de ejecutar la metodología de esta tercera etapa empleando el 10 % del total de la información para la ST “Seno”. Ésta presentó un error de predicción único igual a 0.00006, como se observa en el inciso b) de la Figura 5.71 y el resultado de esta predicción se observa en la Figura 5.71c). De acuerdo a lo anterior, podemos observar que la probabilidad de que al volver a ejecutar la técnica de predicción empleando esta segunda división, existe un 100 % de probabilidad de que obtengamos un buen resultado.

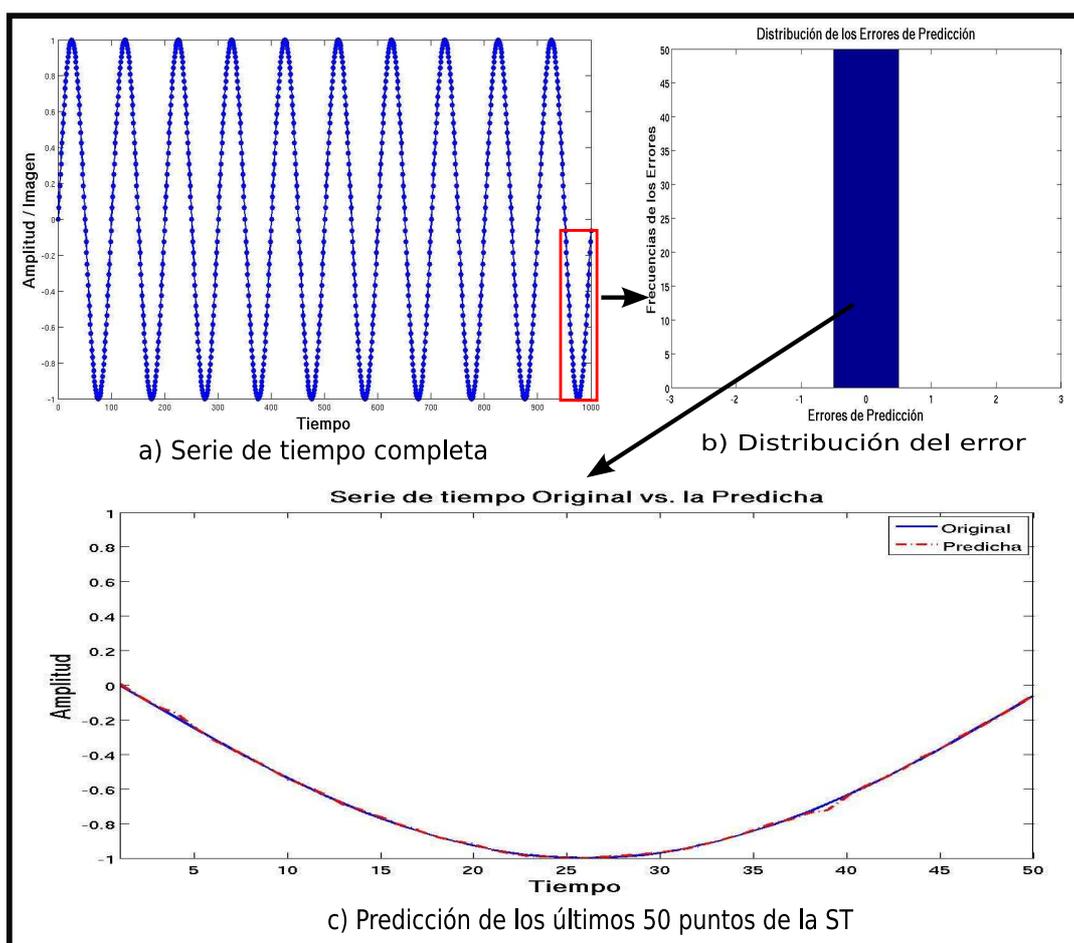


Figura 5.71: Resultados de la predicción de los últimos 50 puntos de la ST “seno”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.72, se muestra el resultado de ejecutar la metodología de esta tercera división empleando el 5 % del total de la información para la ST “Seno”.. Por el contrario esta ST presenta un error de predicción entre el rango 0.000023 y 0.000712 (ver, Figura 5.72b)). En la Figura 5.72c), observamos el resultado de la mejor predicción, cuyo error es igual a 0.000023.

Es posible observar que este resultado es mejor que los que se obtuvieron con las divisiones anteriores.

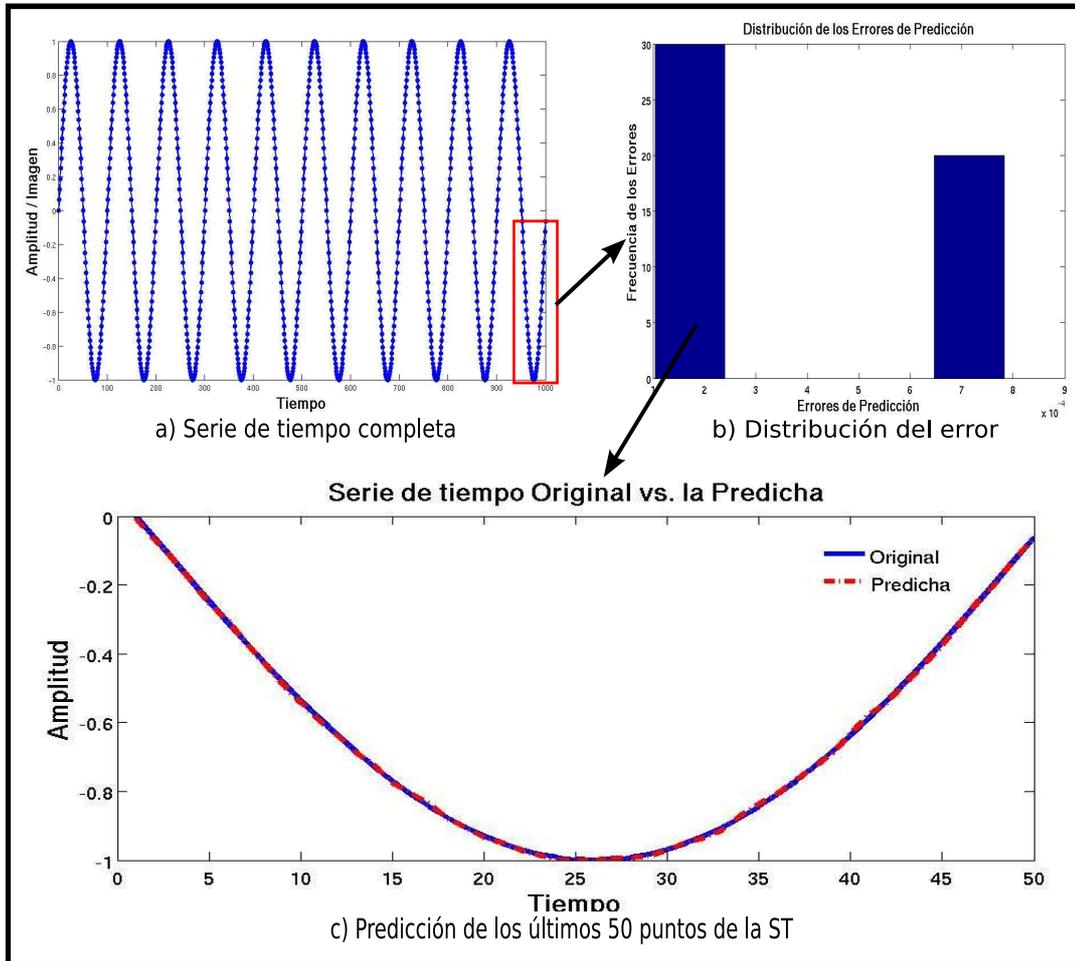


Figura 5.72: Resultados de la predicción de los últimos 50 puntos de la ST "seno", en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos a predecir, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.73, se observan los resultados de aplicar la técnica de predicción propuesta utilizando la metodología de esta tercera etapa con el 10% de la información total a la ST "Qperiodic3". Esta ST presentó un error de predicción (RMSE) dentro del rango de 0.050 y 0.40 (ver Figura 5.73b)). En la Figura 5.73c) observamos el resultado de la mejor predicción, la cual genera un error de predicción igual a 0.06007.

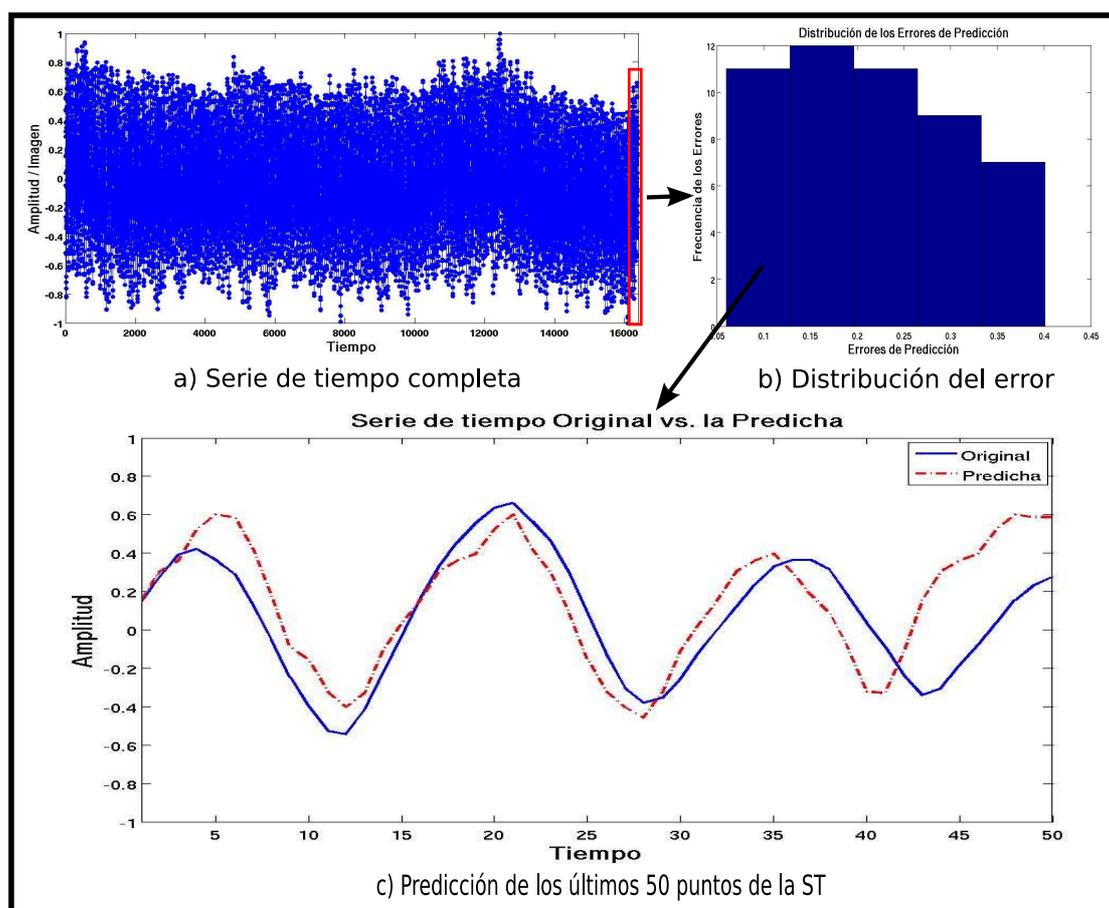


Figura 5.73: Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.74, se observan los resultados de aplicar la técnica de predicción propuesta utilizando la metodología de esta tercera división ahora con el 5% de la información total a la ST “Qperiodic3”. Como se observa en la Figura 5.74b), esta ST presentó un error de predicción dentro del rango de 0.050 y 0.45. En la Figura 5.74c) observamos el resultado de la mejor predicción, la cual tiene un error igual a 0.04423. Es posible observar que los puntos predichos del instante $i = 10$ hasta $i = 20$, la amplitud no es muy precisa pero la tendencia sí y nuevamente podemos observar que, aun cuando en estos instantes no se predice la amplitud adecuada, la técnica que proponemos encuentra nuevamente el camino adecuado en el instante $i = 21$, donde se observa que los siguientes puntos predichos siguen de manera casi exacta el comportamiento original.

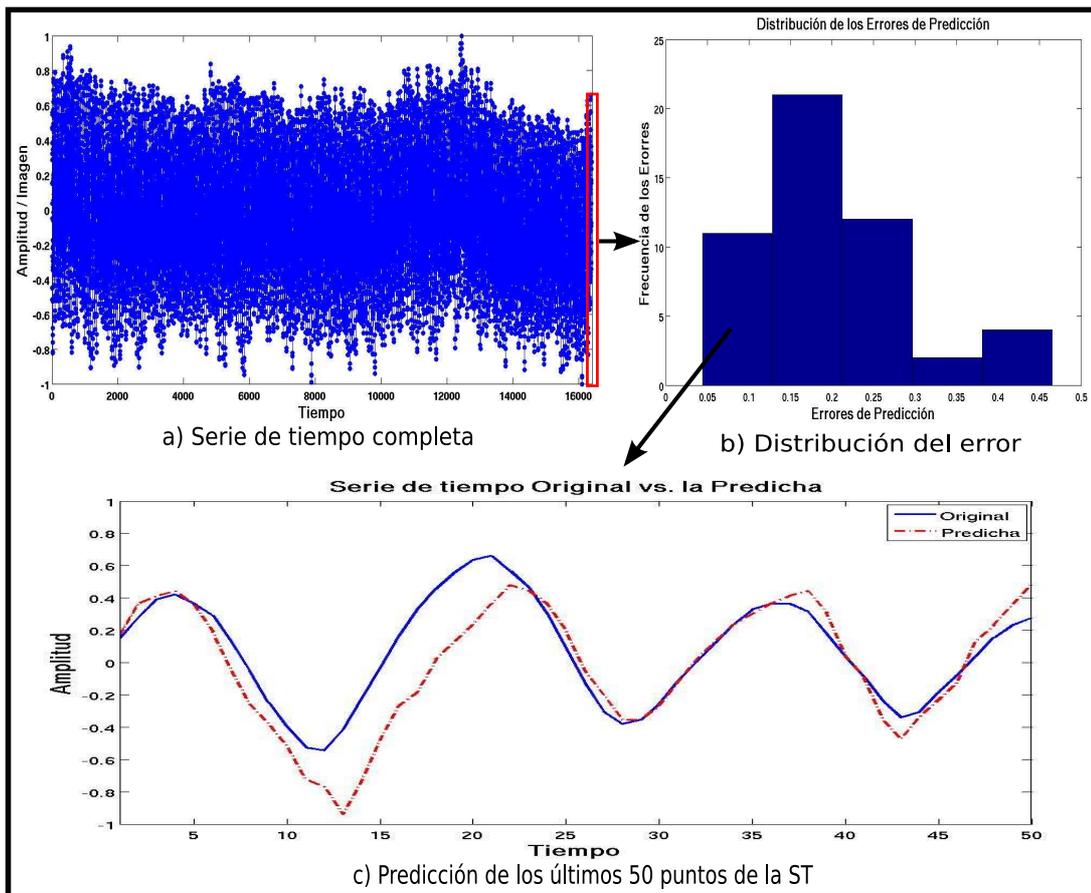


Figura 5.74: Resultados de la predicción de los últimos 50 puntos de la ST “Qperiodic3”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.75, se observan los resultados de aplicar la técnica PBI empleando la metodología de esta división con el 10 % del total de la información a la ST “Lorenz”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.001 y 0.055 (ver Figura 5.75b)). En la Figura 5.75c) observamos el resultado de la mejor predicción cuyo error es igual a 0.00123. Como podemos observar los resultados de esta predicción se parecen a los calculados anteriormente, pero la diferencia como podemos notar radica en que ésta no tiene tan buena precisión en la amplitud de los puntos debido al error que se obtiene. Es importante observar que, la probabilidad de que al volver a ejecutar la técnica se obtenga un buen resultado es cercano al 80 %.

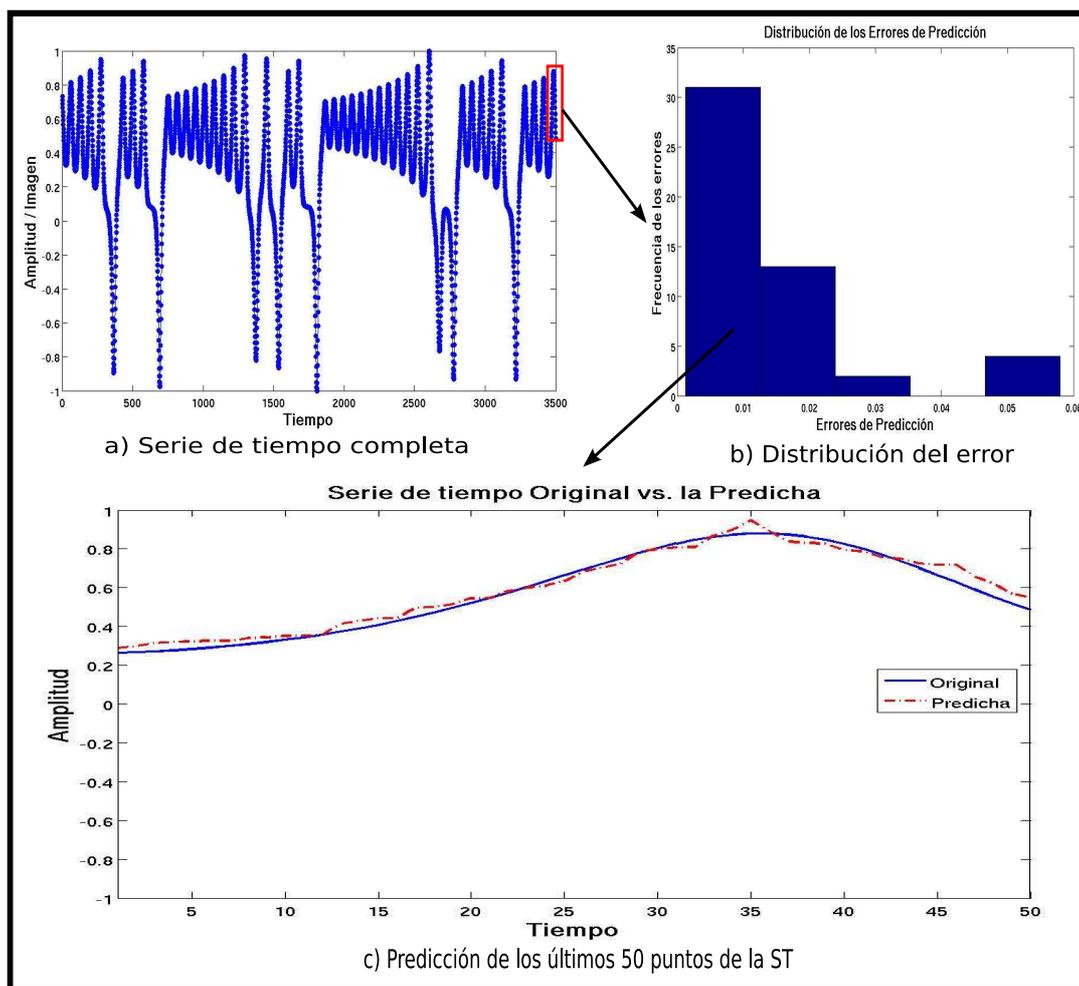


Figura 5.75: Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.

En la Figura 5.76, se observan los resultados de aplicar la técnica PBI empleando la metodología de esta tercera etapa ahora con el 5% del total de la información a la ST “Lorenz”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.0025 y 0.040 (ver Figura 5.76b)). En la Figura 5.76c) observamos el resultado de la mejor predicción, cuyo error es igual a 0.002793.

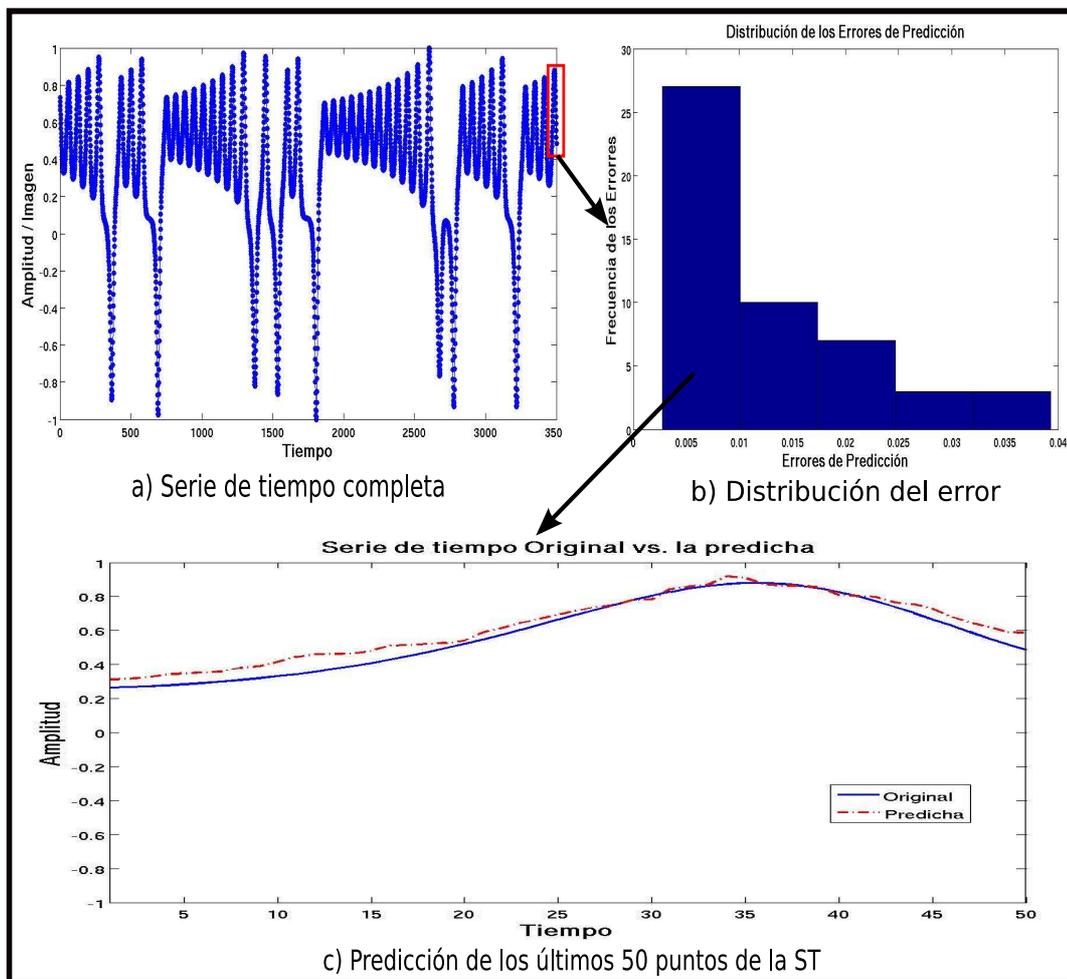


Figura 5.76: Resultados de la predicción de los últimos 50 puntos de la ST “Lorenz”, en a) se observan todos los puntos de la ST original, en b) se observa la gráfica de distribución de los errores de predicción y en c) se observan los resultados de la predicción.

En la Figura 5.77, se observan los resultados de aplicar la técnica de predicción propuesta bajo la metodología de esta tercera etapa empleando el 10 % del total de la información a la ST “tent”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.045 y 0.10 (ver Figura 5.77b)). En la Figura 5.77c) se observa la gráfica de los puntos predichos con respecto a los puntos originales, la cual presenta un error de predicción igual a 0.04156 y como es posible observar los puntos predichos son bastante acertados con respecto a los originales.

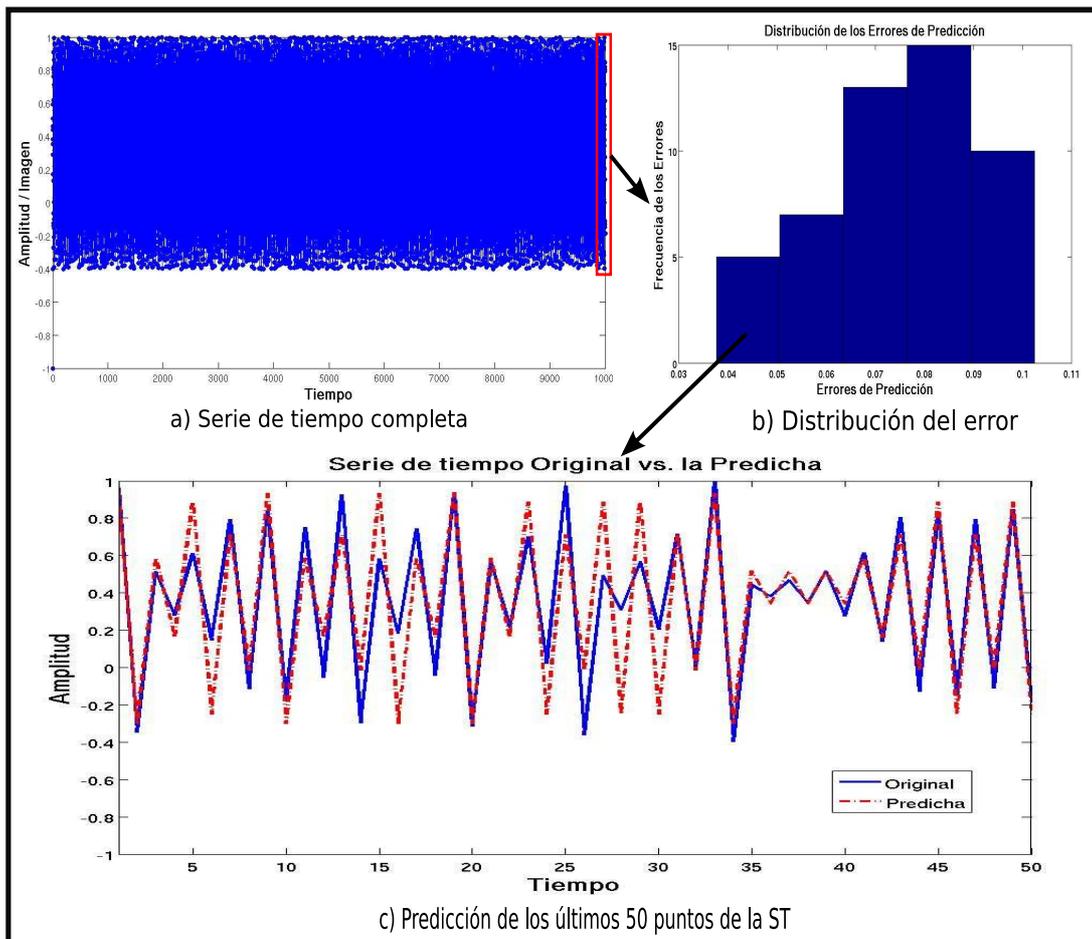


Figura 5.77: Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.78, se observan los resultados de aplicar la técnica de predicción propuesta aplicando la metodología de esta tercera etapa empleando el 5% del total de la información a la ST “tent”. Ésta a diferencia de la anterior, presentó un error de predicción (RMSE) dentro del rango de 0.025 y 0.12 (ver Figura 5.78b)). En la Figura 5.78c) se observa el resultado de la mejor predicción, la cual presenta un error igual a 0.035035, el cual si lo comparamos con los que se obtuvieron en las divisiones anteriores, éste arroja el mejor resultado. Obsérvese que para esta predicción, después de 35 pasos, el sistema de predicción es capaz de corregirse y minimizar los errores de predicción, en cambio los otros métodos de predicción no son capaces de este tipo de recuperación.

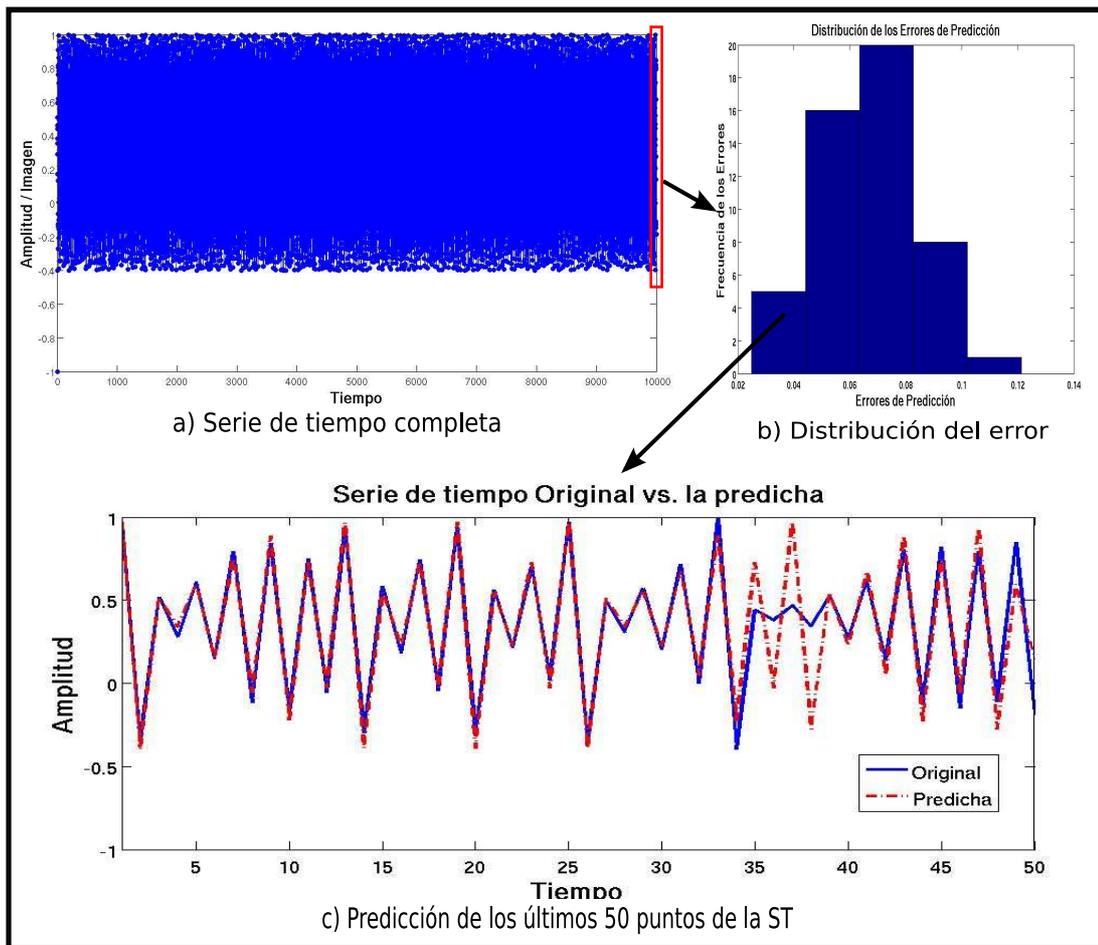


Figura 5.78: Resultados de la técnica de predicción para la ST “tent”, en a) se observan todos los puntos de esta ST, en b) se observa la gráfica de la distribución del error de predicción y finalmente en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.79, se observan los resultados de aplicar la técnica PBI empleando la metodología de la tercera etapa aplicando en 10% del total de la información a la ST “S&P 500”. Al ejecutar la técnica de predicción, ésta presentó un error de aprendizaje (RMSE) dentro del rango de 0.070 y 0.140 . En la Figura 5.79c) se observa la gráfica de los puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul), la cual presenta un error de predicción igual a 0.0842.

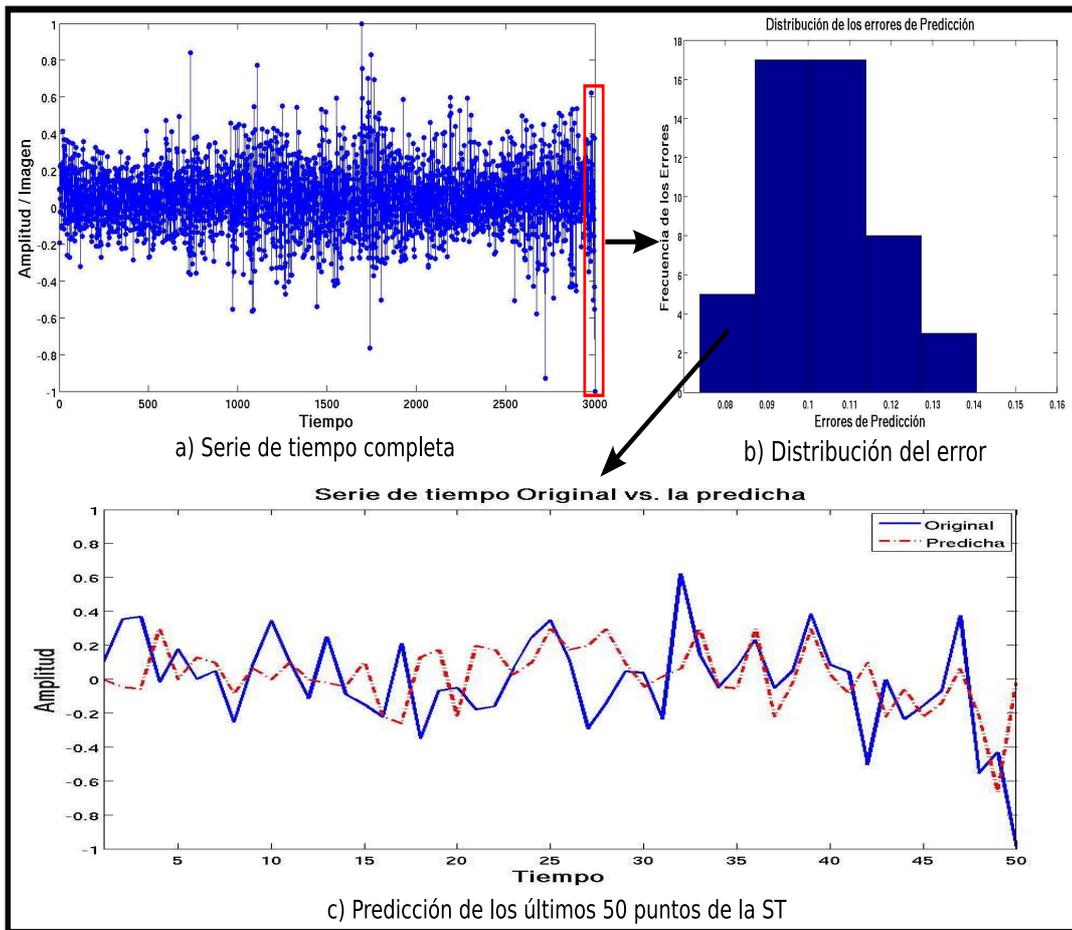


Figura 5.79: Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.80, se observan los resultados de aplicar la técnica PBI empleando la metodología de la tercera etapa aplicando ahora el 5 % del total de la información a la ST “S&P 500”. Al ejecutar la técnica de predicción propuesta 50 veces, éste presentó un error de aprendizaje (RMSE) dentro del rango de 0.080 y 0.160 . En la Figura 5.80c) se observa la gráfica de los puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul), la cual presenta un error de predicción igual a 0.091912. donde se observa que la tendencia de los puntos es más precisa que la amplitud.

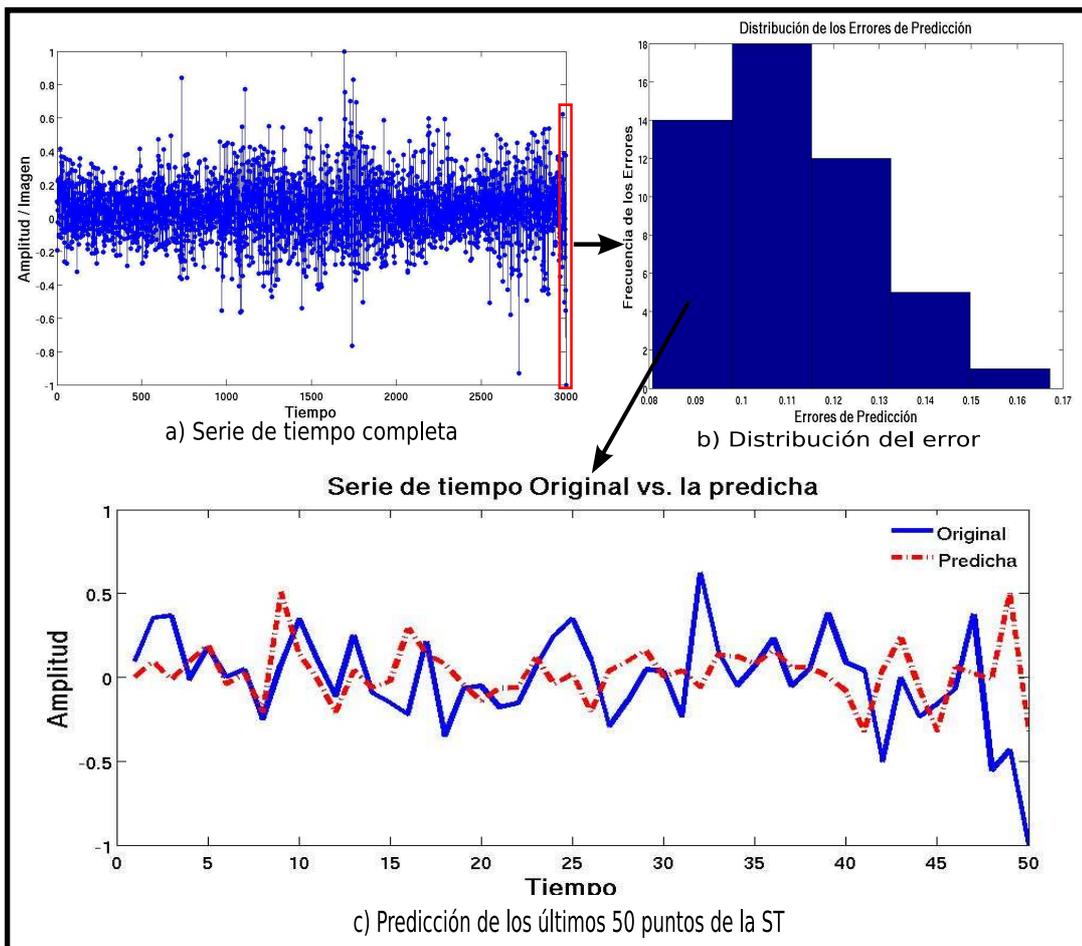


Figura 5.80: Resultados de la predicción de los últimos 50 puntos de la ST “S&P 500”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.81, se observan los resultados de aplicar la técnica PBI a la ST “Brownian Motion”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.0050 y 1.11 (ver Figura 5.81b)). En la Figura 5.81c) observamos el resultado de la mejor predicción, la cual genera un error igual a 0.01651. Lo anterior significa que los resultados que se obtuvieron para esta ST fueron bastante buenos, pero no lo suficiente, pues si se emplea la segunda división se obtienen mejores resultados.

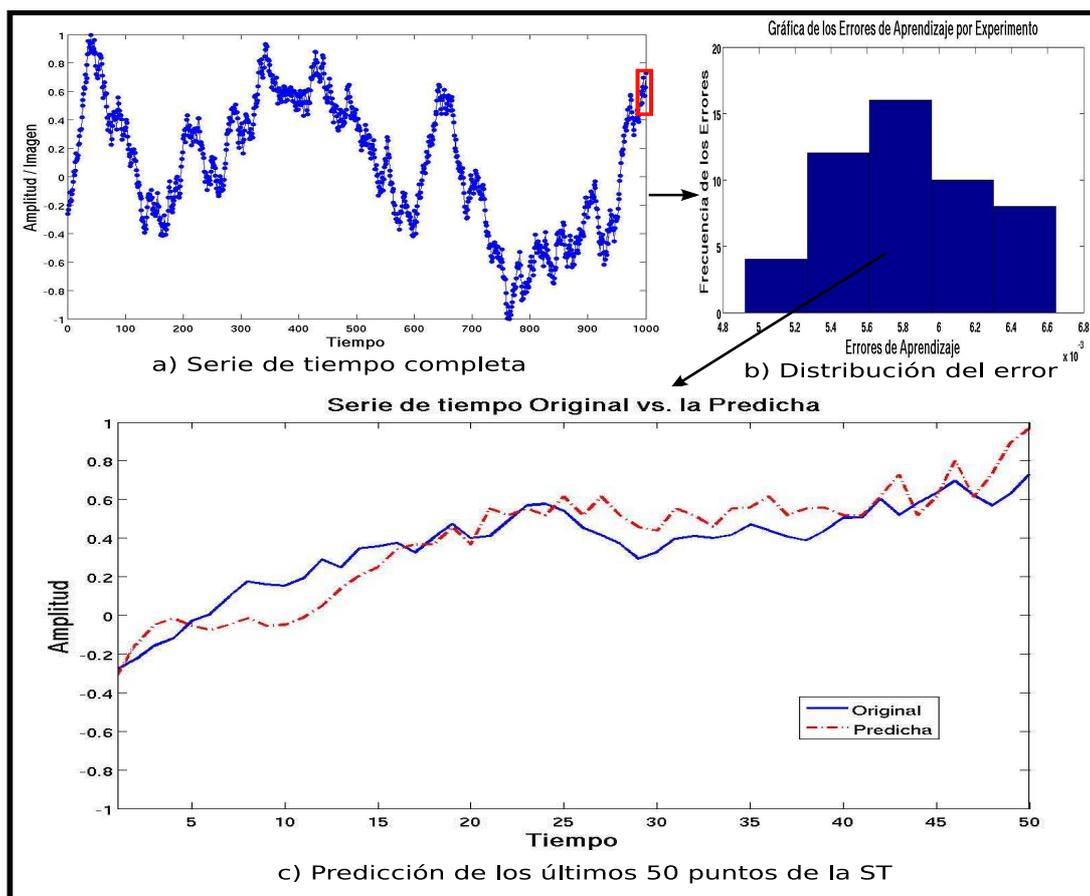


Figura 5.81: Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

En la Figura 5.82, se observan los resultados de aplicar la técnica PBI utilizando la metodología de la tercera etapa utilizando el 5% del total de la información a la ST “Brownian Motion”. Ésta presentó un error de predicción (RMSE) dentro del rango de 0.0050 y 1.11 (ver Figura 5.82b)). En la Figura 5.82c) observamos el resultado de la mejor predicción y el error que se produjo fue igual a 0.021085, observamos que este resultado no es tan preciso como el resultado que se obtiene de aplicar la segunda división.

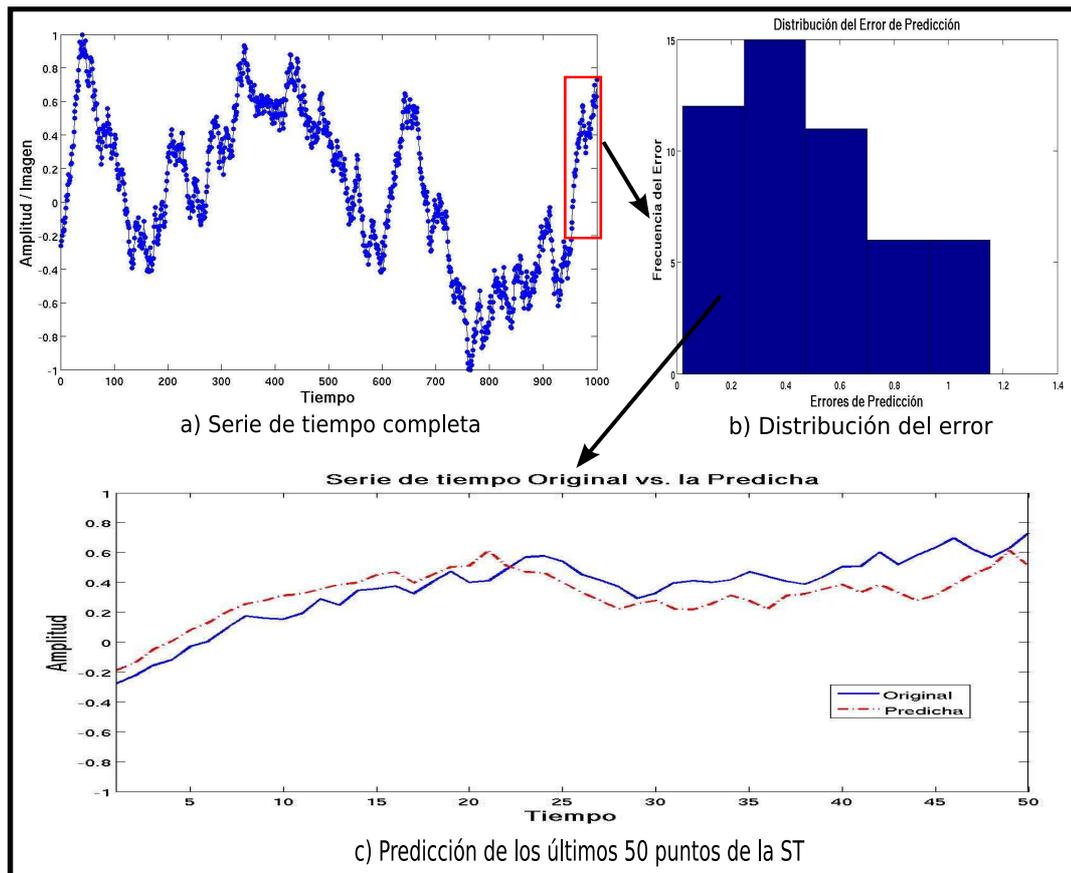


Figura 5.82: Resultados de la predicción de los últimos 50 puntos de la ST “Brownian Motion”, en a) se observan todos los puntos de esta ST, en un recuadro rojo se observan los últimos 50 puntos, en b) se observa la distribución de los errores de predicción y en c) se observan los 50 puntos predichos (línea punteada roja) con respecto a los puntos originales (línea continua azul).

Para que sea posible realizar una comparación entre los resultados obtenidos durante la aplicación de las distintas metodologías para dividir la imagen de la ST, se realizó la siguiente Tabla 5.4, en donde se observa resaltado en negritas el resultado de la mejor predicción de ST para los últimos 50 puntos de acuerdo a la metodología empleada para dividir la imagen de cada ST.

De la tabla anterior, es posible observar que la diferencia entre el mejor resultado con respecto a los demás es muy pequeña, por lo que podemos decir que cualquiera de las maneras que se escoja para dividir la imagen de la ST en un cierto número de cajas, siempre se va a tener un muy buen resultado. También, es importante observar que ninguna de las ST al aplicar la técnica PBI excede de la unidad, un hecho poco común entre las técnicas de predicción [30].

Ahora, observemos los resultados de aplicar nuestra técnica PBI para predecir una menor cantidad de puntos, esto es, se realizará la predicción de los últimos 30 puntos y estos resultados

Cuadro 5.4: Se observan los errores de predicción (RMSE) para una predicción de los últimos 50 puntos de la ST obtenidos mediante la aplicación de las distintas metodologías para dividir la imagen de la ST . Los mejores resultados se observan resaltados en “negritas”.

N°	Serie de Tiempo	Tipo de División			
		MismaAnch.	Frec.	10 %	5 %
1	Seno	0.000051	0.00025	0.00006	0.00002
2	Vanderpol	0.00678	0.00464	0.00787	0.00640
3	Qperiodic2	0.008773	0.00605	0.00546	0.00349
4	Qperiodic3	0.0813	0.0592	0.06007	0.04423
5	Mackey & Glass	0.145985	0.169417	0.19381	0.24519
6	Logistic	0.53587	0.69306	0.57093	0.57065
7	Lorenz	0.0034	0.00067	0.00123	0.002793
8	Rossler	0.01658	0.01557	0.02106	0.01731
9	Ikeda	0.23038	0.2279	0.18738	0.19834
10	Henon	0.29096	0.29971	0.23370	0.25257
11	Tent	0.0502	0.03868	0.04156	0.03503
12	A1	0.06551	0.06997	0.09961	0.04300
13	D1	0.09575	0.10786	0.11304	0.12269
14	Laser	0.257597	0.15061	0.14717	0.21901
15	El niño	0.04162	0.03548	0.05051	0.03642
16	Lovaina	0.021909	0.02564	0.01820	0.01268
17	S&P 500	0.0924	0.1063	0.0842	0.091912
18	Star	0.056140	0.05460	0.08313	0.07023
20	Brownian Motion	0.0397	0.0154	0.01651	0.02108
21	White Noise	0.45440	0.45590	0.406220	0.37084

se observan en la siguiente Tabla 5.5, donde nuevamente se observa que los mejores resultados se encuentran en “negritas”.

5.5. COMPARACIÓN DE LOS RESULTADOS DE PREDICCIÓN CON OTRAS TÉCNICAS 161

Cuadro 5.5: Se observan los errores de predicción (RMSE) obtenidos mediante la aplicación de las distintas metodologías para dividir la imagen de la ST aplicados para una predicción de los últimos 30 puntos de las ST. Los mejores resultados se observan resaltados en “negritas”.

N°	Serie de Tiempo	Tipo de División			
		MismaAnch.	Frec.	10 %	5 %
3	Qperiodic2	0.00802	0.08549	0.08539	0.00802
4	Qperiodic3	0.01908	0.01612	0.01257	0.01860
6	Logistic	0.37472	0.41055	0.15796	0.29841
7	Lorenz	0.00130	0.00095	0.00061	0.00111
8	Rosler	0.01492	0.00697	0.01425	0.00428
9	Ikeda	0.17106	0.13621	0.22388	0.19098
10	Henon	0.26600	0.17097	0.18151	0.28661
13	D1	0.03946	0.06047	0.05161	0.02619
14	Laser	0.15233	0.22707	0.16503	0.21443
16	Lovaina	0.01088	0.00867	0.01111	0.00598
17	S&P 500	0.09053	0.08219	0.11102	0.10071
18	Star	0.03391	0.03882	0.04173	0.04696
19	Flujo diario del río Jokulsa	0.00131	0.00038	0.00021	0.00007
20	Brownian Motion	0.03978	0.01544	0.00865	0.02108
22	Precio del Oro	0.00542	0.00932	0.00522	0.00143
23	Flujo del río Colorado	0.00819	0.01377	0.00973	0.00956
24	Nivel de agua del lago Erie	0.08742	0.06586	0.07819	0.08751
25	Nivel mínimo de agua del río Nilo	0.52967	0.47642	0.43780	0.38271
26	Número de manchas solares	0.00557	0.00716	0.00961	0.01061
27	Temperatura máxima de Melbourne	0.11595	0.16019	0.10469	0.14220
28	Temperatura mínima de Melbourne	0.04880	0.02834	0.05839	0.0355
29	Southern oscilation	0.06553	0.09420	0.08683	0.05550

Nuevamente, observamos que casi los mejores resultados se encuentran en la última columna, correspondiente de utilizar la metodología de dividir la imagen de la ST en el 5 % del total de la información total, este hecho se debe en gran manera a que la mejor información que se obtiene del proceso de aprendizaje recae sobre esta metodología.

5.5. Comparación de los resultados de predicción con otras técnicas

Como se mencionó a lo largo de esta tesis, encontramos dos técnicas que resolvían el problema de predicción para series de tiempo a través de un aprendizaje previo, dichas técnicas son las redes neuronales y las máquinas de soporte vectorial.

A continuación, presentaremos una comparación entre la técnica que nosotros proponemos con respecto a las máquinas de soporte vectorial y redes neuronales.

5.5.1. Máquinas de soporte vectorial vs. Predicción basada en imagen

En el trabajo [30], publicado por Thompson et al. en 2004, se presenta una evaluación de la predicción de múltiples pasos de series de tiempo, mediante un corrimiento de ventana sobre el tiempo para las máquinas de soporte vectorial (SVM, por sus siglas en inglés) con dos funciones de kernel distintas (lineal y con base radial). Para su evaluación, se utilizó un conjunto de 12 ST de diferente origen y comportamiento dinámico. En este trabajo, se encuentra que la SVM poseen una buena capacidad para ajustarse a las diferentes dinámicas de las ST y presenta un buen desempeño para la predicción de los primeros puntos de las series utilizando la función kernel radial.

La metodología que emplearon los autores [30], fue la siguiente: representar el conjunto de entrenamiento de la SVM como una ventana deslizante sobre el eje del tiempo, la cual permite utilizar los últimos p valores de una ventana para predecir el siguiente valor de una ST. Lo anterior es posible debido a que el corrimiento de la ventana a través de la serie, permite a la SVM (mediante la función de kernel utilizada) estimar la dependencia existente entre cada uno de sus valores, de esta forma se obtiene la información necesaria para realizar la predicción del siguiente punto a apartir de los puntos anteriores de la ST. Entonces, para realizar una predicción de múltiples puntos es necesario crear un nuevo conjunto de entrenamiento cada vez que se realiza una predicción simple. Este nuevo conjunto se obtiene agregando al conjunto de entrenamiento original la ventana $w' = (x_{i+p-2}, \dots, x_{i+p})$, siendo x_{i+p} el valor correspondiente a la predicción anterior.

La evaluación de la técnica de predicción con corrimiento de ventana, se llevó a cabo modificando la herramienta mySVM [26]. En los experimentos de este trabajo se utilizaron las funciones de kernel con base radial y lineal para establecer un punto de comparación, entre dos diferentes funciones de kernel de SVM en la predicción de series de tiempo.

Los experimentos de la técnica de SVM se realizaron con casi las primeras 16 ST de la Tabla 5.1 de las cuales se predijeron los últimos 50 puntos y los puntos restantes fueron tomados como puntos de entrenamiento, siendo organizados en ventanas de tamaño 3. Además, dichas ST fueron normalizadas dentro del rango $[-1, 1]$, esto debido a que la SVM presenta dificultades para detectar tanto variaciones grandes (del orden de 1×10^3) como pequeñas (del orden 1×10^{-4}) en los datos de entrenamiento. Para determinar si una predicción utilizando SVM fue exitosa o no, se realizó la comparación entre los puntos predichos con respecto a los originales mediante la métrica RMSE.

A continuación en la Figura 5.83 se muestra una comparación entre la mejor predicción obtenida para la ST “tent” con la técnica de SVM utilizando una función kernel de base radial con $\gamma = 0.01$ con respecto a la mejor predicción encontrada con nuestra técnica Predicción Basada en Imagen (PBI). Como se puede observar, el resultado de la predicción de los puntos de esta ST con la técnica SVM, muestra que el seguimiento de la tendencia de la serie predicha con respecto a la original es parecido a pesar de las discrepancias entre las amplitudes de la señal original y predicha. Por otro lado, si observamos los resultados que se obtienen con la técnica de predicción que estamos proponiendo, se observa tanto una muy buena tendencia así como la misma amplitud entre los puntos predichos y originales. Otro punto importante que vale la pena recalcar es que aun cuando se predice una mala amplitud con la técnica que se propone, ésta tiene la capacidad de recuperar la convergencia del error de predicción, como se observa en la Figura 5.83b).

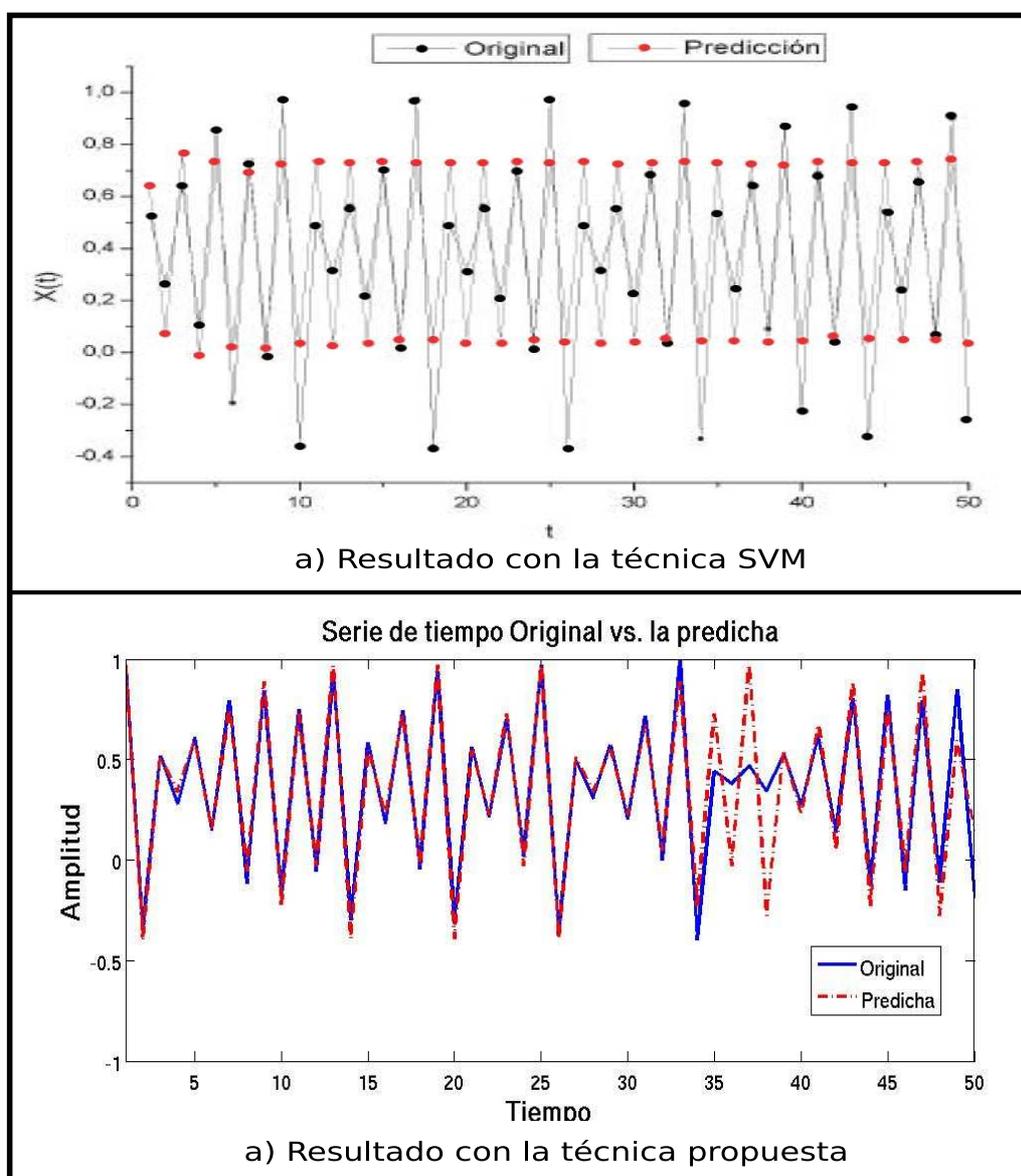


Figura 5.83: Resultados de la mejor predicción obtenida para la ST “tent”, en a) se observa el resultado que se obtuvo con la técnica de SVM y b) muestra los resultados obtenidos con la técnica de predicción que proponemos en este trabajo.

En la Figura 5.84 observamos los resultados de predicción de los últimos 50 puntos de la ST “seno” con la técnica de SVM con una función de kernel lineal y la técnica de predicción que proponemos. Es posible notar que la exactitud de la predicción de los últimos 50 puntos de la ST obtenidos con la técnica SVM disminuye conforme aumenta el número de puntos predichos, esto es, surge un desfazamiento entre los puntos predichos con respecto a los puntos originales (ver Figura 5.84a)), esto es debido al error acumulado. Por otro lado, los resultados de predicción obtenidos mediante la aplicación de la técnica PBI conforme pasa el tiempo no afecta en cuanto a su exactitud en la amplitud y tendencia (ver Figura 5.84b)).

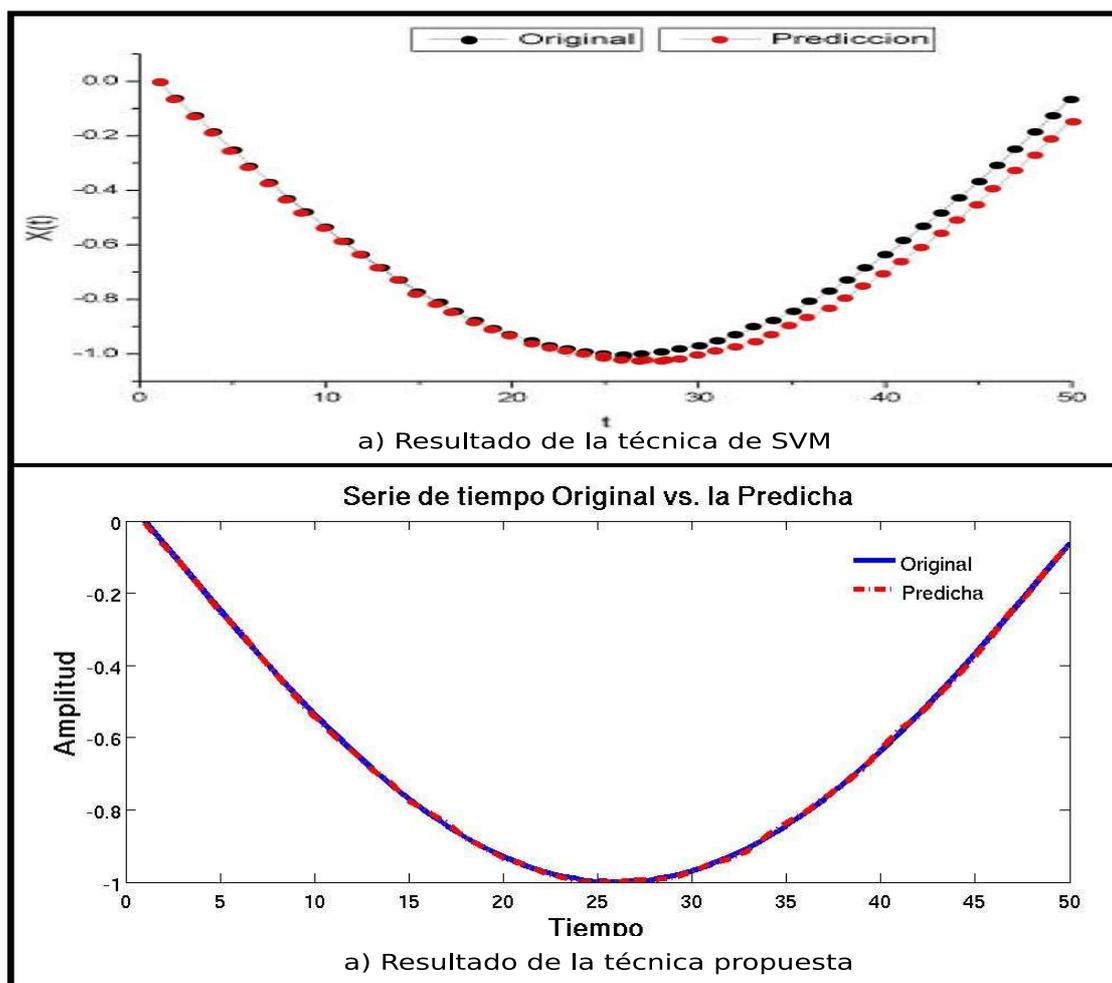


Figura 5.84: Resultados de predicción de la ST “seno”, en a) se observan los últimos 50 puntos predichos con la técnica de SVM y en b) se observa los resultados obtenidos de la técnica propuesta.

En la siguiente Tabla 5.6 se presentan los errores RMSE calculados para los 50 puntos predichos de las 12 ST utilizadas para evaluar la eficiencia de la técnica SVM, las cuales muestran los experimentos exitosos de la técnica SVM con la función kernel radial (FBR), así como también se muestran los resultados obtenidos para dicho conjunto de ST con la

técnica PBI.

Cuadro 5.6: Se observan los errores de predicción (RMSE) obtenidos mediante la aplicación de las técnicas máquinas de soporte vectorial y la técnica propuesta para la predicción de los últimos 50 puntos.

Número	Serie de Tiempo	Máquinas de Soporte Vectorial	Técnica Propuesta
1	Seno	0.0682	0.00002
2	Vanderpol	0.70549	0.00464
3	Qperiodic2	0.17429	0.00349
4	Qperiodic3	0.46989	0.04423
5	Mackey & Glass	0.5007	0.24519
6	Logistic	1.08828	0.53587
7	Lorenz	0.17389	0.00067
8	Rosslor	0.4131	0.01557
11	Tent	0.20708	0.03503
12	A1	7.37899	0.04300
14	Laser	0.32574	0.14717
16	Lovaina	0.16296	0.01268

Nótese que en todos los casos la técnica que proponemos obtiene un error de predicción (RMSE) más bajo que los resultados obtenidos de aplicar la técnica de máquinas de soporte vectorial. Además, nuevamente es importante mencionar que los errores de predicción que se obtienen con la técnica PBI están por debajo de la unidad, un hecho bastante difícil en cuanto a predicción.

5.5.2. Redes neuronales vs. Predicción basada en imagen

En el trabajo de tesis [31] publicado en el año 2006, el autor Landassuri presenta la predicción de series de tiempo usando una técnica de análisis de señales llamada *Descomposición Empírica en Modos* (EMD), la cual proporciona más información sobre el problema, esto con la finalidad de obtener una predicción más precisa. Para lograr esta finalidad, los autores generaron un algoritmo llamado GAAN para diseñar arquitecturas de *Redes Neuronales Artificiales* (ANNs) utilizando *Algoritmo Genético* (GA). Las ANNs se encargaban de realizar la predicción, mientras que el entrenamiento era realizado con el algoritmo de Levenberg-Marquardt, el cual está considerado como el más eficiente, además está considerado como un algoritmo de segundo orden.

La metodología que utilizaron los autores para obtener la mejor predicción de ST fue la siguiente: los datos de entrada del algoritmo podían ser los datos originales o bien los datos originales con las IMFs⁴ al algoritmo, posteriormente se procede a generar la primera generación del GA, el cual se crea de manera aleatoria por ser la primera vez (genotipo), después se decodifica el fenotipo (red neuronal) para cada individuo, de acuerdo a los valores obtenidos

⁴Las “Funciones de Modo Intrínseco” (IMF, por sus siglas en inglés), se obtienen al aplicar la descomposición empírica en modos (EMD) y éstas presentan tiempo y frecuencia al mismo tiempo, algo que el análisis de Fourier no es capaz de hacer.

del genoma necesarios para construir la red neuronal. Posteriormente, se entrena cada una de las redes obtenidas y se introduce un conjunto de prueba, de donde saldrá su adaptabilidad y con esta información se pueden aplicar los operadores del GA (selección, cruzamiento, etc) para crear la siguiente generación. Este procedimiento se repite hasta que se cumpla alguna función de paro. En ese momento el algoritmo nos regresa el mejor individuo encontrado, que corresponde a la arquitectura de la red neuronal que mejor se adaptó a los datos de prueba.

En la siguiente Tabla 5.7, se presentan los errores RMSE calculados para los 30 puntos predichos de las 15 ST utilizadas para evaluar la eficiencia de la técnica de redes neuronales, así como también, se muestran los resultados obtenidos para dicho conjunto de ST con la PBI.

Cuadro 5.7: Se observan los errores de predicción (RMSE) obtenidos mediante la aplicación de la técnica de redes neuronales con respecto a la técnica propuesta para la predicción de los últimos 30 puntos de las ST.

Número	Serie de Tiempo	Redes Neuronales	Técnica Propuesta
3	Qperiodic2	0.1048	0.00802
4	Qperiodic3	0.2470	0.01257
6	Logistic	0.2316	0.15796
7	Lorenz	0.2571	0.00061
8	Rosler	0.0338	0.00428
9	Ikeda	0.4630	0.13621
10	Henon	0.6229	0.17097
13	D1	0.2793	0.02619
14	Laser	22.407	0.15233
16	Lovaina	0.1186	0.00598
17	S & P 500	1.7547	0.08219
18	Star	0.3577	0.03391
19	Flujo Rio Jokulsa	5.7353	0.00007
20	Brownian Motion	0.8108	0.00865
22	Precio del Oro	6.9932	0.00143
23	Rio Colorado	0.3024	0.00819
24	Lago Erie	0.7995	0.06586
25	Rio Nilo	1.1589	0.38271
26	Manchas Solares	25.982	0.00557
27	Temp. Max. Melbourne	6.3652	0.10469
28	Temp. Min. Melbourne	2.3025	0.02834
29	Souther Oscilation	8.068	0.05550

Nuevamente, podemos observar que los resultados obtenidos con la técnica PBI superan los resultados de esta técnica de redes neuronales. También, es posible notar que las predicciones obtenidas con la técnica PBI nuevamente no exceden la unidad, algo que se puede observar es algo complicado para la técnica de redes neuronales.

Capítulo 6

Conclusiones y trabajo futuro

Como comentarios finales se puede decir que durante el desarrollo de esta investigación se implementó un nuevo método de estudio para una técnica de aprendizaje de ST, diferente de las que se encuentran en la literatura actualmente. Esta nueva manera de representar los datos de entrada se propuso de la siguiente manera: se dividió la imagen de la ST en n ventanas, llamadas “cajas” y cada una de éstas contiene una cierta anchura y un adecuado número de puntos.

Posteriormente, generamos una nueva técnica de aprendizaje para el comportamiento dinámico de ST estacionarias y semi-estacionarias, a través de la información obtenida de su imagen. Como se comentó a lo largo de este trabajo, utilizamos el principio de submetas, el cual aplicamos de la siguiente manera: dividimos el problema de aprender toda la dinámica del sistema de alguna ST en problemas más sencillos en donde cada uno de éstos aprende una dinámica local de alguna determinada caja (segmento de la imagen) y la solución del problema de aprendizaje puede ser la suma de las soluciones independientes de cada subproblema o la suma de un subconjunto de éstas. Para probar el rendimiento de nuestro algoritmo, generamos una serie de tiempo que aprendiera el comportamiento de alguna señal objetivo y la siguiera, demostrando así que esta nueva serie aprendió dicho comportamiento. Los resultados indican que con esta nueva técnica es posible aprender el comportamiento de alguna serie de tiempo con buena precisión utilizando cualquiera de las tres tipos de división de la imagen en un número adecuado de cajas.

La información obtenida del proceso de aprendizaje puede servir para aplicarla a un problema más complejo que es el de la *predicción*. Esto es, implementamos un mecanismo de indexación para recuperar la información obtenida durante el proceso de aprendizaje, después empleamos técnicas probabilísticas para tomar decisiones sobre los puntos a predecir. Es importante mencionar que el tipo de predicción que se manejó en esta investigación fue la *predicción iterada a múltiples puntos*, la cual, es de las predicciones más complejas. En los resultados experimentales de nuestra técnica de Predicción Basada en Imagen, se presentan tres características interesantes, la primera se refiere a la predicción de la amplitud de los puntos con respecto a los puntos originales, el cual como observamos en el capítulo anterior, genera un error de predicción global (RMSE) muy pequeño. Por otro lado, la segunda característica se refiere a la tendencia de los puntos predichos, la cual se observa que en la mayoría de los resultados, la tendencia fue mejor incluso que la amplitud. La característica más interesante de nuestra técnica de predicción, es su capacidad de recuperar la convergencia del error de predicción, es decir, encontrar la mejor amplitud y/o tendencia eventualmente. Lo anterior significa que un

posible error durante la predicción no afecta a las futuras predicciones, debido a esta capacidad de convergencia. Finalmente, comparamos los resultados que se obtuvieron de este proceso de predicción con los resultados obtenidos de ejectutar las técnicas modernas como las redes neuronales y máquinas de soporte vectorial y la técnica PBI generó los errores de predicción (RMSE), más bajos entre éstas.

6.1. Trabajo a futuro

A continuación se presenta un conjunto de posibles líneas de trabajo a futuro derivadas de esta tesis.

6.1.1. Mapeo de series de tiempo a redes complejas

El matemático Paul Erdős inventó junto con su colaborador Alfred Rényi la teoría formal sobre los grafos aleatorios. Un grafo aleatorio, como su nombre lo indica, es una red, en donde sus nodos están conectados mediante enlaces aleatorios. Para ilustrar este hecho, utilizemos la analogía del biólogo Stuart Kauffman, imagine tirar una caja llena de botones sobre el piso, después escoja un par de botones (nodos) al azar y únalos con un hilo (enlaces) de tamaño apropiado [54].

Las redes desordenadas, tales como las redes del mundo pequeño, recientemente han sido el centro de atención entre la comunidad académica debido a su potencialidad como modelos para la interacción de redes de sistemas complejos.

Las redes del mundo pequeño están caracterizadas por las siguientes propiedades:

- El vecindario local se preserva.
- El diametro de la red está cuantificado por el promedio de la distancia mínima entre dos vértices. Éste incrementa logarítmicamente de acuerdo al número n de vértices.
- Esta última propiedad, le da el nombre de redes del mundo pequeño a este tipo de grafos, debido a que es posible conectar cualesquiera dos vértices en la red a través de unos cuantos enlaces y la conectividad local podría sugerir que dicha red tiene una dimensionalidad finita.

De acuerdo a lo anterior se puede decir que las redes complejas tienen las siguientes métricas:

- Coeficiente de clustering, el cual determina la probabilidad de que los nodos conectados a un nodo i estén conectados entre sí.
- Distribución de grado, el cual, indica la distribución del grado de los nodos (# de aristas que salen del nodo) y está caracterizado por una distribución $P(k)$. En algunas ocasiones es una distribución de Poisson y en otras ocasiones una ley de potencias (“power law”, en inglés).

Ejemplo de estas redes complejas son: el Internet, actores de películas, colaboración científica, red de contacto sexual, red celular, llamadas telefónicas, citas bibliográficas, lingüística. Con lo anterior, podemos decir que las redes complejas son muy útiles para cierto tipo de problemas digamos donde se pueden inferir los nodos y las conexiones entre éstos. Es posible observar que

esta línea de investigación es muy útil en especial para conocer las estructuras por ejemplo en la lingüística.

Lo que proponemos es unir la teoría matemática que está detrás de las redes complejas y aplicarlo a ST, pues esto nos ayudaría a encontrar información que posiblemente no se obtenga con las técnicas de análisis de ST. Esta primera aproximación lo realizamos con series de tiempo artificiales¹ con comportamiento dinámico estocástico, pues se tiene la teoría de que si al término de este análisis los resultados que obtengamos son parecidos o iguales respecto a los que se obtienen de las redes aleatorias, entonces será posible que sí se pueda hacer esta traducción de ST a redes complejas.

Para lograr lo anterior, primero ejecutamos un algoritmo, el cual genera ruido correlacionado, este algoritmo depende de dos variables de una semilla (para generar los números aleatorios) y una *beta* que está en función de *gamma* de la siguiente manera: $beta = (gamma - 1)/2$. La *gamma* nos indica el tipo de ruido que generaremos, esto es si $gamma = 1$, entonces estaremos generando ruido blanco y así sucesivamente. Posteriormente, generaremos una estadística de distribución de las entradas y salidas de los enlaces de la red y por cada *gamma* ejecutaremos dicho algoritmo 100 veces y esto nos dará como resultado 100 ST con ruido correlacionado con una determinada gama, pero con diferente semilla. Lo anterior se realiza de la siguiente manera:

1. Dividimos cada ST en ventanas de tiempo, donde en cada una tenga 100 puntos.
2. Aplicamos la técnica que proponemos de aprendizaje a cada una de estas ventanas.
3. Obtenemos de este proceso de aprendizaje una matriz parecida a la utilizada en la predicción, la cual si recordamos la llamamos *saltos*.
4. Unimos los resultados de cada dimensión de la matriz *saltos* hacia una sola la cual llamaremos *saltosmerge* y ésta contendrá los saltos de los puntos que pertenezcan a cada ventana.
5. Binarizamos la matriz *saltosmerge*, pues en esta primera aproximación no tomaremos en cuenta la frecuencia de los enlaces. Es importante notar que esta nueva matriz es parecida a una matriz de adyacencia y ésta es una representación de un grafo (red). Este grafo consta como ya mencionamos de nodos y de enlaces, pero estos últimos son dirigidos, es decir, se les asigna un sentido a los enlaces. Entonces, ahora la matriz *saltosmerge* tiene sólo dos dimensiones, donde las filas se refieren a las salidas y las columnas se refieren a las entradas.
6. Concatenamos las matrices generadas para cada ventana en una matriz común llamada *matrizST*, la cual como es de esperarse es de dimensión 100 y como su nombre lo indica esta matriz encapsula las características de todas las ventanas por ST.
7. Finalmente, generamos una distribución para las entradas de cada dimensión de la matriz *matrizST* de las 100 ST generadas con el mismo valor de *gamma* y lo mismo hacemos para las salidas. De tal manera que, al final tenemos por cada valor de *gamma* dos distribuciones, una correspondiente a las entradas y la otra a las salidas.

Como podemos observar de los pasos anteriores estamos uniendo información del tiempo y de la amplitud. Los pasos anteriores se observan en la siguiente Figura 6.1.

¹Los puntos de las ST artificiales se obtienen através de alguna fórmula matemática

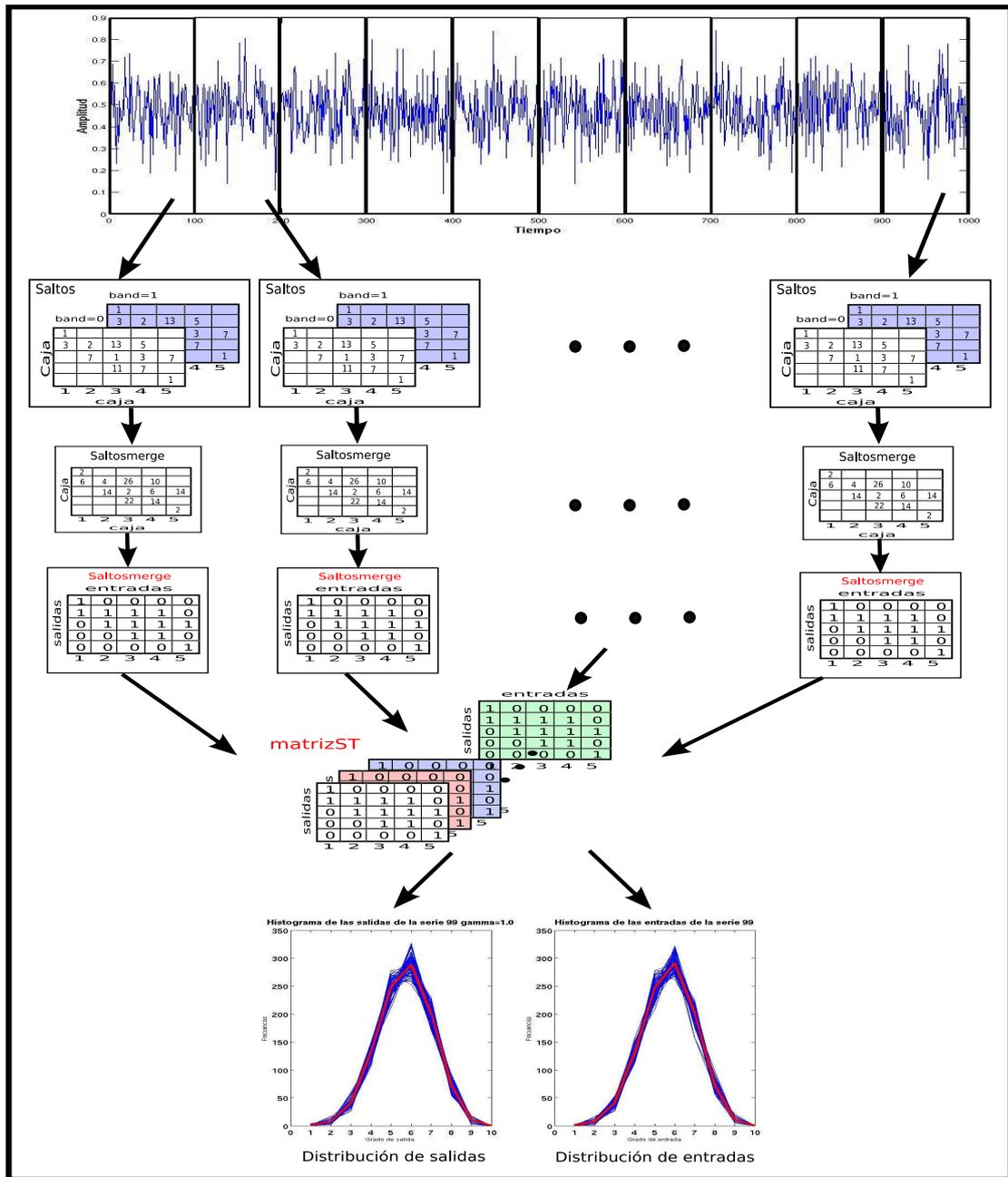


Figura 6.1: Metodología empleada para obtener los primeros resultados de la hipótesis de analizar las ST con técnicas del mundo pequeño.

Este trabajo lo estamos realizando actualmente en colaboración con el Dr. Lev Guzman Vargas.

6.1.2. Agregar información global a la predicción basada en imagen

Si observamos nuevamente la metodología empleada en el proceso de aprendizaje, donde teníamos una matriz llamada *longSaltos*, en donde se almacenaba en las filas la longitud de los puntos, en la columnas el número de caja en la que estaban actualmente los puntos, la siguiente correspondía a la caja hacia la que brincaban los futuros puntos y finalmente se tenía una última dimensión que correspondía al sentido de los puntos. La matriz anterior únicamente almacenaba las frecuencias de los puntos que cumplieran con ciertas características para cada caja, esto es información local.

De lo anterior, podemos decir que la predicción de ST se realiza únicamente en base a las decisiones locales de cada caja y para llegar a la predicción de los *n* puntos a predecir, se unen los resultados de las cajas utilizadas. Ahora, si nosotros además de esta información local, utilizáramos información global, ¿ésta ayudaría en nuestra predicción?, esa es la segunda hipótesis que se genera de esta investigación. Lo anterior se esquematiza en la Figura 6.2.

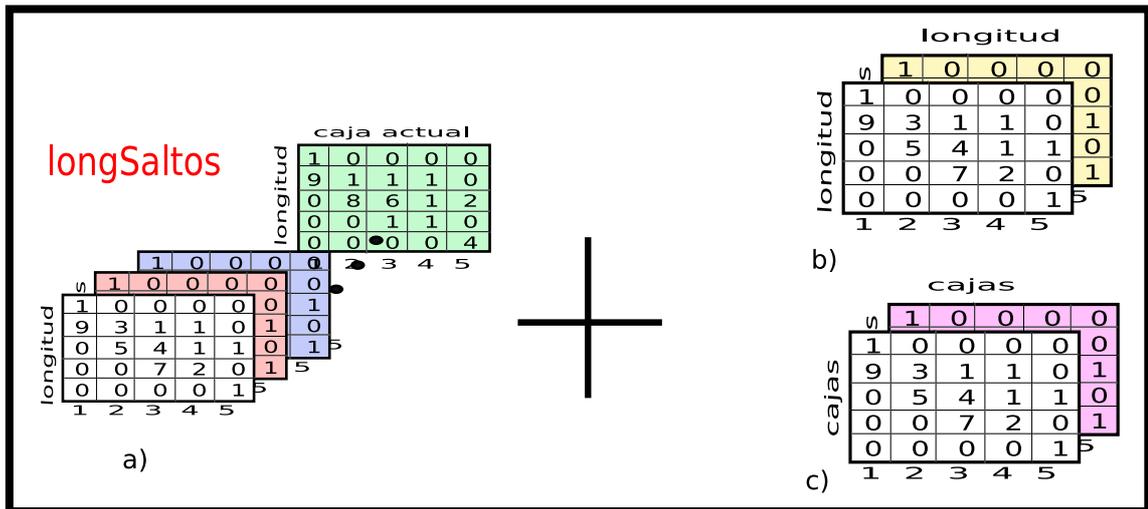


Figura 6.2: Esquema explicativo de como agregar la información global para la toma de decisiones en la predicción.

Después de implementar el incremento de la información global para la toma de decisiones, una siguiente interrogante sería ¿sí a esta técnica le implementáramos aprendizaje por reforzamiento? Esto es, premiaríamos o castigáramos su desempeño, éste ayudaría en los resultados. Además, sería interesante que la predicción se estuviera actualizando de manera periódica con los puntos originales, es decir, implementar también la retroalimentación a dicha técnica de predicción.

Finalmente, existen ST de las cuales se conoce su relación con alguna otra variable, por lo que para este tipo de ST sería interesante ampliar la técnica para que ésta sea multivariada.

Bibliografía

- [1] Matute, H., Vegas, S., & De Marez, P. J.: *Flexible use of recent information in causal and predictive judgments*. Journal of Experimental Psychology: Learning, Memory and Cognition, 28, 714-725, 2002.
- [2] Matute, H.: *Del aprendizaje natural al aprendizaje artificial*. Publicado en Psicoteca, 2004.
- [3] Michie D.: *Experiments on the Mechanization of Game-Learning Part I. Characterization of the Model and its Parameters*. Computer Journal **6** pp. 232-236, 1963.
- [4] Michie D., Chambers, R. A.: *Boxes: An Experiment in Adaptive Control*. eds. E. Dale and D. Michie. Machine Intelligence 2. pp. 125-133, 1968.
- [5] Newell A., Shaw J.C., Simon H.A.: *A Variety of Intelligent Learning*. eds. C Marshall, Yo-vits and Scott Cameron. London: Pergamon: in a general problem solver in Self-organizing Systems, pp. 153-189, 1960.
- [6] Kantz H., Schreiber T.: *Nonlinear Time Series Analysis*. Cambridge: Cambridge University Press, 1997.
- [7] Sprott J.C. : *Chaos and Time-Series Analysis*.Oxford: Oxford University Press, ISBN 0-19-850839-5, 2003.
- [8] Ramírez-Amaro K., Figueroa-Nazuno J.G.: *Empleo de la Técnica Mapa Recurrente en el Análisis de los Índices de Teleconexión Atmosféricas*. México, D.F. Centro de Investigación en Computación, I.P.N. Reporte Técnico. ISBN: 970-36-0330-0, 2006.
- [9] Morik K., Wrobel S., Kietz J-U.,Emde W.: *Knowledge Acquisition and Machine Learning: Theory, Methods and Applications*. Academic Press. ISBN 0-12-506230-3, 1933.
- [10] Motwani R., Raghavan P.: *Randomized Algorithms*. Cambridge University Press, New York (NY), 1995.
- [11] Castillo-Montiel E., Paredes-Farrera M. A., Bustillo-Hernández C., Ramírez-Amaro K. & Figueroa-Nazuno J.: *Evaluación de Modelos de Predicción de Series de Tiempo*. ROC&C 2006, Acapulco Guerrero, Diciembre, 2006.
- [12] Bartle R.G., Sherbert D.R.: *Introducción al Análisis Matemático de una Variable*. Limusa Wiley, ISBN: 968-18-5191-9. 2004.
- [13] Jaulin L., Kieffer M., Didrit O., Walter E.: *Applied Interval Analysis*. Springer-Verlag London, ISBN: 1-85233-219-0. 2001.

- [14] Frigo M., Leiserson C.E., Prokop H.: *Cache-oblivious Algorithm*. Proc. 40th Symp. on the Foundations of Computer Science, 1999.
- [15] Ramírez-Amaro K., Chimal-Eguía J.C.: *On a New Approach to Time Series Tracking*. Advances in Computer Science and Engineering. eds. Torres S., López I. and Calvo H. Research in Computer Science 27, pp. 205-215, México, 2007.
- [16] Ramírez-Amaro K., Chimal-Eguía J.C.: *New Approach to Time Series Tracking*. 27TH International Symposium on Forecasting, New York City, USA 24-27, June 2007.
- [17] Ramírez-Amaro K., Chimal-Eguía J.C.: Una Nueva Técnica para Aprender el Comportamiento de Series de Tiempo. 6ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI'07), Orlando, Florida, EE.UU (2007).
- [18] Diccionario de la Real Academia Española, <http://www.rae.es/>.
- [19] Simon H. A.: *Why Should Machines Learn?* In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pp. 25-38. Tioga, Palo Alto, CA, 1983.
- [20] Mitchell T.M., *Machine Learning*. McGraw Hill, ISBN: 0070428077, 1997.
- [21] Michalski R. S.: *Understanding the Nature of Learning*. In Michalski, Carbonell and Mitchell, editors, *Machine Learning - An Artificial Intelligence Approach*, volume II. Morgan Kaufmann, Los Altos, California, 1986.
- [22] Box, G.E.P. and Jenkins, G.M.: *Time Series Analysis, Forecasting and Control*. San Francisco: Holden-Day (revised edn., 1976), 1970.
- [23] Chatfield C.: *Time Series Forecasting*. United Kingdom: Chapman & HALL/CRC, ISBN: 1-58488-063-5, 2000.
- [24] Forrester J. W.: *Dinámica Industrial*. Editorial Ateneo, Buenos Aires, 1981.
- [25] Rüping S.: *SVM Kernels for Time Series Analysis*. En Klinkenberg R. and S. Rüping, A. Fick, N. Henze, C. Herzog, R. Molitor y O. Schröder, editores, LLWA 01 - Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität in series Forschungsberichte des Fachbereichs Informatik der Universität Dortmund, 2001, pp. 43-50.
- [26] Rüping S. : *mySVM-Manual*. University of Dortmund, Lehrstuhl Informatik 8, <http://www-ai.cs.unidortmund.de/SOFTWARE/MYSVM/>, 2000.
- [27] Lodi H. C., Saunders J., Shawe-Taylor N., Cristianini, Watkins C.: *Text Classification using String Kernels*. Journal of Machine Learning Research, Vol. 2, 2002, pp. 419-444.
- [28] Müller K. R., Samola A., Rätsh G., Schölkopf B. , Kohlmorgen J., Vapnik V.: *Predicting Time Series with Support Vector Machines*. Proceedings of ICANN'97, Springer LNCS 1327, 1997, pp. 999-1004.
- [29] Yeang C., Ramaswamy S., Tamayo P., Mukherjee S., Rifkin R., Angelo M., Reich M., Lander E., Mesirov J., Golub T.: *Molecular Classification of Multiple Tumor Types*. Bioinformatics, Vol. 17, Suppl 1, 2001, pp. S316-22.

- [30] Bautista–Thompson E., Guzmán–Ramírez E., Figueroa–Nazuno J.: *Predicción de Múltiples Puntos de Series de Tiempo Utilizando Support Vector Machines*. Computación y Sistemas. Vol. 7 No. 3 pp. 148 - 155 CIC-IPN, ISSN 1405-5546 Impreso en México, 2004.
- [31] Landassuri-Moreno V. M.: *Predicción de Series de Tiempo con Descomposición Empírica en Modos, Algoritmo Genético y Redes Neuronales Artificiales*. Tesis de Maestría, Centro de Investigación en Computación del Instituto Politécnico Nacional. 2006.
- [32] Doynne F., Sidorowich, J.J.: *Predicting Chaotic Time Series*. Physical Review Letters (August 1987). 59(8), 845.
- [33] Yoshua B.: *Using a Financial Training Criterion Rather than a Prediction Criterion*. Int. J. Neural Syst (1997). 8(4): 433-443.)
- [34] Feng L., Huo Yu X., Gregor S., Irons R.: *Time Series Forecasting with Neural Networks*. Complexity International (1995). Vol 2. ISSN: 1320-0682
- [35] Branke J.: *Evolutionary Algorithms for Neural Networks Design and Training*. Technical Report No. 322, University of Karlsruhe, Institute AIFB.
- [36] Dorffner G.: *Neural Networks for Time Series Processing*. Neural Network World (1996), 6(4):447-468.
- [37] Cholewo T.J., Zurada J. M.: *Sequential Network Construction for Time Series Prediction*. In Proceedings of the IEEE International Joint Conference on Neural Networks, Houston, Texas, USA (June 1997). Pp 2034-2039.
- [38] Zhang G., Patuwo B., Hu M.: *Forecasting with Artificial Neural Networks: The State of the Art*. Elsevier, International Journal of Forecasting (1998). 14 35-62, N. H.
- [39] Medina-Apodaca J. M.: *Herramienta para la Extracción de Redes Bayesianas Predictivas a partir de Bases de Datos Temporales*. Tesis de Maestría. Centro de Investigación en Computación del Instituto Politécnico Nacional, 2005.
- [40] Rumelhart D. E., Hinton G. E., Williams R. J.: *Learning Representations by Back-Propagating Errors*. Nature (1986), vol. 323, pp. 533-536.
- [41] McCulloch, W., Pitts, W.: *A Logical Calculus of the Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics, 7:115 - 133. 1943.
- [42] Vapnik V.: *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
- [43] Cristianini N., Shawe-Taylor J.: *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*. Cambridge University Press, UK, 2000.
- [44] Wan E. A., Nelson A. T.: *Dual Kalman Filtering Methods for Nonlinear Prediction, Smoothing and Estimation*. NIPS pp: 793-799. 1996.
- [45] De Gooijer J. G, Hyndman R.J.: *25 Years of Time Series Forecasting*. Elsevier, International Journal of Forecasting 22 pp. 443-473, 2006.
- [46] Ott E.: *Chaos in Dynamical Systems*. Cambridge University Press, Cambridge, 2000.

- [47] Ott, E., Sauer T., Yorke, J.A. (Editores): *Coping with Chaos. Analysis of Chaotic Data and the Exploitation of Chaotic Systems*. John Wiley & Sons, New York, 1994.
- [48] Mackey M.C., Glass L.: *Oscillation and Chaos in Physiological Control Systems*. Science, vol 197, pp. 287- 289, 1977.
- [49] Lorenz E.N.: *Deterministic Non periodic Flow*. J. Atmos. Sci., vol. 20, pp. 130-141, 1963.
- [50] Rossler O.E.: *An Equation for Continuous Chaos*. Physics Letters A, vol. 57, pp. 397-398, 1976.
- [51] Hénon M.: *A Two-Dimensional Mapping with a Strange Attractor*. Comm. Math. Phys., vol. 50, pp. 69-77, 1976.
- [52] Weigend A., Gershenfeld N. (Eds.): *Time Series Prediction. Forecasting the Future and Understanding the Past*. SFI Studies in the Science of Complexity, Proc. Vol XVII, Addison-Wesley, Reading, 1994.
- [53] Suykens J.A.K., Vandewalle J. (Eds.): *Nonlinear Modeling Advanced Black-Box Techniques*. Kluwer Academic Press. Amsterdam, 1998.
- [54] Watts D. : *Six Degrees: The science of a connected age*. Norton & Company, Inc. ISBN:0-393-32542-3.